



Integration Guide | PUBLIC

SAP Marketing Cloud

2021-11-05

# Integration Guide

# Content

- 1 Overview of Integration Options. . . . . 6**
- 2 Document History. . . . . 8**
- 3 Implementing Integrations for Business Scenarios. . . . . 9**
  - 3.1 Business Scenario: Dynamic Customer Profiling and Segmentation. . . . . 10
  - 3.2 Business Scenario: Campaign and Journey Orchestration. . . . . 11
  - 3.3 Business Scenario: Commerce Marketing. . . . . 16
  - 3.4 Business Scenario: Lead- and Account-Based Marketing. . . . . 19
  - 3.5 Business Scenario: Marketing Planning and Performance. . . . . 21
  - 3.6 Business Scenario: Marketing Analytics. . . . . 23
    - Core Data Services-Based Extraction from SAP Marketing Cloud to SAP BW Systems. . . . . 24
    - Core Data Services-Based Extraction from SAP Marketing Cloud to Other SAP and Non-SAP Systems. . . . . 25
    - Models for Core Data Service-Based Extractions. . . . . 32
- 4 Integration Scenarios. . . . . 49**
  - 4.1 Overview of Integration Scenarios (Table). . . . . 50
  - 4.2 Inbound. . . . . 61
    - Commerce, Social Media, Web, and IoT. . . . . 62
    - Landing Pages and Forms. . . . . 91
    - Survey. . . . . 107
    - Extensions. . . . . 108
  - 4.3 Outbound. . . . . 108
    - Sending Emails and Text Messages. . . . . 109
    - Setting Up External Campaign Execution. . . . . 155
    - Open Channel Integration. . . . . 194
    - Mobile, Social, and Digital Channel. . . . . 242
    - Setting Up Captcha Configuration for Forms. . . . . 264
  - 4.4 Application-Enabling Integrations. . . . . 267
    - Integrating Custom Themes. . . . . 267
    - Integration with SAP Analytics Cloud (ISO). . . . . 268
    - Content Studio Integrations. . . . . 304
    - Enabling Geospatial Segmentation with here.com. . . . . 317
    - Integration with Baidu Maps for Geospatial Segmentation (Deprecated). . . . . 318
    - SAP Jam Integration for Collaboration. . . . . 320
    - Verifying Email Addresses Using a Partner Solution. . . . . 321
    - Integration with an External Coupon Service System. . . . . 321

	Partner Extension: Integrate with Digital Market Intelligence. . . . .	330
	Marketing Events. . . . .	330
4.5	Suite-Enabling Integrations. . . . .	333
	Sales and Service (Inbound). . . . .	334
	Sales Automation (Outbound). . . . .	352
	Financial Data. . . . .	379
	Survey Data. . . . .	383
	Personalized Commerce. . . . .	385
<b>5</b>	<b>Integration APIs. . . . .</b>	<b>387</b>
5.1	Getting Started. . . . .	387
	Videos - Best Practices for Data Load. . . . .	388
	Quick Guide - Which API for Which Entity . . . . .	389
	Consuming the Integration APIs. . . . .	395
	Extending the Integration APIs. . . . .	396
	Optimize Performance During OData Service Calls. . . . .	397
	Best Practices and Recommended Package Sizes. . . . .	400
	Import Monitor. . . . .	404
	Data Load Monitor. . . . .	407
	HTTP Response Status Codes. . . . .	408
5.2	Contact Profiling. . . . .	410
	Contacts. . . . .	412
	Interaction Contacts. . . . .	469
	Corporate Accounts. . . . .	512
	Business Partners from SAP Cloud for Customer. . . . .	555
	Import Business Partners. . . . .	574
	Products. . . . .	582
	Product Hierarchies and Categories. . . . .	604
	Interactions. . . . .	615
	Interest Items. . . . .	648
	Business Documents . . . . .	661
	Agreements. . . . .	681
	Scores. . . . .	700
	Marketing Locations. . . . .	710
	Classifications (Deprecated). . . . .	722
	Marketing Attribute Categories. . . . .	735
	Import Monitoring. . . . .	740
5.3	Landing Pages. . . . .	743
	External Landing Pages. . . . .	743
	External Landing Page Value Help. . . . .	749
5.4	Segmentation. . . . .	754
	Target Groups. . . . .	755

	Export Target Groups and Target Group Member Data. . . . .	761
5.5	Campaign Management. . . . .	761
	Campaign and Target Group Data. . . . .	761
	Campaign Execution Plans. . . . .	763
	Campaigns. . . . .	767
	Campaign Templates. . . . .	788
	Campaign Message Content and Personalized Email Content. . . . .	793
	Campaign Success Data. . . . .	812
	Import Campaign Performance Data. . . . .	817
	Survey. . . . .	886
	Read Content of Export Files in Campaigns. . . . .	901
	Marketing Events. . . . .	904
5.6	Commerce Marketing. . . . .	922
	Recommendations (SAP Business Technology Platform). . . . .	923
	Recommendations. . . . .	939
	External Recommendations. . . . .	954
	Recommendations Interaction Data. . . . .	971
	Import Offers. . . . .	973
	Read Offers. . . . .	1002
	Discover Offers. . . . .	1008
	Coupons. . . . .	1026
5.7	Marketing Analytics. . . . .	1032
	Import Analytical Data for Marketing Executive KPI. . . . .	1033
5.8	Marketing Planning and Performance. . . . .	1040
	Actual and Committed Spend Data. . . . .	1041
	Marketing Programs. . . . .	1045
5.9	Custom Business Objects. . . . .	1057
	Import of Data into Custom Business Object. . . . .	1057
5.10	Business Users. . . . .	1058
	Business User. . . . .	1058
	Business User - Read. . . . .	1070
	Business User - Read Metadata. . . . .	1085
<b>6</b>	<b>Business Event Handling. . . . .</b>	<b>1092</b>
6.1	Campaign File Export. . . . .	1093
6.2	Campaigns. . . . .	1094
6.3	Coupon Code Usages. . . . .	1095
6.4	Interactions. . . . .	1096
6.5	Interaction Contacts. . . . .	1097
6.6	Marketing Permissions. . . . .	1098
6.7	Marketing Subscriptions. . . . .	1099



<b>7</b>	<b>Integration Technologies</b> .....	<b>1101</b>
<b>8</b>	<b>Create Your Own Apps: SAP Rapid Application Development by Mendix</b> .....	<b>1102</b>

# 1 Overview of Integration Options

This guide explains the different integration options with SAP Marketing Cloud. Its aim is to help you quickly find the documentation that will guide you through the integration process from wherever your integration journey starts.

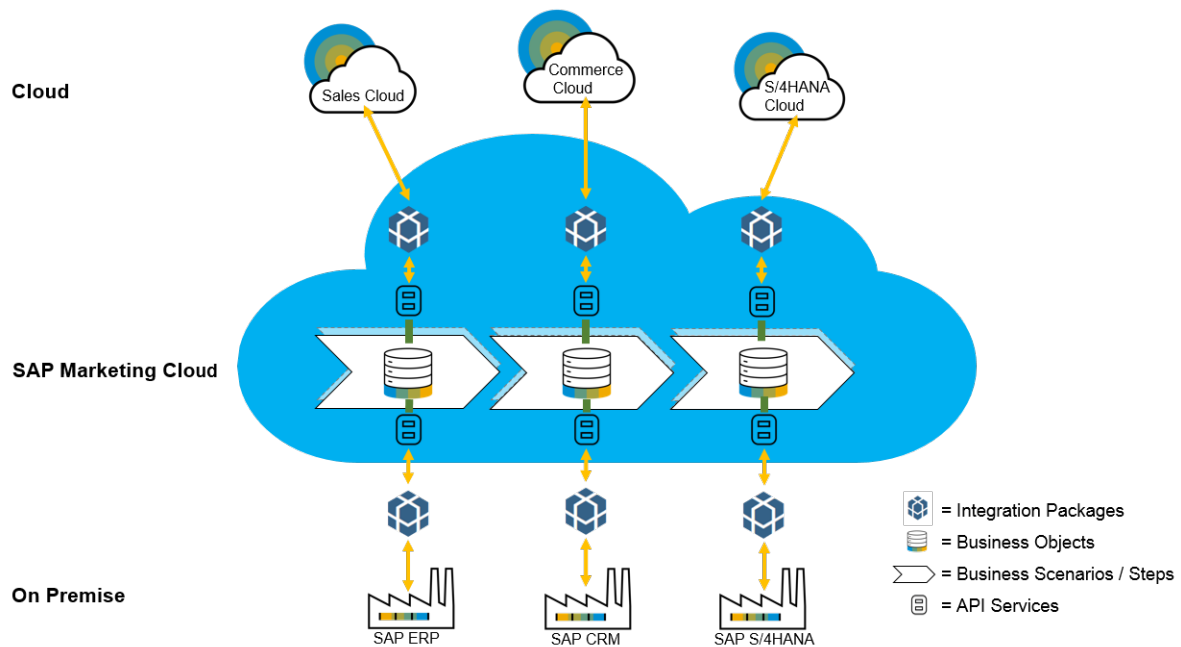
## i Note

Before you start, make sure you have the required version of this document. You can find the available versions at the following location: <https://help.sap.com/mkt>

## i Note

In the PDF version of the guide, some links to topics may be missing. All links are available in the HTML version of the guide.

## Integration Options



The graphic shows only the major integration options.

## Where to Find the Information You Need

Questions This Guide Answers	Read Me
Which business scenarios do I want to implement and how do I connect with the outside world?	<p>Learn more about the integration options from a marketing process perspective.</p> <p><a href="#">Implementing Integrations for Business Scenarios [page 9]</a></p>
How do I connect the marketing solution to an existing solution, for example, to a Sales solution from SAP?	<p>Find out how to bridge the gap between marketing and other business domains that are built on cloud or on premise solutions.</p> <p><a href="#">Integration Scenarios [page 49]</a></p>
Which API should I use if I want to integrate a third-party data source that provides, for example, <b>agreement</b> , <b>campaign</b> , or <b>contact</b> information?	<ul style="list-style-type: none"><li>• <a href="#">Quick Guide - Which API for Which Entity [page 389]</a></li><li>• <a href="#">Integration APIs [page 387]</a></li></ul>

## 2 Document History

The following table provides an overview of the most important document changes.

Document History

Date	Description
2021-10-27	Initial version for the Integration Guide 2111.

# 3 Implementing Integrations for Business Scenarios

Overview of business scenarios, their scope items, related main and additional integration activities, as well as configuration and business administration activities.

A business scenario is a sequence of business processes designed to achieve key business objectives. A scope item is a self-contained and reusable entity of predefined content for the implementation of a business process. A business scenario can include one or more scope items.

You can find scope items in the *Manage Your Solution* application under *View Solution Scope*.

## i Note

You can use a business scenario only if its corresponding scope item is active in your system, that is, it has a green flag.

Scope items that are active in your system will only work if you have the required licenses and the integration to the corresponding application is implemented.

If a scope item is not active in your system, it has a gray sign. If you want to activate a scope item, contact SAP.

Integration of SAP Marketing Cloud with external systems can be achieved by using integration scenarios which are predelivered packages or by using integration services, such as public APIs.

Furthermore, communication scenarios are technical references used to enable the integration of SAP Marketing Cloud with external systems. A communication arrangement describes a communication scenario with a remote system during configuration time and provides the necessary metadata for service configuration.

The following business scenarios and scope items are available:

- [Business Scenario: Dynamic Customer Profiling and Segmentation \[page 10\]](#)
  - Scope Item: Consumer and Customer Profiling (JC1)
- [Business Scenario: Campaign and Journey Orchestration \[page 11\]](#)
  - Scope Item: Segmentation and Campaign Execution by Email (JC2)
  - Scope Item: External Campaigns (JC9)
  - Scope Item: Facebook Campaigns (JC6)
  - Scope Item: Trigger-Based Campaigns and Trigger-Based Campaigns - with Abandoned Shopping Cart (JC8)
  - Scope Item: Google Ads Campaigns (Assign) and Google Campaign Manager (Assign) (JC7)
  - Scope Item: Asian Network Campaigns (23T)
  - Scope Item: Permission Marketing (1T1)
  - Scope Item: Marketing Events (3ZE)
- [Business Scenario: Commerce Marketing \[page 16\]](#)
  - Scope Item: Product Recommendation (JC3)
  - Scope Item: Offer Recommendation (1SW)

- Scope Item: Offer and Coupon Management in Marketing (1HQ)
  - Offers with Coupons on Mobile App
  - Offers with External Coupon Service
  - Offer and Coupons with External Services
- [Business Scenario: Lead- and Account-Based Marketing \[page 19\]](#)
  - Scope Item: Marketing Lead Management (JC0)
  - Scope Item: Marketing-Driven Sales Enablement (1SY)
  - Scope Item: Marketing Lead Nurturing (2ZM)
- [Business Scenario: Marketing Planning and Performance \[page 21\]](#)
  - Scope Item: Marketing Planning (JC5)
- [Business Scenario: Marketing Analytics \[page 23\]](#)
  - Scope Item: Analytics Extensibility and Data Extraction (3SM)

## 3.1 Business Scenario: Dynamic Customer Profiling and Segmentation

Overview of the *Dynamic Customer Profiling and Segmentation* business scenario, its scope item, related main and additional integration activities, as well as configuration and business administration activities.

For information about the business scenario and its corresponding process steps, see [Dynamic Customer Profiling and Segmentation](#).

### Scope Item: Dynamic Customer Profiling (JC1)

#### Main Integration Activities

You can perform the following main integration activities for this scope item:

- [Contacts \[page 412\]](#) (SAP\_COM\_0207) integration service
- [Interaction Contacts \[page 469\]](#) (SAP\_COM\_0206) integration service
- [Corporate Accounts \[page 512\]](#) (SAP\_COM\_0207) integration service
- [Enabling Geospatial Segmentation with here.com \[page 317\]](#)

#### Additional Integration Activities

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- To import customer and contact data, as well as sales volume data like quotes, orders and returns, use the [SAP ERP Integration with SAP Marketing Cloud \(1KW\)](#) scope item with SAP\_COM\_0060 communication scenario. For more information, see [SAP ERP Integration with SAP Marketing Cloud](#).
- [Integration with SAP Commerce](#) (SAP\_COM\_0082) communication scenario and [Integration with SAP Product Content Management](#) (SAP\_COM\_0051) integration scenario. For more information, see [Integration with SAP Commerce Cloud \[page 62\]](#) and [Integration with SAP Product Content Management \[page 314\]](#).



- [Google Analytics Integration](#) (SAP\_COM\_0079) integration scenario. For more information, see [Integration with Google Analytics \[page 87\]](#).
- [SAP Customer Data Cloud Integration for Contacts and Accounts](#) (SAP\_COM\_0264) integration scenario. For more information, see [SAP Customer Data Cloud and SAP Marketing Cloud \[page 338\]](#).
- To integrate with SAP Cloud for Customer, the following integration scenarios are available:
  - [Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)
  - [Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)
- [SAP S/4HANA Integration with SAP Marketing Cloud](#) (23L) scope item. For more information, see [SAP S/4HANA Integration with SAP Marketing Cloud](#).
- To use scores with this business scenario, you must set up a predictive scenario with external score values. For more information, see [Predictive Scenarios](#). The [Scores](#) (SAP\_COM\_0307) integration service is available. For more information, see [Scores \[page 700\]](#).

### Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- [Contacts and Profiles](#)
- [Segmentation Configuration](#)
- [General Settings](#)
- [Map Free Texts](#)
- [Manage Interests](#)
- [Marketing Attribute Categories](#)

## 3.2 Business Scenario: Campaign and Journey Orchestration

Overview of the [Campaign and Journey Orchestration](#) business scenario, its scope items, related main and additional integration activities, as well as configuration and business administration activities.

For information about the business scenario and its corresponding process steps, see [Campaign and Journey Orchestration](#).

### Scope Item: Segmentation and Campaign Execution by Email (JC2)

#### Main Integration Activities for Sinch

You can perform the following main integration activities for this scope item:

- [Marketing – Campaign Execution E-Mail Integration](#) (SAP\_COM\_0040 for emails and SAP\_COM\_0041 for text messages) integration scenario. For more information, see [Setting Up Service Provider for Emails and Text Messages \[page 110\]](#).
- [Enabling Geospatial Segmentation with here.com \[page 317\]](#)

## Main Integration Activities for Amazon

You can perform the following main integration activities for this scope item:

- *Marketing – Campaign Execution E-Mail Integration* (SAP\_COM\_0016 for emails and SAP\_COM\_0039 for bounces) integration scenario. For more information, see [Setting Up Amazon \[page 138\]](#).
- [Enabling Geospatial Segmentation with here.com \[page 317\]](#)

## Additional Integration Activities

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- *Marketing – Campaign Open Channel Integration* (SAP\_COM\_0049) integration scenario. For more information, see [Open Channel Integration \[page 194\]](#).
- *Marketing – Generic DAM Integration* (SAP\_COM\_0050) integration scenario. For more information, see [Integrate with Content Management Systems or Digital Asset Management Systems \[page 306\]](#) and [Integrate with SAP Document Center \[page 312\]](#).
- *Marketing – SAP Product Content Management Integration* (SAP\_COM\_0051) integration scenario. For more information, see [Integration with SAP Product Content Management \[page 314\]](#).
- *Marketing - Campaign Message Integration* (SAP\_COM\_0208) integration service. For more information about exporting and importing message content for multiple languages, see [Campaign Message Content and Personalized Email Content \[page 793\]](#).
- *Marketing - Export File Content Integration* (SAP\_COM\_0311) integration service. For more information about reading export file content in campaigns, see [Read Content of Export Files in Campaigns \[page 901\]](#).

## Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- [Define Marketing Permission Check](#)
- [Activate Campaign Triggers](#)
- [Custom Business Objects](#)
- [Custom Fields in Segmentation](#)
- [Segmentation Configuration](#)
- [Workflow for Business Objects](#)
- [Sender Profiles \[page 155\]](#)
- [Managing Approval Workflows](#)
- [Verifying Email Addresses Using a Partner Solution \[page 321\]](#)

## Scope Item: Marketing Events (3ZE)

### Main Integration Activities

You can perform the following main integration activities for this scope item:

- *Marketing - Marketing Events Integration* (SAP\_COM\_0474) will be deprecated in a future release. *Marketing - Event Outbound Integration* (SAP\_COM\_0541) and *Marketing - Event Inbound Integration* (SAP\_COM\_0371) integration scenarios.  
For more information, see in the integration flow guide [Integrating Marketing Events Data with SAP Marketing Cloud](#) under [Create Communication Arrangement](#).

## Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- If event data is to be fetched from different event provider platforms, define ID origin using the ID Origin configuration application. For more information, see [Configuring Origins](#).
- If you want to create an event with a specific media type, for example, `EVENTS`, define media types using the [Media Types](#) configuration application. For more information, see [Media Types](#).
- If you want to assign an event to a specific marketing area, for example, `GLOBAL`, define marketing areas using the Marketing Areas configuration application. For more information, see [Marketing Areas](#).
- Create and schedule application jobs to import marketing events data from event provider platforms. For more information, see [Marketing Events: Import Marketing Events](#).

## Scope Item: External Campaigns (JC9)

### Main Integration Activities

You can perform the following main integration activities for this scope item:

- [Marketing – External Campaign Execution](#) (`SAP_COM_0037`) integration scenario. For more information, see [Setting Up External Campaign Execution \[page 155\]](#).

### Additional Integration Activities

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- [Marketing – Campaign Success Integration](#) (`SAP_COM_0390`) integration service. For more information, see [Campaign Success Data \[page 812\]](#).

## Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- [Define Marketing Permission Check](#)
- [Activate Campaign Triggers](#)
- [Custom Business Objects](#)
- [Custom Fields in Segmentation](#)
- [Segmentation Configuration](#)

## Scope Item: Facebook Campaigns (JC6)

### Main Integration Activities

You can perform the following main integration activities for this scope item:

- [Marketing – Campaign Execution Facebook Integration](#) (`SAP_COM_0031`) integration scenario. For more information, see [Social Campaigns Using Facebook and Instagram \[page 254\]](#).

## Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- [Contacts and Profiles](#)
- [Campaigns](#)
- [General Settings](#)
- [Segmentation Configuration](#)

## Scope Item: Trigger-Based Campaigns and Trigger-Based Campaigns - with Abandoned Shopping Cart (JC8)

### Main Integration Activities

You can perform the following main integration activities for this scope item:

- [Marketing – Campaign Execution E-Mail Integration](#) (SAP\_COM\_0016) integration scenario. For more information, see [Setting Up Service Provider for Emails and Text Messages \[page 110\]](#).

### Additional Integration Activities

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- [Marketing – Landing Pages Integration](#) (SAP\_COM\_0023) integration scenario. For more information, see [Custom Integration of Forms \[page 92\]](#).
- For the abandoned shopping cart process, the [Marketing - SAP Commerce Data Integration](#) (SAP\_COM\_0082) integration scenario is required. For more information, see [Integration with SAP Commerce Cloud \[page 62\]](#).

## Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- [Define Marketing Permission Check](#)
- [Activate Campaign Triggers](#)
- [Custom Business Objects](#)
- [Custom Fields in Segmentation](#)
- [Segmentation Configuration](#)
- [Sender Profiles \[page 155\]](#)

## Scope Item: Google Ads and Google Campaign Manager Campaigns (JC7)

### Main Integration Activities

You can perform the following main integration activities for this scope item:

- [Marketing - Google Ads Integration](#) (SAP\_COM\_0030) integration scenario. For more information, see [Integration with Google Ads \[page 243\]](#).

- [Marketing - External Campaign Integration](#) (SAP\_COM\_0037). For more information, see [Integration with Google Campaign Manager](#) [page 259].

## Scope Item: Asian Network Campaigns (23T)

### Main Integration Activities for WeChat Campaigns

For the *WeChat Campaigns* variant of this scope item, perform the following main integration activities:

- [Marketing – Network Channel Events Integration](#) (SAP\_COM\_0174) integration scenario
- [Marketing – Campaign Execution – WeChat Integration](#) (SAP\_COM\_0085) integration scenario
- [Marketing – Baidu Map Integration](#) (SAP\_COM\_0075) integration scenario

### Configuration and Business Administration Activities for WeChat Campaigns

Perform the following configuration and business administration activities:

- Create WeChat official accounts.
- Create and schedule application jobs.
- Activate the *All China Consumers (B2C)* segmentation profile.

For more information, see [WeChat Integration](#) [page 65] and [Integration with Baidu Maps for Geospatial Segmentation \(Deprecated\)](#) [page 318].

### Main Integration Activities for LINE Campaigns

For the *LINE Campaigns* variant of this scope item, perform the following main integration activities:

- [Marketing – Network Channel Events Integration](#) (SAP\_COM\_0174) integration scenario
- [Marketing – Campaign Execution – LINE Integration](#) (SAP\_COM\_0218) integration scenario

### Configuration and Business Administration Activities for LINE Campaigns

Perform the following configuration and business administration activities:

- Create LINE accounts.
- Create and schedule application jobs.

For more information, see [LINE Integration](#) [page 77].

## Scope Item: Permission Marketing (1T1)

### Main Integration Activities

You can perform the following main integration activities for this scope item:

- [Marketing – Campaign Execution E-Mail Integration](#) (SAP\_COM\_0016) integration scenario. For more information, see [Setting Up Service Provider for Emails and Text Messages](#) [page 110].

## Additional Integration Activities

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- *Marketing – Form Integration* (SAP\_COM\_0023) integration scenario. For more information, see [Custom Integration of Forms \[page 92\]](#).
- *Marketing - Form Publication Integration* (SAP\_COM\_0148) integration scenario. For more information, see [Form Publication \[page 105\]](#).
- *Marketing - SAP BTP Landing Page Publication Integration* (SAP\_COM\_1074) integration scenario. For more information, see [Landing Page Design](#).
- *Marketing – SAP BTP Form Integration* (SAP\_COM\_1041) integration scenario. For more information, see [Standard Integration of Forms \[page 102\]](#).

## Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- [Contacts and Profiles](#)
- [Configuration for Permission Marketing](#)
- [Campaigns](#)
- [General Settings](#)
- [Sender Profiles \[page 155\]](#)
- [Segmentation Configuration](#)

## 3.3 Business Scenario: Commerce Marketing

Overview of the *Commerce Marketing* business scenario, its scope items, related main and additional integration activities, as well as configuration and business administration activities.

For information about the business scenario and its corresponding process steps, see [Commerce Marketing](#).

### Scope Item: Product Recommendation (JC3)

#### Main Integration Activities

You can perform the following main integration activities for this scope item:

- [Contacts \[page 412\]](#) (SAP\_COM\_0207) integration service
- [Interactions \[page 615\]](#) (SAP\_COM\_0206) integration service
- [Interaction Contacts \[page 469\]](#) (SAP\_COM\_0207) integration service
- [Corporate Accounts \[page 512\]](#) (SAP\_COM\_0207) integration service
- *Marketing - Recommendations* (SAP\_COM\_1043) integration scenario. For more information, see [Recommendations \(SAP Business Technology Platform\) \[page 923\]](#).
- *Marketing - Recommendations Integration* (SAP\_COM\_0019) integration scenario. For more information, see [Recommendations \[page 939\]](#).



## Additional Integration Activities

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- [Marketing - SAP Commerce Data Integration](#) (SAP\_COM\_0082) integration scenario. For more information, see [Integration with SAP Commerce Cloud \[page 62\]](#).
- [Integration with SAP Product Content Management \[page 314\]](#) (SAP\_COM\_0207) integration service.
- [External Recommendations Integration](#) (SAP\_COM\_0300) integration service. For more information, see [External Recommendations \[page 954\]](#).
- [Recommendations Interaction Data \[page 971\]](#)

## Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- [Recommendation Algorithms](#)
- [Recommendation Data Source Pre-Filters](#)

## Scope Item: Offer Recommendation (1SW)

### Main Integration Activities

You can perform the following main integration activities for this scope item:

- [Marketing - Offer Integration](#) (SAP\_COM\_0020) integration service. For more information, see [Import Offers \[page 973\]](#).

### Additional Integration Activities

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- [Marketing - Offer Discovery Integration](#) (SAP\_COM\_0021) integration service. For more information, see [Discover Offers \[page 1008\]](#).
- [Marketing - SAP Commerce Data Integration](#) (SAP\_COM\_0082) integration scenario. For more information, see [Integration with SAP Commerce Cloud \[page 62\]](#).
- [External Recommendations Integration](#) (SAP\_COM\_0300) integration service. For more information, see [External Recommendations \[page 954\]](#).

## Scope Item: Offer and Coupon Management in Marketing (1HQ)

### Main Integration Activities for Offers with Coupons on Mobile App

You can perform the following main integration activities for the Offers with Coupons on Mobile App variant of this scope item:

- [Contacts \[page 412\]](#) (SAP\_COM\_0207) integration service
- [Interaction Contacts \[page 469\]](#) (SAP\_COM\_0206) integration service
- [Corporate Accounts \[page 512\]](#) (SAP\_COM\_0207) integration service

- *Marketing - Offer Integration* (SAP\_COM\_0020) integration service. For more information, see [Import Offers \[page 973\]](#).
- *Marketing - Mobile Channel in Campaign Management* (SAP\_COM\_0061) integration scenario. For more information, see [Mobile App Integration with Google Firebase \[page 244\]](#).

### **Additional Integration Activities for Offers with Coupons on Mobile App**

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for the Offers with Coupons on Mobile App variant of this scope item:

- *Marketing - Offer Discovery Integration* (SAP\_COM\_0021) integration service. For more information, see [Discover Offers \[page 1008\]](#).
- *Marketing - Marketing Location Integration* (SAP\_COM\_0305)
- *Offer – for wallet use case* (SAP\_COM\_0306)
- *Marketing - Mobile Channel - Inbound Interactions with Campaign Reference* (SAP\_COM\_0169)

### **Main Integration Activities for Offers with External Coupon Service**

You can perform the following main integration activities for the Offers with External Coupon Service variant of this scope item:

- *Marketing - External Coupon Management Service Integration* (SAP\_COM\_0286) integration scenario. For more information, see [Integration with an External Coupon Service System \[page 321\]](#).
- *Marketing - Coupon Integration* (SAP\_COM\_0317) integration scenario. For more information, see [Coupons \[page 1026\]](#).
- *Marketing - Campaign Execution - Shared Mobile Services E-Mail Integration* (SAP\_COM\_1025) integration scenario.

### **Additional Integration Activities for Offers with External Coupon Service**

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for the Offers with External Coupon Service variant of this scope item:

- *Marketing - Offer Discovery Integration* (SAP\_COM\_0021) integration service. For more information, see [Discover Offers \[page 1008\]](#).
- *Marketing - Offer Integration* (SAP\_COM\_0020) integration service. For more information, see [Import Offers \[page 973\]](#).

The following integration scenarios provide APIs for dependent offer objects such as products, target groups, or marketing locations. For example, you can first import products and then assign the imported products to an imported offer.

- *Marketing - Target Group UI Integration* (SAP\_COM\_0205). For more information, see [Target Groups \[page 755\]](#).
- *Marketing - Marketing Location Integration* (SAP\_COM\_0305). For more information, see [Marketing Locations \[page 710\]](#).

### **Main Integration Activities for Offer and Coupons with External Services**

You can perform the following main integration activities for the Offers and Coupons with External Services variant of this scope item:

- *Marketing - External Coupon Management Service Integration* (SAP\_COM\_0286) integration scenario. For more information, see [Integration with an External Coupon Service System \[page 321\]](#).

- [Marketing - Coupon Integration](#) (SAP\_COM\_0317) integration scenario. For more information, see [Coupons \[page 1026\]](#).
- [Marketing - Campaign Execution - Shared Mobile Services E-Mail Integration](#) (SAP\_COM\_1025) integration scenario.
- [Marketing - Interaction UI Integration](#) (SAP\_COM\_0206) integration service.

### Additional Integration Activities for Offer and Coupons with External Services

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for the Offers and Coupons with External Services variant of this scope item:

- [Marketing - Offer Discovery Integration](#) (SAP\_COM\_0021) integration service. For more information, see [Discover Offers \[page 1008\]](#).
- [Marketing - Offer Integration](#) (SAP\_COM\_0020) integration service. For more information, see [Import Offers \[page 973\]](#).

The following integration scenarios provide APIs for dependent offer objects such as products, target groups, or marketing locations. For example, you can first import products and then assign the imported products to an imported offer.

- [Marketing - Target Group UI Integration](#) (SAP\_COM\_0205). For more information, see [Target Groups \[page 755\]](#).
- [Marketing - Marketing Location Integration](#) (SAP\_COM\_0305). For more information, see [Marketing Locations \[page 710\]](#).

## 3.4 Business Scenario: Lead- and Account-Based Marketing


Overview of the [Lead Management and Nurturing](#) business scenario, its scope items, related main and additional integration activities, as well as configuration and business administration activities.

For information about the business scenario and its corresponding process steps, see [Lead- and Account-Based Marketing](#).

### Scope Item: Marketing Lead Management (JCO)

#### Main Integration Activities

For the *Lead Campaign* feature of this scope item, you need to integrate with SAP Cloud for Customer or with SAP Customer Relationship Management (SAP CRM). For more information, see:

- [SAP Cloud for Customer Integration with SAP Marketing Cloud](#) . The following information on integration scenarios is available:
  - [Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)
  - [Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)

- [SAP CRM Integration with SAP Marketing Cloud](#). The following information on integration scenarios is available:
  - [Integration with SAP CRM - Inbound Channel \[page 342\]](#)
  - [Integration with SAP CRM - Outbound Channel \[page 354\]](#)

For the *Call Qualifications* feature of this scope item, you can integrate with SAP Cloud for Customer. For more information, see [SAP Cloud for Customer Integration with SAP Marketing Cloud](#). The following information on integration scenarios is available:

- [Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)
- [Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)

### Additional Integration Activities

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- *UI Integration with SAP Cloud for Customer* (SAP\_COM\_0045) integration scenario, for navigation from contacts to sales system

### Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

- Optionally, you can set up the workflow.  
[Workflow for Business Objects](#)
- In case you have activated more than one communication arrangement, that is, you have configured more than one target system, the Business Add-In (BAI) *Lead Management: Determine Target System Type* is performed. With the *Custom Logic* app, you can implement the BAI. You define the target system type (either SAP\_C4C or SAP\_CRM) depending on different attributes of the contact that is currently in process. The BAI is performed once for each member of the target group. That is, you define in which target system the correspondings leads or activities are created.

## Scope Item: Marketing-Driven Sales Enablement (1SY)

### Main Integration Activities

For the *Activity for Sales* feature of this scope item, you need to integrate with SAP Cloud for Customer or with SAP CRM.

For more information, see:

- [SAP Cloud for Customer Integration with SAP Marketing Cloud](#). The following information on the integration scenarios is available:
  - [Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)
  - [Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)
- [SAP CRM Integration with SAP Marketing Cloud](#). The following information on the integration scenarios is available:
  - [Integration with SAP CRM - Inbound Channel \[page 342\]](#)
  - [Integration with SAP CRM - Outbound Channel \[page 354\]](#)

For the *Sales Insights on Marketing Campaigns* feature of this scope item, you need to integrate with SAP Cloud for Customer.

For more information, see [SAP Cloud for Customer Integration with SAP Marketing Cloud](#). The following information on the integration scenarios is available:

- [Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)
- [Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)

### **Additional Integration Activities**

Depending on your business needs and the business scenarios and scope items you want to use, you might have to perform the following additional integration activities for this scope item:

- [UI Integration with SAP Cloud for Customer \(SAP\\_COM\\_0045\)](#) integration scenario, for navigation from contacts to sales system

### **Configuration and Business Administration Activities**

To set up this scope item, perform the following configuration and business administration activities:

- Optionally, you can set up the workflow.  
[Workflow for Business Objects](#)
- For the feature *Sales Insights on Marketing Campaigns*, you have to activate the campaign transfer.  
[Campaigns: Transfer Campaigns to Sales](#)

## **Scope Item: Marketing Lead Nurturing (2ZM)**

The *Lead Nurture* feature uses email campaigns and lead campaigns (optional). To use this scope item, please refer to the following dependent scope items:

- [Scope Item: Segmentation and Campaign Execution by Email \(JC2\) \[page 11\]](#): To design the stream and run email campaigns.
- [Scope Item: Marketing Lead Management \(JC0\) \[page 19\]](#) (optional): Only if you want to transfer the qualified leads to sales.

## **3.5 Business Scenario: Marketing Planning and Performance**

Overview of the *Marketing Planning and Performance* business scenario, its scope item, related main integration activities, as well as configuration and business administration activities.

For information about the business scenario and its corresponding process steps, see [Marketing Planning and Performance](#).

## Scope Item: Marketing Planning (JC5)

### Main Integration Activities

You can perform the following main integration activities for this scope item:

- *Marketing - Business Data Integration* (SAP\_COM\_0004) integration service. For more information, see [Importing Actual and Committed Spend from SAP ERP \[page 382\]](#).
- *Marketing - Planning Spends Integration* (SAP\_COM\_0018) integration scenario. For more information, see [Integration with SAP ERP for Spend Planning \[page 379\]](#).

### Configuration and Business Administration Activities

To set up this scope item, perform the following configuration and business administration activities:

1. If custom dimensions are going to be used for budget planning, define them using the *Custom Dimensions* configuration application. For more information, see [Custom Dimensions](#).
2. If you defined custom dimensions, define or import custom dimension values in the system from a comma-separated value (CSV) file in the *Custom Dimension Values* business administration application. For more information, see [Custom Dimension Values](#).
3. If brand is going to be used as a dimension for budget planning, define or import brands in the system from a comma-separated value (CSV) file in the *Brands* business administration application. For more information, see [Brands](#).
4. If market is going to be used as a dimension for budget planning, define markets and assign countries to markets using the *Markets* configuration application. For more information, see [Markets](#).
5. If audience is going to be used as a dimension for budget planning, define or import audiences using the *Audiences* business administration application. For more information, see [Audiences](#).
6. If planning is going to be done for different media types, define media types using the *Media Types* configuration application. For more information, see [Media Types](#).
7. Define marketing areas using the *Marketing Areas* configuration application. For more information, see [Marketing Areas](#).
8. Define planning models using the *Planning Models* configuration application. For more information, see [Planning Models](#).
9. If you want to restrict the values that can be used for budget plans in budget planning, define dimension relationships using the *Dimension Relationships* business administration application. For more information, see [Dimension Relationships](#).
10. If you defined custom dimensions or if you want to change the labels for the standard dimensions, define labels for dimensions using the *Labels for Dimensions and Measures* configuration application. For more information, see [Labels for Dimensions and Measures](#).
11. If you want to change the labels for the standard measures, define labels for measures using the *Labels for Dimensions and Measures* configuration application. For more information, see [Labels for Dimensions and Measures](#).
12. Define the actual spend data you want to display in planning using the *Actual Spend and Ad Serving Cost* configuration application. For more information, see [Actual Spend and Ad Serving Cost](#).
13. Define spend types using the *Spend Types* configuration application. For more information, see [Spend Types](#).
14. Activate workflow for business objects using the *Workflow for Business Objects* configuration application. For more information, see [Workflow for Business Objects](#).
15. Define workflows for marketing approvals using the *Manage Workflows* business administration application. For more information, see [Managing Approval Workflows](#).



16. Activate change log for business objects using the [Change Log](#) configuration application. For more information, see [Change Log](#).
17. Activate snapshots for business objects using the [Snapshots for Business Objects](#) configuration application. For more information, see [Snapshots for Business Objects](#).

## 3.6 Business Scenario: Marketing Analytics

Overview of the [Marketing Analytics](#) business scenario, its scope items, and related main integration activities.

For information about the business scenario and its corresponding process steps, see [Marketing Analytics](#).

### Scope Item: Analytics Extensibility and Data Extraction (3SM)

#### Analytics Extensibility

These are the available extensibility options for this scope item. To learn more about the two types of integration and which one you have, see [Setup of SAP Analytics Cloud, Embedded Edition/SAP Analytics Cloud](#).

- [Create Custom Analytics Stories](#)  
For more information, see [Create Custom Stories, SAP Analytics Cloud](#) and [Create Custom Stories, SAP Analytics Embedded Edition](#).
- [Create Custom Operational Reports](#)  
For more information, see [Custom Operational Reports](#).

#### Core Data Services (CDS)-Based Data Extraction

You can perform the following main integration activities for this scope item:

- [Core Data Services-Based Data Extraction from SAP Marketing Cloud to SAP BW Systems](#)  
For more information, see [Setting Up Analytics Extensibility and Data Extraction](#).  
For more information, see [Core Data Services-Based Extraction from SAP Marketing Cloud to SAP BW Systems \[page 24\]](#).
- [Core Data Services-Based Data Extraction from SAP Marketing Cloud to Other SAP and Non-SAP Systems](#)  
For more information, see [Core Data Services-Based Extraction from SAP Marketing Cloud to Other SAP and Non-SAP Systems \[page 25\]](#).  
For more information, see [Cloud Data Integration API \[page 30\]](#).

## 3.6.1 Core Data Services-Based Extraction from SAP Marketing Cloud to SAP BW Systems

This procedure helps you set up the Core Data Services-Based Extraction from SAP Marketing Cloud to an SAP BW system.

### Context

#### *BW Modeling Tools installation*

SAP HANA Studio with BW Modeling Tools must be installed on the local PC to perform the setup. For more information, see [Install BW Modeling Tools](#).

#### *Object List*

Ensure that the following objects have been created in prerequisites setup instructions. There may be different names based on different system environment, you can ask for them from Administrators who performed the setup.

Object	Name
Source System	<system ID>-<client>
DataSource	IMKTTG_DS
DataStore Object	IMKTTGDSO
Data Transfer Process	DTP-Full RSDS IMKTTG_DS <system ID>-<client> ADSO > IMKTTGDSO

### Procedure

1. Verify the source system.
  - a. Open SAP BW/4HANA, and log on to SAP BW/4HANA system.

You successfully logged on, and [SAP Easy Access](#) page is displayed.
  - b. Access the BW Workbench, and run TCode RSA1.

The [Process Chain Display Planning View](#) page is displayed.
  - c. In the [Source System](#) panel, expand [ODP – ABAP CDS Views](#) and find the source system [XXX](#) that you created in set-up instructions or you're required to test in your own scenario.

The source system is displayed.
  - d. Right-click the source system and choose [Check](#).

Source system connection XXX OK is displayed at the bottom of the screen.
2. Verify the data flow.
  - a. Open the SAP HANA studio, and navigate to the SAP BW/4HANA system using the [BW Modeling Perspective](#).

You've successfully opened the SAP BW/4HANA project.

- b. In the *Data Sources* node, expand the *ABAP CDS Views* folder and then expand the source system. Check if data source (for example, IMKTTG\_DS) exists and activated.

The Data Source IMKTTG\_DS is displayed.

- c. In the *BW Repository* node, expand the *NODESNOTCONNECTED* folder and then expand the *DataStore Object (advanced)* folder. Check if DSO (for example, IMKTTGDSO) exists and activated.

The Data Store Object IMKTTGDSO is displayed.

- d. In the *DataStore Object (advanced) > IMKTTGDSO* node, expand the *Data Transfer Process* folder and check if DTP (for example, *DTP-Full RSDS IMKTTG\_DS <system><client> > ADSO IMKTTGDSO*) exists and activated.

The Data Transfer Process *DTP-Full RSDS IMKTTG\_DS <system><client> > ADSO IMKTTGDSO* is displayed.

### 3. Verify the extracted data.

- a. Open the SAP HANA studio, and navigate to the SAP BW/4HANA system using the *BW Modeling Perspective*.

You've successfully opened the SAP BW/4HANA project.

- b. Open DSO *IMKTTGDSO* and navigate to *Properties – DDIC* tab. Choose */BIC/AXXX1* link beside *Active Table*.


The SAP BW/4HANA screen is opened in a new page.

- c. On the *Dictionary: Display Table* screen, choose *Contents*. In the *Select Fields for Selection* screen, check *Target Group* field according to your own scenario and then choose *Execute*.

The Data Browser page (SE16) is displayed.

- d. On the *Data Browser* screen, choose *Number of Entries*. If data exists in source system, the result must be XX.

The extracted data records are displayed.

- If you are interested to figure out which CDS Views are capable of extracting data from SAP Marketing Cloud into the desired target system, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#) 

## 3.6.2 Core Data Services-Based Extraction from SAP Marketing Cloud to Other SAP and Non-SAP Systems

This section provides information on how to extract data from an SAP Marketing Cloud system to other SAP and non-SAP systems using the SAP Cloud Data Integration (CDI) API.

The previous section, [Core Data Services-Based Extraction from SAP Marketing Cloud to SAP BW Systems \[page 24\]](#) explains data extraction from an SAP Marketing Cloud system to SAP BW systems while this section deals with data extraction to other SAP and non-SAP systems.

You can extract data from an SAP Marketing Cloud system to other SAP and non-SAP systems using the communication scenario *SAP\_COM\_0531*.

The CDI API enables data extraction from SAP Marketing Cloud to be consumed via an ODataV4 endpoint. You must implement the ODataV4 client for consuming the CDI API services using one of the following:

- SAP Cloud Integration (SCI) **(Deprecated)**
- SAP Data Intelligence (Data Hub)
- SAP Smart Data Integration (SDI)

### i Note

SAP Cloud Integration (SCI) has been deprecated, because only SAP Smart Data Intelligence (DI) or SAP Data Intelligence (Data Hub) is the standard platform for data integration, whereas SCI is positioned as middleware for process integration on SAP BTP.

### i Note

Consumption of the CDI API services in a non-SAP is supported system only using the above-listed services. For more information, see [Cloud Data Integration API \[page 30\]](#).

## Prerequisites

You have to create a communication arrangement for the communication scenario `SAP_COM_0531`. Creating a communication arrangement is required on SAP Marketing Cloud irrespective of the consuming channel, that is, Data Hub, or SAP SDI.

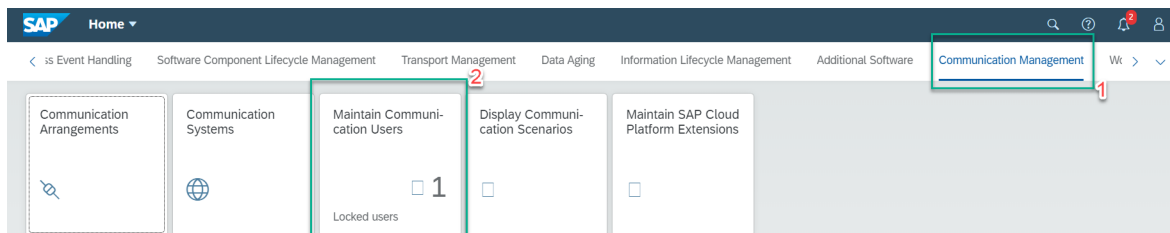
## Set Up the Communication Arrangement on SAP Marketing Cloud

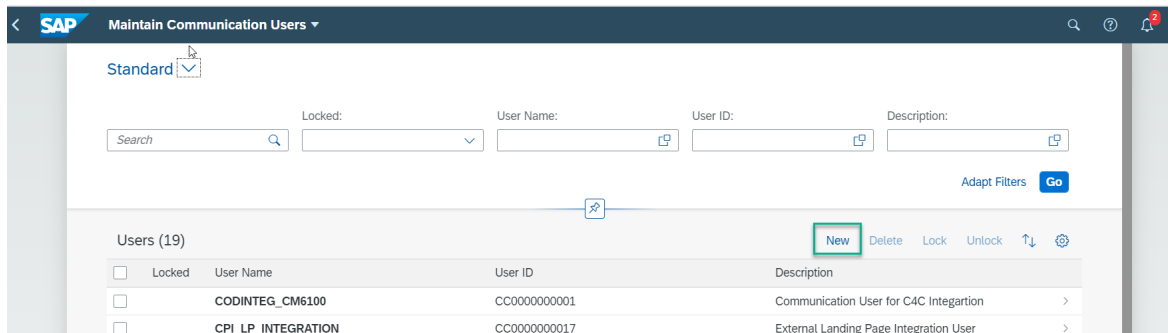
To access the CDI API services, complete the following steps on SAP Marketing Cloud.

### Create a Communication User

A technical user is required to access the CDI API services. This user is a special user used for data extraction purposes.

1. Log in to the SAP Marketing Cloud system as an Administrator.
2. In the SAP Marketing Cloud system, choose the *Maintain Communication Users* app, click *New* to create a new communication user.





3. Enter the *User Name*, *Description*, and *Password* (either enter a password manually or use the proposed password). Click *Create*.

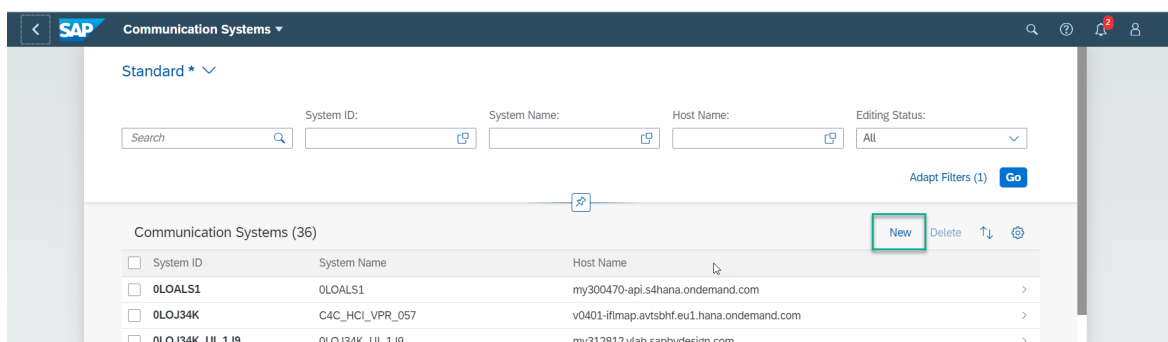
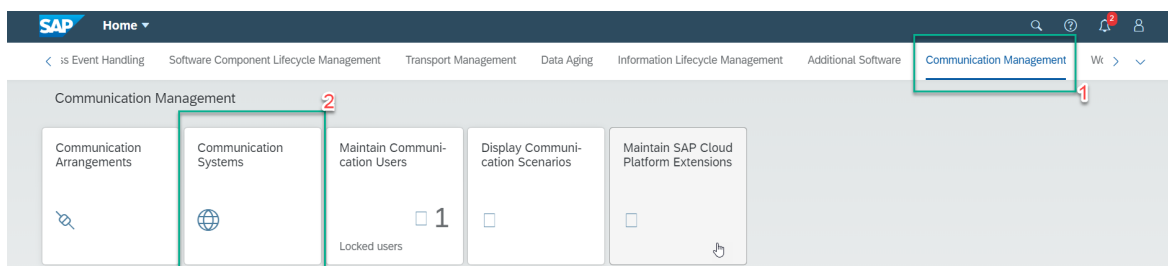
### i Note

The certificate upload isn't mandatory for this scenario.

## Create a Communication System

The connection management needs to know in which system the connection is being set up, hence a communication system has to be created.

1. Log in to the SAP Marketing Cloud system as an Administrator.
2. Open SAP Marketing Cloud in a new browser window. In the *Communication Systems* app, click *New* to create a new communication system.



3. Enter a System ID and System Name for your communication system, and choose *Create*.

**New Communication System**

System ID:\*

System Name:\*

4. On the *Communication System* page, under *Technical Data*, enter the following:
  1. Host Name:  
If host name not required then System ID is sufficient. Host Name is required when the connection needs to be set up with third-party system and its Host ID is required.
  2. Logical System: Not required.
  3. Port: Default port is 443.  
Don't change the port number.
5. Under *Users for Inbound Communication*, choose + button.
6. In the *New Inbound Communication User* pop-up screen, enter the User Name and select User Name and Password option for Authentication Method, and then click *OK*.

**New Inbound Communication User**

User Name:\*

Authentication Method:\*

7. On the *Communication System* page, choose *Save*.

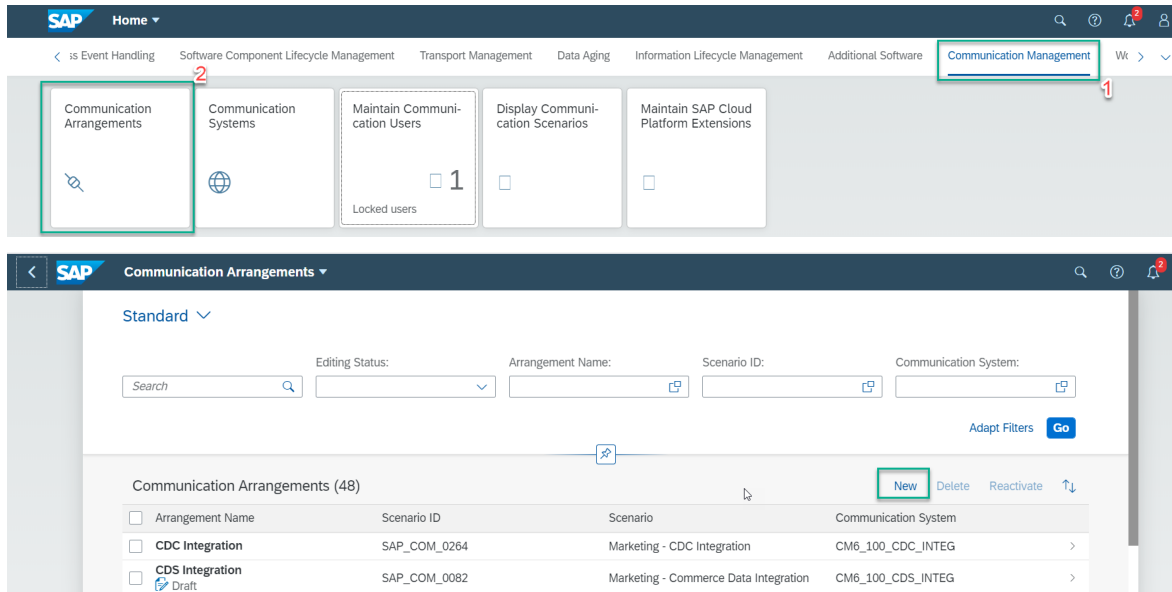
### Create Communication Arrangement

To access the CDI API services, a communication arrangement is required. The communication arrangement generates the service endpoints and assigns the right authorization roles required to access the provider data.

Use the Communication Scenario ID `SAP_COM_0531` to create a communication arrangement. While creating a communication arrangement, the authorizations required to access the CDI API services are granted to the communication user from the communication scenario role `SAP_COM_0531`. With the creation of communication arrangement, the service endpoints are exposed with the communication user and are ready for consumption.

1. Log in to the SAP Marketing Cloud system as an Administrator.
2. In the SAP Marketing Cloud system, choose the *Communication Arrangements* app, click *New* to create a new arrangement.





3. In the *New Communication Arrangement* pop-up screen, enter the scenario **SAP\_COM\_0531** and click *Create*.

### New Communication Arrangement

Scenario: \*

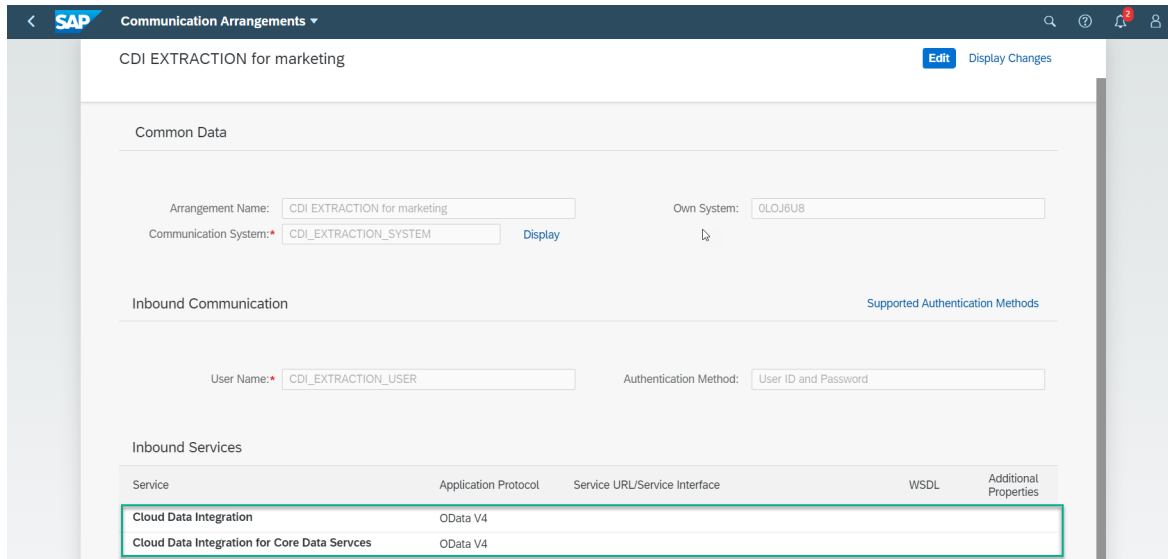
SAP\_COM\_0531

Arrangement Name:

CDI EXTRACTION for marketing

Create Cancel

4. Enter the Communication System that you defined while setting up a Communication System, and click *Save*.  
Once the communication system is saved, the two ODataV4 service groups (Cloud Data Integration (CDI) and Cloud Data Integration for Core Data Services (CDI\_CDS)) are populated on the Communication Arrangements page, under Inbound Services.



### Note

Upon creation of Communication Arrangement service, the service endpoints aren't listed but only OData groups are listed as shown. To access the CDI admin service endpoint, you must fetch the URL in the following way: `<Host>/<Service Path>` where, `<Service Path>` is `/sap/opu/odata4/sap/cdi/default/sap/cdi/0001/`.

This communication arrangement is a customer-managed Communication Arrangement, so the certificate is optional. You must append '-api' to the uri host, to make a successful call from the client system as shown: <https://myXXXXXX-api.s4hana.ondemand.com/sap/opu/odata4/sap/cdi/default/sap/cdi/0001/>. Once the admin service endpoint is accessible, you can fetch the service path for each of the providers with a GET call to the provider EntitySet as shown: `<Host>/sap/opu/odata4/sap/cdi/default/sap/cdi/0001/Providers`.

For more information on the CDI API, see [Cloud Data Integration API \[page 30\]](#).

### Related Information

- For an overview of CDS-based data extraction in SAP Marketing Cloud, see the blog: [CDS-Based Data Extraction - An Overview](#)
- For a general overview of CDS-based data extraction, see [CDS-Based Data Extraction - An Overview](#)
- If you are interested to figure out which CDS Views are capable of extracting data from SAP Marketing Cloud into the desired target system, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#)

## 3.6.2.1 Cloud Data Integration API

This section provides the technical information of the Cloud Data Integration (CDI) API.

To access the CDI admin service endpoint, you must fetch the URL in the following way: `<Host>/<Service Path>` where, `<Host>` is the SAP Marketing Cloud host similar to <https://myXXXX-api.s4hana.ondemand.com> and `<Service Path>` is `/sap/opu/odata4/sap/cdi/default/sap/cdi/0001/`.

Accordingly, the service endpoint URL is: <https://myXXXXXXXX-api.s4hana.ondemand.com/sap/opu/odata4/sap/cdi/default/sap/cdi/0001/>.

The response to the service endpoint call consists of the following entity sets:

```
{ "@odata.context": "$metadata", "value": [ { "name": "Namespaces", "url": "Namespaces" }, { "name": "Providers", "url": "Providers" }, { "name": "Subscriptions", "url": "Subscriptions" } ] }
```

## Entity Sets

The following are the CDI API Entity Sets:

- **Namespaces:** For SAP Marketing Cloud, the ABAP CDS Views are the applicable namespaces.
- **Providers:** Providers are the different CDS Views that have been enabled for extraction. In SAP Marketing Cloud, all the CDS Views that start with the naming convention I\_MKT\_\* or C\_MKT\_\* are the providers.

GET /sap/opu/odata4/sap/cdi/default/sap/cdi/0001/Providers

```
{ "NamespaceID": "ABAP_CDS", "ProviderID": "I_MKT_CONTACTFACETDATA_2", "Description": "Marketing: Contact Facet Data", "ServiceURL": "/sap/opu/odata4/sap/cdi_cds/cdi_cds/sap/i_mkt_contactfacetdata_2/0001/" }
```

GET /sap/opu/odata4/sap/cdi\_cds/cdi\_cds/sap/i\_mkt\_contactfacetdata\_2/0001/

Based on the type of CDS View, the corresponding entity set is displayed, for example the MasterData, Facets, etc.

```
{ "@odata.context": "$metadata", "value": [ { "name": "MasterData", "url": "MasterData" } ] }
```

The provider-specific OData service contains the entity set with data access. The client can access the list of columns accessed by sending a GET request to <serviceRoot>/\$metadata.

GET /sap/opu/odata4/sap/cdi\_cds/cdi\_cds/sap/

i\_mkt\_contactfacetdata\_2/0001/\$metadata provides the service metadata.

- **Subscriptions:** There can be (0: N) subscriptions for a provider. This information is required for delta extraction scenario.

The CDS Views advertise their change-tracking capabilities by annotating entity sets with the `Capabilities.ChangeTracking` term. The client requests the service track changes by specifying track-changes preference on a request in the Prefer header.

Prefer: odata.track-changes

If supported for the request, the service includes a Preference-Applied header in the response containing the track-changes preference and includes a delta link on the last page of results.

A subscription is created implicitly by accessing the provider data with `odata.track-changes` enabled.

The Subscriptions are stopped via a DELETE call to the Subscriptions entity set. It's possible to create a subscription explicitly via a POST call. In this case, the client can set an external ID. The `currentDeltaLink` and the `previousDeltaLink` are calculated by the server.

### Deletion of Subscriptions

Subscriptions can be deleted with a DELETE request specifying the **NamespaceID**, **ProviderID**, and **SubscriptionID** key fields.

### Note

The key fields must be part of the URI when delete operation is performed as shown <Serv\_path>/EntitySet (NamespaceID='ABAP\_CDS', ProviderID='ProviderID#', SubscriptionID='SubscriptionID#'). Since the DELETE request is a Modify operation for EntitySet, the CSRF token is fetched before performing the actual delete call.

## Provider Data Access

- **Data Preview:** A GET call to the EntitySet of a provider URL with top and/or skip fetches the data in preview mode. For example, `<Host>/<Serv_path>/EntitySet?$top=2&$skip=1`.
- **Full Mode:** A GET call to the EntitySet of a provider URL fetches the data in preview mode. For example, `<Host>/<Serv_path>/EntitySet`.  
In the GET `<HOST>/sap/opu/odata4/sap/cdi_cds/cdi_cds/sap/i_mkt_contactfacetdata_2/0001/MasterData` call along with the header information **Prefer: odata.maxpagesize=<pagesize>**, you can set the desired pagesize for pagination. The `@odata.nextLink` contains the link to fetch the next set of entries.
- **Delta Mode:** A GET call to the EntitySet of a provider URL with header fetches the data in Delta mode. The first fetch is delta init and the delta link is provided at the end of payload. Use this link to make subsequent calls to retrieve Deltas only.  
For example, `<Host>/<Serv_path>/EntitySet` with Header `Prefer: odata.track-changes`  
The response is similar to the following with the data ending with delta link as shown:  
`"@odata.deltaLink" : "<Serv_path>/EntitySet?$deltatoken=D_KJKAAPANFEPNVANLRAZBI3KCHM"`  
This action results in implicit subscription creation for provider.

### i Note

Delta mode can be used along with the pagesize attribute. If the specified pagesize is smaller than the data, the response contains data along with a link to the next page and the subsequent requests fetch the data from the next pages. During the last page fetch, the delta link is returned.

## 3.6.3 Models for Core Data Service-Based Extractions

### [CDS Modeling: Campaign and Campaign Performance \[page 33\]](#)

Overview of the data model that illustrates the relationships between the CDS views for campaign and campaign performance.

### [CDS Modeling: Contacts and Profiles \[page 34\]](#)

Models available to show relationships between CDS views for data extraction.

### [CDS Modelling: Marketing Events and Event Participants \[page 37\]](#)

Overview of the data model that illustrates the relationships between the CDS views for marketing events and event participants.

### [CDS Modeling: Marketing Planning \[page 40\]](#)

Models available to show relationships between CDS views for data extraction.

### [CDS Modeling: Scores and Predictive Studio \[page 45\]](#)

Overview of CDS views that are modeled according to the relationships between the different entities of scores.

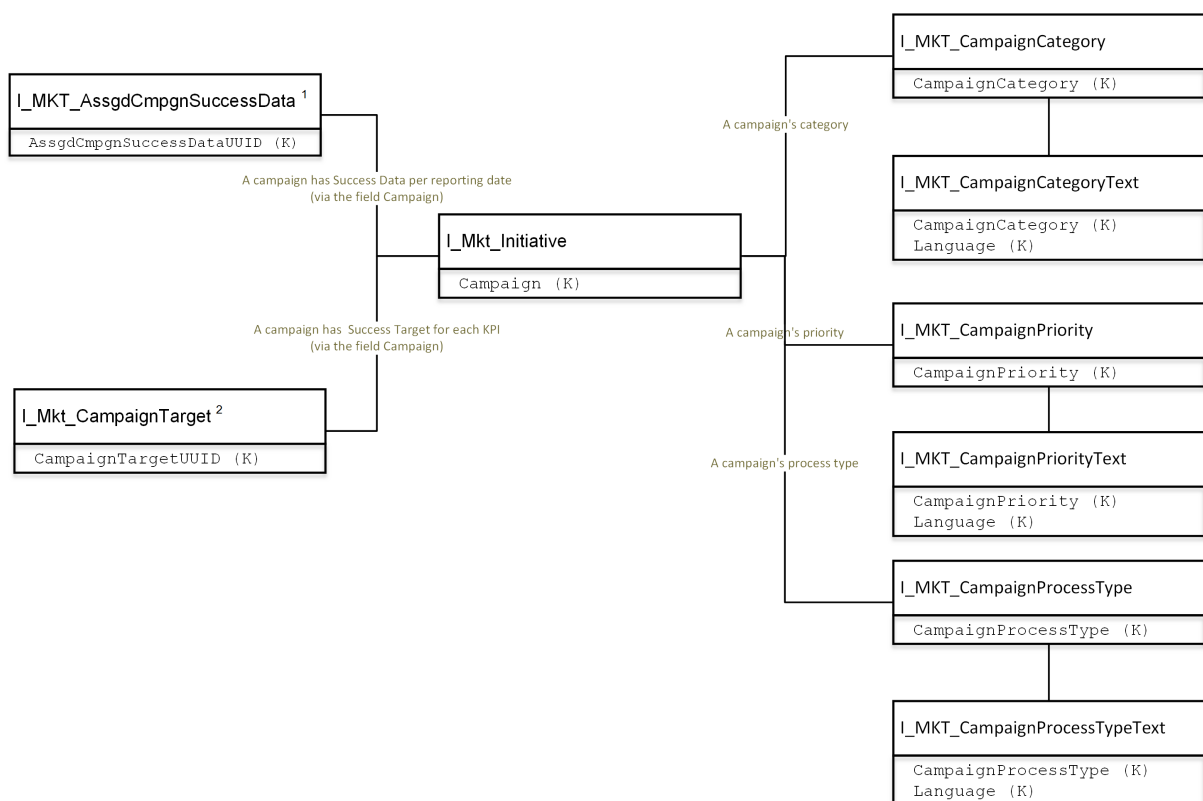
### 3.6.3.1 CDS Modeling: Campaign and Campaign Performance

Overview of the data model that illustrates the relationships between the CDS views for campaign and campaign performance.

#### Prerequisites

For information regarding prerequisites, see [Business Scenario: Marketing Analytics \[page 23\]](#) and open the collapsible section titled *Core Data Services (CDS)-Based Data Extraction*.

#### Technical Details



**Anno-  
tations Description**

1	The I_MKT_AssgdCmpgnSuccessData view contains the actual campaign performance data in the date level. You can aggregate the data into the campaign level and compare with the target defined in the I_MKT_CampaignTarget view. The view also contains additional dimensions, for example, gender, age, and country/region.
2	The I_MKT_CampaignTarget view contains the defined performance target in the campaign level.
(K)	Represents key fields.

**i Note**

The different version of campaign generated by the snapshot functionality of the marketing plan application can be extracted from the I\_MKT\_InitiativeVersion / I\_MKT\_CampaignTargetVersion CDS view. The MktgObjVersHdrUUID field identifies the snapshot version. And the CDS view I\_MKT\_ObjectVersionHeader gives the information on the snapshot version.

The purpose of this diagram is to show how the CDS views for Campaigns and Campaign Performance are linked together to form a model. The complete CDS view definitions, including all of the available fields can be viewed in the [View Browser](#) application.

Released CDS Views that are enabled for data extraction contain the following annotation:

@Analytics.dataextraction.enabled: true.

Released CDS Views that support delta extraction contain the following annotation:

@Analytics.dataextraction.delta.

For more information about the views that are enabled for extraction, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#).

## Change History

New as of 2008 release.

### 3.6.3.2 CDS Modeling: Contacts and Profiles

Models available to show relationships between CDS views for data extraction.

[Interaction Contacts \[page 35\]](#)

Overview of the data model that illustrates the relationships between the CDS views for Interaction Contacts.

[Interactions \[page 36\]](#)

View the data model that visualizes the relationships between the CDS views for Interactions.

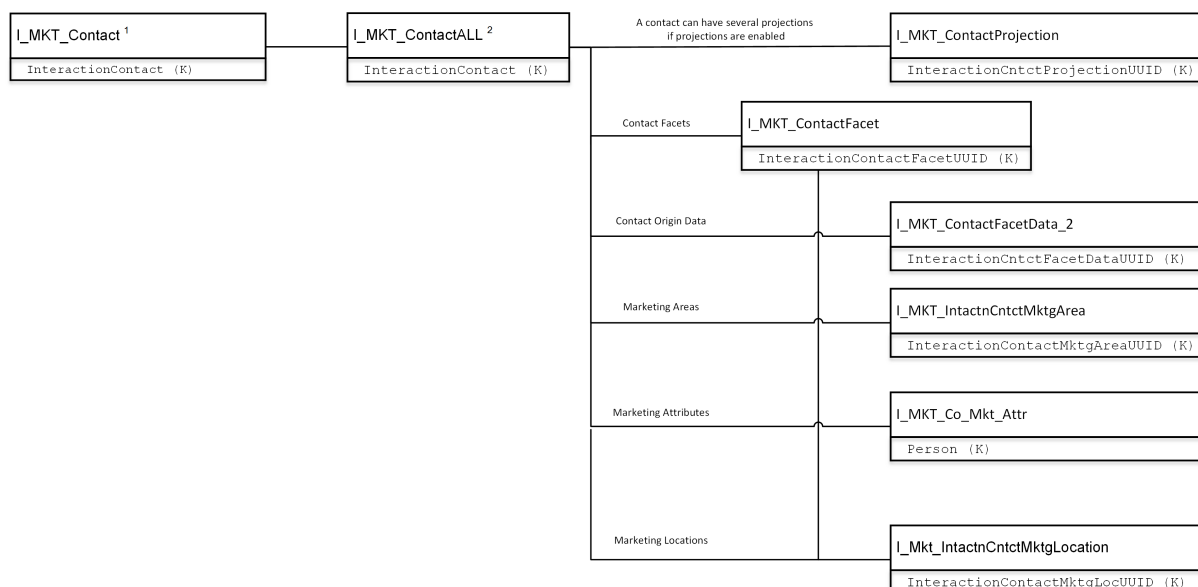
## 3.6.3.2.1 Interaction Contacts

Overview of the data model that illustrates the relationships between the CDS views for Interaction Contacts.

For information about prerequisites, see [Business Scenario: Marketing Analytics \[page 23\]](#) and open the collapsible section titled *Core Data Services (CDS)-Based Data Extraction*.

### Technical Details

Numbers indicate there is additional information below.



Annotations	Description
1	I_MKT_Contact : filters out contacts and container contacts that are either obsolete or flagged for deletion.
2	I_MKT_ContactAll : filters out obsolete contacts and container contacts.
(K)	Represents key fields.

More details regarding complete CDS view definitions with all available fields can be viewed in the [View Browser](#) application.

Released CDS Views that are enabled for data extraction contain the annotation `@Analytics.dataextraction.enabled: true`.

Released CDS Views that support delta extraction contain the annotation `@Analytics.dataextraction.delta`.

For more information regarding which views are enabled for extraction, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#).

## 3.6.3.2.2 Interactions

View the data model that visualizes the relationships between the CDS views for Interactions.

### Prerequisites

For information regarding prerequisites, see [Business Scenario: Marketing Analytics \[page 23\]](#) and open the collapsible section titled *Core Data Services (CDS)-Based Data Extraction*.

### Technical Details

Numbers indicate there is additional information below.





Anno-	Description
-------	-------------

1	I_MKT_Interaction filters out interactions that are either obsolete or flagged for deletion.
---	----------------------------------------------------------------------------------------------

(K)	Represents key fields.
-----	------------------------

The purpose of this diagram is to show how the CDS views for Interactions can be linked together to form a model. More details regarding complete CDS view definitions with all available fields can be viewed in the [View Browser](#) application.

All of the CDS Views in the graphic support delta extraction and contain the annotation @Analytics.dataextraction.delta.

For more information regarding which views are enabled for extraction, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#).

### 3.6.3.3 CDS Modelling: Marketing Events and Event Participants

Overview of the data model that illustrates the relationships between the CDS views for marketing events and event participants.

#### Use

Use these CDS views if you want to extract the definition and values of events and event participants to process them further in an external system. These views support custom fields of events and participants.

#### Prerequisites

For information regarding prerequisites, see [Business Scenario: Marketing Analytics \[page 23\]](#) and open the collapsible section titled *Core Data Services (CDS)-Based Data Extraction*.

#### Technical Details

Numbers indicate that there's additional information provided below the diagram.

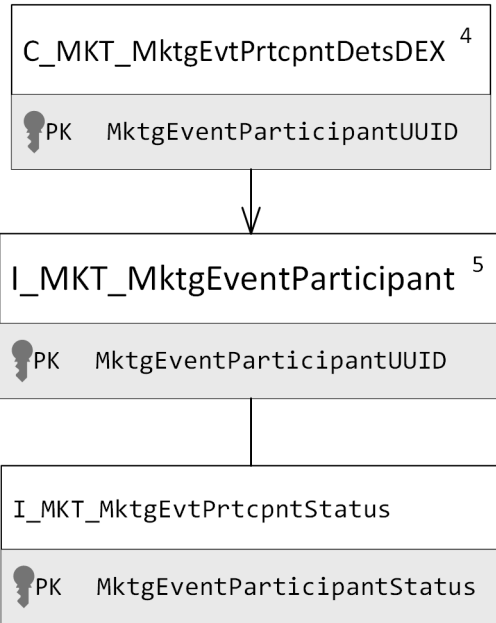
The graphic below illustrates the relationship between the CDS views for marketing events.



**Annotations Description**

1	C_MKT_MktgEventDetailsDex: This data extraction enabled CDS view provides the values of marketing event details such as event ID, name, description, event type, and so on.
2	I_MKT_MktgEvent: This basic view provides the values of events details such as event ID, name, description, event type, and so on.
3	I_MKT_CalendarDate: This basic view represents the event start date. The date is illustrated by the calendar year, month, quarter, and day. This helps to aggregate the extracted event details for the calendar dates.
PK	Represents key fields.

The graphic below illustrates the relationship between the CDS views for marketing event participants.

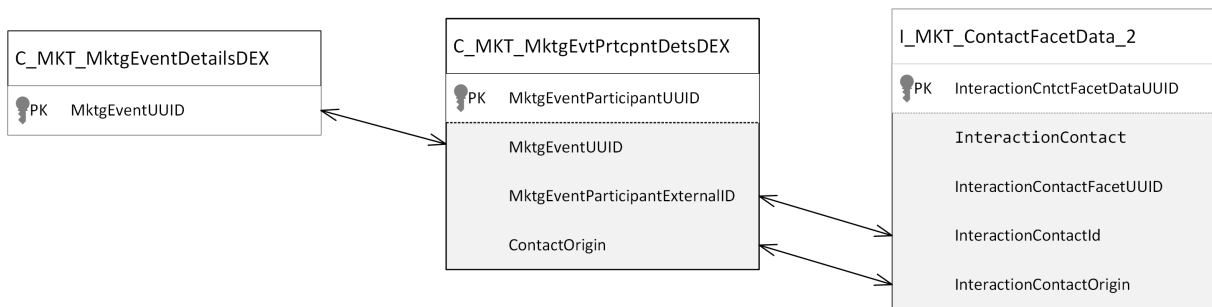


**Anno-**

**tations Description**

Annotations	Description
4	C_MKT_MktgEvtPrtcptntDetsDex: This data extraction enabled CDS view provides details of marketing event participant such as participant ID, contact origin, number of polls answered, number of surveys answered and so on. This view supports delta extraction as well as data of custom fields.
5	I_MKT_MktgEventParticipant: This basic view provides the values of event participants such as participant ID, contact origin, number of polls answered, number of surveys answered etc.
PK	Represents key fields.

The purpose of the graphic below is to show how the CDS views for marketing event and event participants can be linked together to form a model. More details regarding complete CDS view definitions with all available fields can be viewed in the [View Browser](#) application.



Released CDS Views that are enabled for data extraction contain the following annotation:

@Analytics.dataextraction.enabled: true.

Released CDS Views that support delta extraction contain the following annotation:

@Analytics.dataextraction.delta.

For more information about the views that are enabled for extraction, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#).

### 3.6.3.4 CDS Modeling: Marketing Planning

Models available to show relationships between CDS views for data extraction.

[Planned and Actual Spend for Campaigns \[page 40\]](#)

View the data model which visualizes the relationships between the CDS views for Planned and Actual Spend for Campaigns.

[Proposed Spend for Programs \[page 42\]](#)

Overview of the data model that illustrates the relationships between the CDS views for Proposed Spend for Programs.

[Marketing Plan, Planned Budget, and Program Funding \[page 43\]](#)

Overview of the data model that illustrates the relationships between the CDS views for Marketing Plans, Planned Budget, and the Funding Source for Programs.

#### 3.6.3.4.1 Planned and Actual Spend for Campaigns

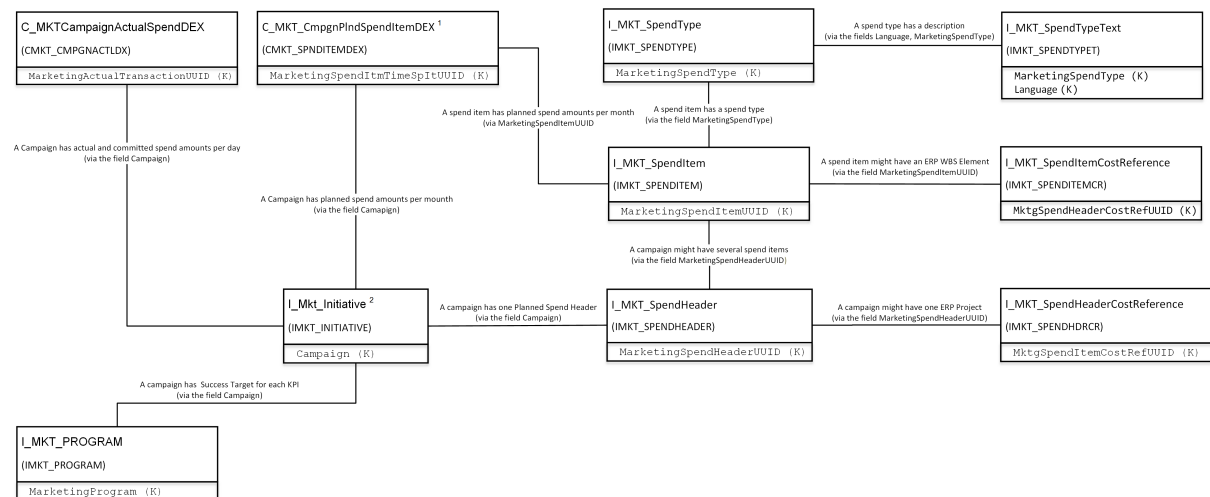
View the data model which visualizes the relationships between the CDS views for Planned and Actual Spend for Campaigns.

### Prerequisites

For information regarding prerequisites, see [Business Scenario: Marketing Analytics \[page 23\]](#) and open the collapsible section titled *Core Data Services (CDS)-Based Data Extraction*.

### Technical Details

Numbers indicate there is additional information below.



**Anno-  
tations Description**

1	C_MKT_CmpgnPlndSpendHeaderDEX: For some campaigns, the planned spend amount might be defined at the campaign level. Therefore, you can find the planned spend amount per month in the following CDS view instead.
2	I_MKT_Initiative: If you use the marketing plan snapshot functionality, you will find the campaign snapshot in the CDS view I_MKT_InitiativeVersion. Therefore, you might need to merge the CDS view I_MKT_Initiative and I_MKT_InitiativeVersion together to have the entire campaign history.
(K)	Represents key fields.

**i Note**

The different versions of campaigns generated by the snapshot functionality of the marketing plan application can be extracted from the same CDS views. The field `MktgObjVersHdrUUID` identifies the snapshot version. The CDS view `I_MKT_ObjectVersionHeader` gives the information on the snapshot version.

The only exception is the CDS view `I_MKT_Initiative`. In this case, the snapshot version can be extracted from the CDS view `I_MKT_InitiativeVersion`.

The purpose of this diagram is to show how the CDS views for Planned and Actual Spend for Campaigns can be linked together to form a model. More details regarding complete CDS view definitions with all available fields can be viewed in the [View Browser](#) application.

Released CDS Views that are enabled for data extraction contain the annotation `@Analytics.dataextraction.enabled: true`.

Released CDS Views that support delta extraction contain the annotation `@Analytics.dataextraction.delta`.

For more information regarding which views are enabled for extraction, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#).

## Change History

New as of 2102 release.

## Related Information

[CDS Modeling: Campaign and Campaign Performance \[page 33\]](#)

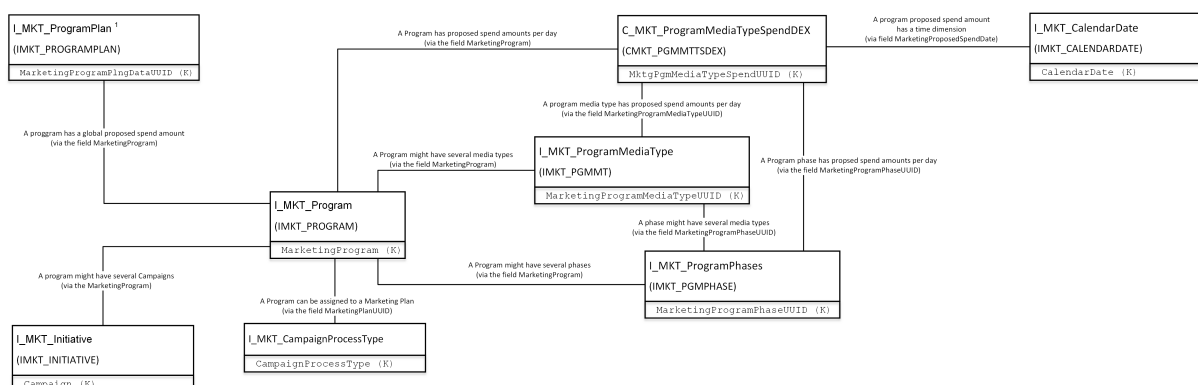
## 3.6.3.4.2 Proposed Spend for Programs

Overview of the data model that illustrates the relationships between the CDS views for Proposed Spend for Programs.

### Prerequisites

For information regarding prerequisites, see [Business Scenario: Marketing Analytics \[page 23\]](#) and open the collapsible section titled *Core Data Services (CDS)-Based Data Extraction*.

### Technical Details



### Annotations Description

- | Annotations | Description                                                                                                                                                                                                                                                        |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1           | The global proposed spend amount for a program is only relevant if no media type has been defined for the program. If a media type is defined for a program, the valid proposed spend amount comes from the CDS view <code>C_MKT_ProgramMediaTypeSpendDEX</code> . |

(K) Represents key fields.

### i Note

The different version of programs generated by the snapshot functionality of the marketing plan application can be extracted from the same CDS views. The field `MktgObjVersHdrUUID` will identify the snapshot version. The CDS view `I_MKT_ObjectVersionHeader` gives the information on the snapshot version.

The purpose of this diagram is to show how the CDS views for Proposed Spend for Programs can be linked together to form a model. More details regarding complete CDS view definitions with all available fields can be viewed in the [View Browser](#) application.

Released CDS Views that are enabled for data extraction contain the annotation `@Analytics.dataextraction.enabled: true`.

Released CDS Views that support delta extraction contain the annotation @Analytics.dataextraction.delta.

For more information regarding which views are enabled for extraction, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#).

## Change History

New as of 2102 release.

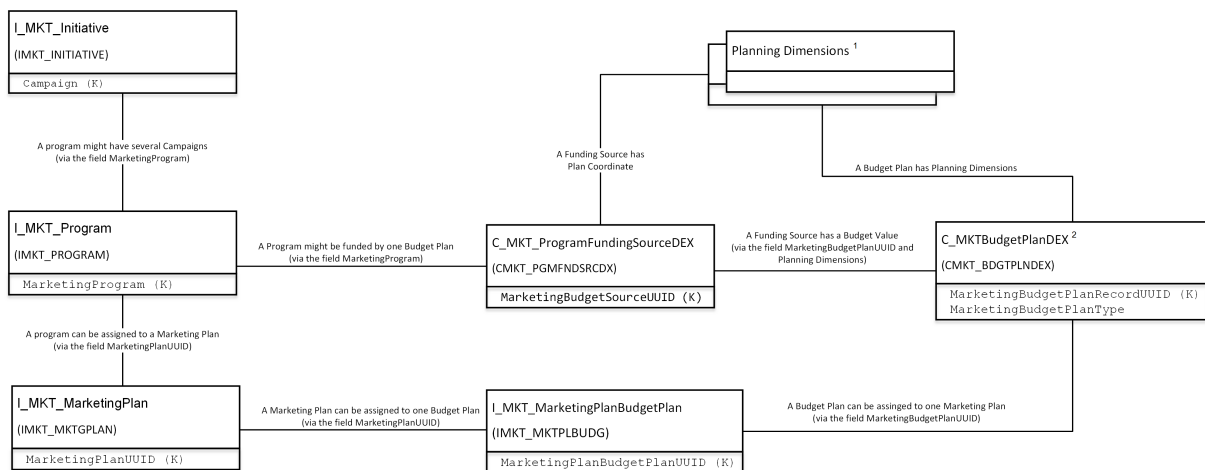
### 3.6.3.4.3 Marketing Plan, Planned Budget, and Program Funding

Overview of the data model that illustrates the relationships between the CDS views for Marketing Plans, Planned Budget, and the Funding Source for Programs.

## Prerequisites

For information regarding prerequisites, see [Business Scenario: Marketing Analytics \[page 23\]](#) and open the collapsible section titled *Core Data Services (CDS)-Based Data Extraction*.

## Technical Details



Annotations	Description
-------------	-------------

- |   |                                                                                                                                                                                                                                                                                                                                                                 |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | <p>The planning dimensions refer to the following CDS views:</p> <ul style="list-style-type: none"><li>• I_MKT_Market (MKT_MARKET)</li><li>• I_MKT_Country (MKT_COUNTRY)</li><li>• I_MKT_Region (MKT_REGION)</li><li>• I_MKT_Brand (MKT_BRAND)</li><li>• I_MKT_Audience (MKT_AUDIENCE)</li><li>• I_MKT_CustomDimension01 to 10 (MKT_CSTMDIMN01 to 10)</li></ul> |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- |   |                                                                                                                                                                                                                                              |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2 | <p>This CDS view contains 2 types of Budget that can be distinguish via the field <code>MarketingBudgetPlanType</code>:</p> <ul style="list-style-type: none"><li>• Public: Budget Plan</li><li>• Allocated: Allocated Budget Plan</li></ul> |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

This CDS view contains delta records and need to be aggregated by the Planning Dimensions and `MarketingBudgetPlanUUID` to get the total budget value.

(K)	Represents key fields.
-----	------------------------

### Note

The different versions of marketing plan, planned budget, and funding source generated by the snapshot functionality of the marketing plan application can be extracted from the same CDS views. The field `MktgObjVersHdrUUID` will identify the snapshot version. The CDS view `I_MKT_ObjectVersionHeader` gives the information on the snapshot version.

The purpose of this diagram is to show how the CDS views for marketing plans, planned budget, and funding sources for programs can be linked together to form a model. More details regarding complete CDS view definitions with all available fields can be viewed in the [View Browser](#) application.

Released CDS Views that are enabled for data extraction contain the annotation `@Analytics.dataextraction.enabled: true`.

Released CDS Views that support delta extraction contain the annotation `@Analytics.dataextraction.delta`.

For more information regarding which views are enabled for extraction, see the following blog at: [Discover CDS View Based Extractors from SAP S/4HANA Cloud](#).

## Change History

New as of 2102 release.



### 3.6.3.5 CDS Modeling: Scores and Predictive Studio

Overview of CDS views that are modeled according to the relationships between the different entities of scores.

#### Use

Use the Score CDS views if you want to extract the definition and the values of rule-based scores and predictive scores and further process them in an external system. If you have score values persisted, you can also extract score values. Non-persisted score values cannot be extracted. The API supports custom scores and SAP delivered scores.

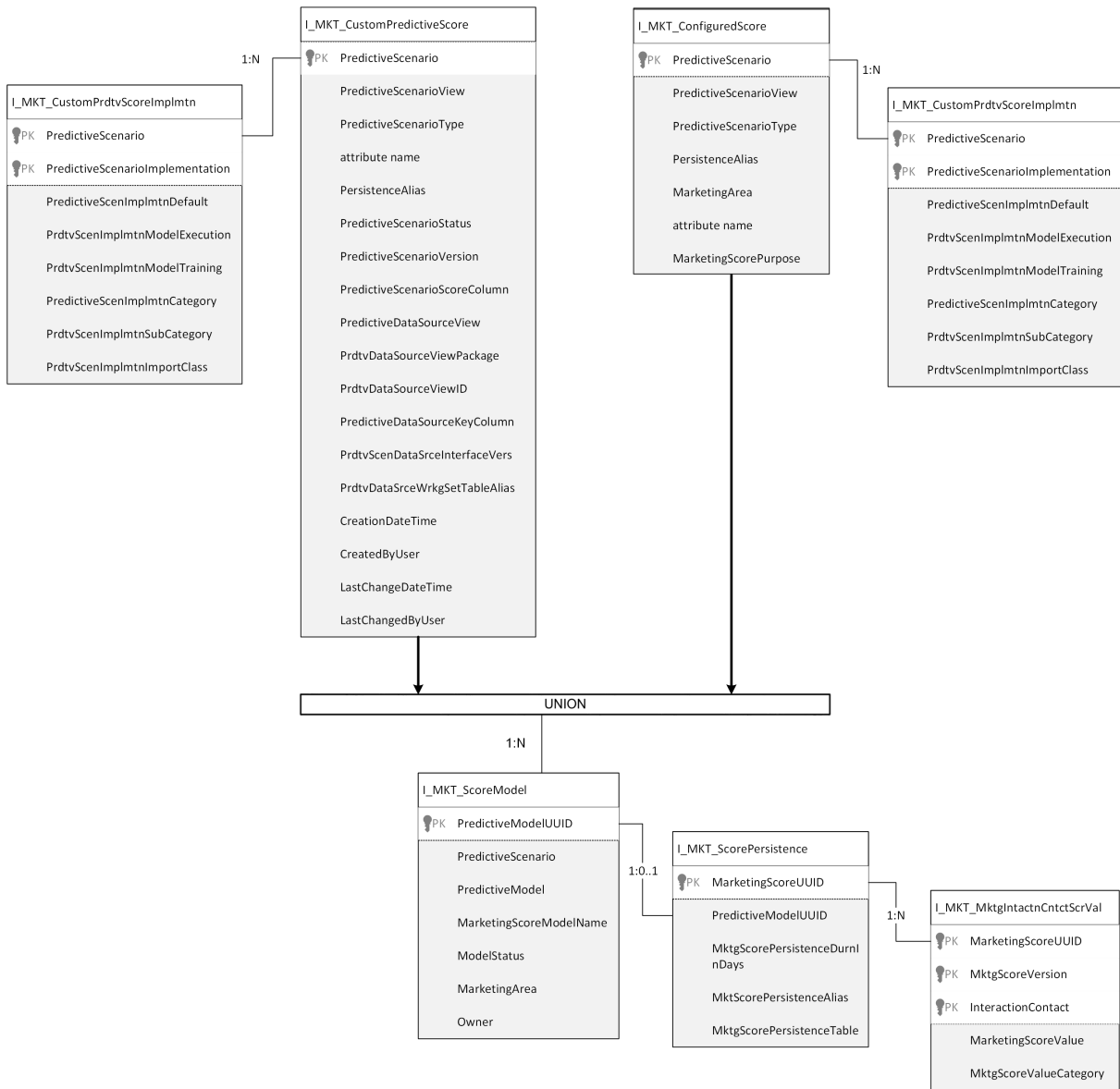
#### Prerequisites

The prerequisite for using these CDS views is that you use one of the following integrations:

- CDI (Cloud Data Integration)
- BW or SAP DI to export data of Score entities into non-marketing systems (better performance than Rest API, and with delta capability)

For more information about prerequisites, see [Business Scenario: Marketing Analytics \[page 23\]](#) and open the collapsible section titled *Core Data Services (CDS)-Based Data Extraction*.

## Technical Details



### Score CDS Views

Entity	CDS View	Description
Score	I_MKT_CustomPredictiveScore	Custom Scores created with Predictive Scenarios application
	I_MKT_ConfiguredScore	<ul style="list-style-type: none"> <li>Scores configured with customizing: SAP delivered scores</li> <li>Score Builder scores</li> </ul>
Score Implementation	I_MKT_CustomPrdtvScoreImplmtn	Implementation methods of custom scores created with Predictive Scenarios application

Entity	CDS View	Description
	I_MKT_ConfiguredScoreImplmnt	Implementation methods of scores that are either configured via customizing or score builder scores
Score Model	I_MKT_ScoreModel	Score Models (Predictive Models)
Score Persistence	I_MKT_ScorePersistence	Score Persistence
Score Value	I_MKT_MktgIntactnCntctScrVal	Persisted Score Values

## Best Practices: Score Extraction and System Performance

### Note

With an ODP source system, we recommend using ABAP runtime, since the extraction is done by ABAP anyway. A transformation in SAP HANA would cause unnecessary effort, since the data must first be persisted. For more information, see [Transformation in the SAP HANA Database](#).

Before extracting score values, please consider how it can influence system performance. Especially consider the following points:

- You can achieve better performance by scheduling regular full loads if
  - Your delta load regularly contains large amounts of data
  - The amount of data in your delta load is similar to or even greater than that of the full load
 One possible scenario where this applies is where you have many scores that allow only one version to be persisted. Since adding the new version will delete the previous version, this doubles the number of records contained in the delta.
- Consider the update frequency of scores. The more daily scores you have, the heavier your delta load.
- Although the CDS-based extraction provides excellent performance, it still adds to the overall resource consumption.
- Additionally, a huge delta load leads to longer extraction runtime.

### Recommendations for Consumption

Please note the following recommendations to ensure optimum system performance:

- **BW**
  - The number of records contained in a delta load **should not exceed 600 million**.
  - The recommended package size is 1 000 000 (one million).
- **OData API**
  - The number of records contained in a delta load **should not exceed 600 million**.
  - The recommended package size is 100 000 (one hundred thousand).
  - The maximum number of parallel processes is 10 (ten).

### Example: How to Calculate Your Data Load Volume

The numbers shown in this example serve only as reference.

Let's assume you have the following number of scores, score versions and interaction contacts in your system:

Score Persistence	Number of Score Versions Kept	Number of Scores Created	Number of Contacts
Daily	10	5	5 million
Weekly	10	5	5 million
Monthly	10	5	5 million

Then the load of score values ranks from 750 million in the initial load to 50 million score values in the daily delta load.

Load	Calculation of Score Values	Number of Score Values	Estimated Duration of Load
Initial Load	(10 versions * 5 scores + 10 versions * 5 scores + 10 versions * 5 scores) * 5 million contacts	750 million	5 hours
Delta Load for Daily Persistence	5 newly created scores * 5 million contacts + 5 deleted scores * 5 million contacts  Since the maximum number of score versions has been reached, adding a new version will delete the oldest score version. The delta load also includes the deleted versions.	50 million	30 minutes
Delta Load for Weekly Persistence	(5 newly created scores * 5 million contacts + 5 deleted scores * 5 million contacts ) + (5 newly created scores * 5 million contacts + 5 deleted scores * 5 million contacts )	100 million  Once a week, the delta load contains 100 million score values. 50 million score values for the daily load and another 50 million score values for the weekly load.	45 minutes
Delta Load for Monthly Persistence	(5 newly created scores * 5 million contacts + 5 deleted scores * 5 million contacts ) + (5 newly created scores * 5 million contacts + 5 deleted scores * 5 million contacts ) + (5 newly created scores * 5 million contacts + 5 deleted scores * 5 million contacts )	150 million  Once a month, the delta load contains 150 million values. 50 million score values for the daily load, 50 million score values for the weekly load and another 50 million score values for the monthly load.	1 hour

# 4 Integration Scenarios

See the section for information about the integration of SAP Marketing Cloud with external systems.

## Overview

Integration Scenarios provide a tight business process integration between SAP Marketing Cloud and other solutions. Technically they are loosely coupled by the SAP Cloud Integration middleware. We distinguish between [Inbound \[page 61\]](#) processes, where an external process step triggers a business process step in SAP Marketing Cloud and [Outbound \[page 108\]](#) processes, where a business process step in SAP Marketing Cloud triggers and external process step.

[Application-Enabling Integrations \[page 267\]](#) feature the integration of complete applications such as SAP Analytics Cloud or Google AdWords.

[Suite-Enabling Integrations \[page 333\]](#) include the integrations that form the SAP Customer Experience portfolio.

## Apps for Setting Up a Typical Integration

In general, you enable the data exchange with an external system in the cloud by setting up the communication with the system. Each integration has its specific parameters, described in the respective integration guide topic, but all are set up in the following apps: [Communication Management](#).

See the following topics for information about how to perform the steps in general:

- [Maintain Communication Users](#)
- [Communication Arrangements](#)
- [How to Create a Communication Arrangement](#)
- [Maintain Communication Systems](#)

## Prerequisites and Details

For prerequisites and details you specify to enable specific integration options, see the topics in this section.

[Overview of Integration Scenarios \(Table\) \[page 50\]](#)

[Inbound \[page 61\]](#)

[Outbound \[page 108\]](#)

[Application-Enabling Integrations \[page 267\]](#)

The section provides information about integration options that enable specific applications of SAP Marketing Cloud, such as geospatial segmentation, or analyzing marketing data based on the analytic capabilities of SAP BusinessObjects Cloud.

[Suite-Enabling Integrations \[page 333\]](#)

This section contains details of integration with applications in the SAP Suite, such as SAP Customer Experience, S/4HANA, CRM, ERP, and includes inbound, outbound, and bidirectional integration.

## 4.1 Overview of Integration Scenarios (Table)



### Inbound

The content of the following table can be sorted and filtered.

Inbound Scenarios

Category	Connected Solution	Content	Communication Scenario/ Comment	Integration Package on API Hub
Market Data and Events	<a href="#">Integration with Data Management Platforms [page 88]</a> (DMP)	Cookie-based user data & interactions	SAP_COM_0 343	Not available on SAP API Hub.
Market Data and Events	List of external providers, event participation <a href="#">Import CSV Using SAP Cloud Integration (Deprecated) [page 108]</a>	Prospects: contact data, corporate accounts	SAP_COM_0 003	<a href="https://api.sap.com/package/SAPHybrisMarketingCloud-filebaseddataload?section=Overview">https://api.sap.com/package/SAPHybrisMarketingCloud-filebaseddataload?section=Overview</a> SAP Marketing Cloud - File-Based Data Load

Category	Connected Solution	Content	Communication Scenario/ Comment	Integration Package on API Hub
Market Data and Events		<ol style="list-style-type: none"> <li>1. Import third party contact, permission and interaction data to SAP Marketing Cloud</li> <li>2. Create external landing pages using an iFlow</li> <li>3. Bring prefill functionality to external tools using the OutboundID</li> </ol>	SAP_COM_0342	<a href="https://api.sap.com/package/ThirdPartyLandingPageDataIntegrationwithSAPMarketingCloud?section=Overview">https://api.sap.com/package/ThirdPartyLandingPageDataIntegrationwithSAPMarketingCloud?section=Overview</a>  External Landing Page Integration
Sales and Service Data	Non-SAP SFA solutions	For example accounts and contacts from Salesforce Sales Cloud	SAP_COM_0017	Not available on SAP API Hub.
Market Data and Events	<a href="#">Landing Pages and Forms [page 91]</a>	Marketing permissions, contact data, and subscriptions	SAP_COM_0023; SAP_COM_0048	<a href="https://api.sap.com/package/ThirdPartyLandingPageDataIntegrationwithSAPMarketingCloud?section=Overview">https://api.sap.com/package/ThirdPartyLandingPageDataIntegrationwithSAPMarketingCloud?section=Overview</a> 
Social media, Web, Commerce, Mobile, IoT	<a href="#">SAP Jam Communities [page 64]</a>	User profiles, created product reviews, read product reviews	SAP_COM_0003; SAP_COM_0004	Currently not on API Hub

Category	Connected Solution	Content	Communication Scenario/ Comment	Integration Package on API Hub
Social media, Web, Commerce, Mobile, IoT	<a href="#">Integration with Google Analytics [page 87]</a>	Web Tracking: Campaign Conversion and Device Category information	SAP_COM_0079	<a href="https://api.sap.com/package/SAPHybrisMarketingGoogleAnalyticsIntegration?section=OVERVIEW">https://api.sap.com/package/SAPHybrisMarketingGoogleAnalyticsIntegration?section=OVERVIEW</a> 
Social media, Web, Commerce, Mobile, IoT	<a href="#">WeChat Integration [page 65]</a>	Posting events (follow, unfollow, send messages)	SAP_COM_0174	Not available on SAP API Hub.
Social media, Web, Commerce, Mobile, IoT	<a href="#">LINE Integration [page 77]</a>	Posting events (follow, unfollow, send messages) of Network Channel	SAP_COM_0174	Not available on SAP API Hub.
Social media, Web, Commerce, Mobile, IoT	Fitbit Activity Tracker	Enrich profile with IoT data, project based see <a href="https://blogs.sap.com/2017/10/13/leverage-fitbit-data-to-run-sap-hybris-marketing-cloud-campaigns-part-1-connecting-fitbit-with-hybris-marketing/">https://blogs.sap.com/2017/10/13/leverage-fitbit-data-to-run-sap-hybris-marketing-cloud-campaigns-part-1-connecting-fitbit-with-hybris-marketing/</a> 		Not available on SAP API Hub.



Category	Connected Solution	Content	Communication Scenario/ Comment	Integration Package on API Hub
Social media, Web, Commerce, Mobile, IoT	<a href="#">Khoros</a> (fka Spredfast / Lithium) activity tracker	Contacts with their social IDs, interactions like raw marketing leads for follow ups		Not available on SAP API Hub.
Survey	<a href="#">Integration with Third-Party Survey Providers [page 107]</a>	Import Survey Metadata and Responses from third-party tools.	SAP_COM_0073	<a href="#">Third Party Survey Data Integration with SAP Marketing Cloud</a>

## Outbound

The content of the following table can be sorted and filtered.

Outbound Scenarios

Category	Connected Solution	Content	Communication Scenario/ Comment	Integration Package on API Hub
Mobile & Social Channel	<a href="#">Mobile App Integration with Google Firebase [page 244]</a>	Mobile Push Notifications	SAP_COM_0061 and SAP_COM_0169	Not available on SAP API Hub.
Mobile & Social Channel	<a href="#">Social Campaigns Using Facebook and Instagram [page 254]</a>	Social Campaigns & Custom Audiences	SAP_COM_0031	Not available on SAP API Hub.
Digital Channel	<a href="#">WeChat Integration [page 258]</a>	WeChat Message Campaigns	SAP_COM_0085	Not available on SAP API Hub.

Category	Connected Solution	Content	Communication Scenario/ Comment	Integration Package on API Hub
Digital Channel	<a href="#">LINE Integration [page 258]</a>	Line Message Campaigns	SAP_COM_0218	Not available on SAP API Hub.
Digital Channel	<a href="#">Integration with Google Ads [page 243]</a>	GoogleAd-words (Paid Search, Display Ads)	SAP_COM_030	<a href="https://api.sap.com/package/SAPHybrisMarketingGoogleAdWordsPaidSearchIntegration?section=Documents">https://api.sap.com/package/SAPHybrisMarketingGoogleAdWordsPaidSearchIntegration?section=Documents</a>
Digital Channel	<a href="#">Integration with Google Campaign Manager [page 259]</a>	Performance Data	SAP_COM_037	<a href="https://api.sap.com/package/SAPMarketingCloudIntegrationwithGoogleDoubleClickCampaignManager?section=Artifacts">https://api.sap.com/package/SAPMarketingCloudIntegrationwithGoogleDoubleClickCampaignManager?section=Artifacts</a>
Email and Text Messages	<a href="#">Setting Up Amazon [page 138]</a> (Email Service Provider)	Outbound Mails incl. Bounce/Success	SAP_COM_016, SAP_COM_039	Not available on SAP API Hub.
Email and Text Messages	<a href="#">Setting Up Sinch [page 112]</a> (Email Service Provider)	Outbound Emails	SAP_COM_040	Not available on SAP API Hub.
Email and Text Messages	<a href="#">Setting Up Sinch [page 112]</a>	SMS	SAP_COM_041	Not available on SAP API Hub.
Email and Text Messages	<a href="#">Setting Up a Generic Email and Text Message Interface [page 116]</a> (Any Email Service Provider)	Outbound Emails	SAP_COM_0234	Not available on SAP API Hub.
Email and Text Messages	<a href="#">Setting Up a Generic Email and Text Message Interface [page 116]</a>	SMS	SAP_COM_0258	Not available on SAP API Hub.

Category	Connected Solution	Content	Communication Scenario/ Comment	Integration Package on API Hub
Extensions	<a href="#">Setting Up External Campaign Execution [page 155]</a>	External Campaign Execution: Transfer Target Group Member Data	SAP_COM_0037	Not available on SAP API Hub.
Extensions	<a href="#">Open Channel Integration [page 194]</a>	Open Campaign Channel: Create customer specific actions & follow-on objects	SAP_COM_0049	Not available on SAP API Hub.

## Options to Integrate with External Outbound Channels for Campaigns

	<a href="#">Setting Up External Campaign Execution [page 155]</a>	<a href="#">Open Channel Integration [page 194]</a>	<a href="#">Generic Email and Text Message Integration [page 120]</a>	<a href="#">Generic Email and Text Message Integration [page 120]</a>
<b>Overview</b>	Replicate full campaigns and target groups member keys regularly (optional) for execution in external platforms/systems, for example, for Ad campaigns (any DMP, DSP).	Transfer individual contacts from target group for execution on external platform, for example, for 3rd party CRM integration (e.g. lead creation) while the campaign flow is controlled within SAP Marketing Cloud.	Integrate any Email Service Provider (ESP), for example, Inxmail.	Integrate any text message service provider
<b>Entity Types</b>	Campaign (optional, incl. parameters)	Campaign Target group members with personalization data	Email messages	Text Messages

	<a href="#">Setting Up External Campaign Execution [page 155]</a>	<a href="#">Open Channel Integration [page 194]</a>	<a href="#">Generic Email and Text Message Integration [page 120]</a>	<a href="#">Generic Email and Text Message Integration [page 120]</a>
<b>Personalizations</b>	Performed in external system	Performed in external system  Transfer of personalized contact attributes through Export Definition	Performed in SAP Marketing Cloud  Transfer of personalized emails	Performed in SAP Marketing Cloud  Transfer of personalized text messages
<b>Implementation Details</b>	Communication Arrangement Setup  SCI (optional)	BAdI Implementation  Communication Arrangement Setu  SCI (optional)	Communication Arrangement Setup  SCI (optional)	Communication Arrangement Setup  SCI (optional)
<b>Standard Campaign Success Handling</b>	SAP Marketing Cloud requests campaign success data in a periodic way (every four hours).  Implementation with API	BAdI implementation  Implementation with API	SAP Marketing Cloud requests bounce information every 10 minutes, when the campaign has been executed within the last 48 hours. After 48 hours, the bounces are collected every four hours  Typically tracked by ESP: bounce, complaint  Tracked by SAP Marketing Cloud: mail sent, mail opened, link click  Implementation with API	SAP Marketing Cloud requests status information every 10 minutes, when the campaign has been executed within the last 48 hours. After 48 hours the bounces are collected every four hours  Typically tracked by message provider: bounce  Tracked by SAP Marketing Cloud: message sent, link click  Implementation with API
<b>Transfer Messages/Campaign Content</b>	No	No	Yes	Yes
<b>Instances per System</b>	Unlimited	Unlimited	Unlimited	Unlimited

## Application-Enabling

The content of the following table can be sorted and filtered.

Application-Enabling scenarios

Connected Solution	Use Scenario	Communication Scenario/ Comment	Integration Package on API Hub
SAP Predictive Analytics - Automated Predictive (fka SAP Infinite Insight)	Consumer Buying Propensity	Not applicable	Not available on SAP API Hub.
<a href="#">SAP Jam Integration for Collaboration [page 320]</a>	Collaboration for Marketing Planning, Campaign Management	SAP_COM_002 6	Not available on SAP API Hub.
<a href="#">Integration with SAP Analytics Cloud (1SO) [page 268]</a>	Self-service BI/ agile analytics in the cloud	SAP_COM_006 5	Not available on SAP API Hub.
<a href="#">Enabling Geospatial Segmentation with here.com [page 317] (fka Nokia Here)</a>	Geospatial Segmentation	Not applicable	Not available on SAP API Hub.
<a href="#">Integration with Baidu Maps for Geospatial Segmentation (Deprecated) [page 318]</a>	Geospatial Segmentation for Chinese geolocation data	SAP_COM_007 5	Not available on SAP API Hub.
<a href="#">Integrate with Content Management Systems or Digital Asset Management Systems [page 306] (OpenText and others)</a>	Enrich email campaign content with creative assets from PCM/DAM solutions	SAP_COM_005 0	<a href="#">SAP Marketing Cloud Integration With Content Management System</a> 




Connected Solution	Use Scenario	Communication Scenario/ Comment	Integration Package on API Hub
<a href="#">Integrate with SAP Document Center [page 312]</a>	Upload images and access images for use via the Content Studio app	SAP_COM_005 0	Not available on SAP API Hub.
<a href="#">Verifying Email Addresses Using a Partner Solution [page 321] (Neverbounce)</a>	Email ID lists for verification and Hard Bounce Prevention, see Blog for CSV based integration and blog for CPI based integration	SAP_COM_004 9	Not available on SAP API Hub.
<a href="#">Integration with an External Coupon Service System [page 321]</a>	Ingest externally generated coupons codes for offers	SAP_COM_028 6	Not available on SAP API Hub.
<a href="#">Marketing Events [page 330]</a>	Fetch Events data from third-party event provider platforms	SAP_COM_047 4 (Deprecated) Marketing - Event Outbound Integration (SAP_COM_054 1) and Marketing - Event Inbound Integration (SAP_COM_037 1)	<a href="#">Third Party Marketing Events Integration with SAP Marketing Cloud</a> 

## Suite-Enabling Integrations

The content of the following table can be sorted and filtered.

Suite-Enabling Integrations

Category	Connected Solution	Content	Communication Scenario/Comment	Integration Package on API Hub
Financial data	<a href="#">Integration with SAP ERP for Spend Planning [page 379]</a>	WBS spend for campaigns including project creation	SAP_COM_0018	<a href="#">SAP Marketing Cloud - SAP ERP Actual and Committed Spend Integration</a>
Industry data	(CAR) <a href="#">SAP Customer Activity Repository retail applications bundle [page 352]</a>	POS data	SAP_COM_0004 / requires SAP CAR 2.0 FP1	Not available on SAP API Hub.
Market Data and Events	<a href="#">Integration with SAP Qualtrics Surveys [page 384]</a>	Import survey response data	SAP_COM_0073	<a href="#">SAP Qualtrics Surveys Integration with SAP Marketing Cloud</a>
Sales Automation	<a href="#">Integration with External Sales Systems - Outbound Channel [page 374]</a>	For example: Lead Handover to Salesforce Sales Cloud	SAP_COM_0017	Partner Offering by <a href="#">Advantco</a>
Sales and Service Data	<a href="#">Integration with SAP CRM - Inbound Channel [page 342]</a>	<ol style="list-style-type: none"> <li>1. Customers, consumers &amp; contacts, marketing attributes mobile &amp; social channel</li> <li>2. Sales business documents such as orders, opportunities etc.</li> <li>3. Permissions (Migration)</li> </ol>	SAP_COM_0017	<a href="https://api.sap.com/package/SAPHybrisMarketingCloud-SAPCRMIntegration?section=Overview">https://api.sap.com/package/SAPHybrisMarketingCloud-SAPCRMIntegration?section=Overview</a>
Sales and Service Data	<a href="#">Integration with SAP Cloud for Customer - Inbound Channel [page 338]</a> <a href="#">Integrating Service Tickets [page 350]</a>	<ol style="list-style-type: none"> <li>1. Individual customers, corporate accounts and contacts</li> <li>2. Leads and opportunities, call activities, appointments/visits,</li> <li>3. Marketing attributes,</li> <li>4. Permissions (one time migration)</li> </ol>	SAP_COM_0017	<a href="https://api.sap.com/package/SAPHybrisCloudforCustomerIntegrationwithSAPHybrisMarketing?section=Overview">https://api.sap.com/package/SAPHybrisCloudforCustomerIntegrationwithSAPHybrisMarketing?section=Overview</a>

Category	Connected Solution	Content	Communication Scenario/Comment	Integration Package on API Hub
Sales and Service Data	<a href="#">Integration with SAP ERP [page 349]</a>	Customers and contacts, consumers, products, sales orders, returns, quotations, EoP for customer/ contacts	SAP_COM_0060	<a href="https://api.sap.com/package/SAPHybrisMarketingCloudSAPERPOrderandBusinessPartnerIntegration?section=Overview">https://api.sap.com/package/SAPHybrisMarketingCloudSAPERPOrderandBusinessPartnerIntegration?section=Overview</a> 
Sales and Service Data	<a href="#">Order Management Data Replication to SAP Marketing Cloud [page 348]</a>	Customers and contacts, consumers, products, EoP for business partner, sales order	SAP_COM_0060	<a href="https://api.sap.com/package/SAPS4HANAEnterpriseManagementOnPremiseIntegrationwithSAPHybrisMarketingCloud?section=Overview">https://api.sap.com/package/SAPS4HANAEnterpriseManagementOnPremiseIntegrationwithSAPHybrisMarketingCloud?section=Overview</a>   <a href="https://api.sap.com/package/SAPHybrisMarketingCloud-SAPS4HANAEnterpriseCloudIntegration?section=Overview">https://api.sap.com/package/SAPHybrisMarketingCloud-SAPS4HANAEnterpriseCloudIntegration?section=Overview</a> 
Social media, Web, Commerce, Mobile, IoT	<a href="#">SAP Customer Data Cloud and SAP Marketing Cloud [page 338]</a> (fka Gigya)	User profiles, marketing attributes & permissions, newsletter subscriptions	SAP_COM_0264	Not available on SAP API Hub.



Category	Connected Solution	Content	Communication Scenario/Comment	Integration Package on API Hub
Social media, Web, Commerce, Mobile, IoT	<a href="#">Integration with SAP Commerce Cloud [page 62]</a>	<ol style="list-style-type: none"> <li>Master data For example, customers and products.</li> <li>Transactional data For example, orders and carts.</li> </ol> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin: 5px 0;"> <p><b>i Note</b></p> <p>Standard Commerce 6.0 solutions use SAP Commerce Data Hub or <a href="#">Expressway</a> to send <b>master</b> and <b>transactional</b> data.</p> </div> <ol style="list-style-type: none"> <li>Product and offer recommendations</li> <li>Personalized content</li> <li>Clickstream data For example, products viewed.</li> </ol>	<p>Marketing - Commerce Data Integration (SAP_COM_0082)</p> <p>Marketing - Recommendation Integration (SAP_COM_0019)</p>	<p><a href="#">SAP Commerce Cloud Integration with SAP Marketing Cloud</a></p> <p><a href="#">SAP Commerce Cloud, Context-Driven Services Integration with SAP Marketing Cloud</a></p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin: 5px 0;"> <p><b>i Note</b></p> <p>Clickstream integration requires a SAP Commerce Cloud, Context-Driven Services license.</p> </div>

## 4.2 Inbound

[Commerce, Social Media, Web, and IoT \[page 62\]](#)

[Landing Pages and Forms \[page 91\]](#)

Integration options for landing pages and forms.

[Survey \[page 107\]](#)

[Extensions \[page 108\]](#)

## 4.2.1 Commerce, Social Media, Web, and IoT

### [Integration with SAP Commerce Cloud \[page 62\]](#)

Support omnichannel activities by integrating SAP Marketing Cloud with SAP Commerce Cloud.

### [SAP Jam Communities \[page 64\]](#)

Provides user profiles and product reviews.

### [WeChat Integration \[page 65\]](#)

With this integration, you can synchronize the followers of your WeChat official accounts as well as the follower interactions to SAP Marketing Cloud. What's more, you can create and carry out WeChat campaigns through SAP Marketing Cloud. Analytical reports about WeChat followers and interactions are available as well.

### [LINE Integration \[page 77\]](#)

With this integration, you can synchronize the followers of your LINE accounts as well as the follower interactions to SAP Marketing Cloud. What's more, you can create and carry out LINE campaigns through SAP Marketing Cloud. Analytical reports about LINE followers and interactions are available as well.

### [Integration with Google Analytics \[page 87\]](#)

Overview of the integration scenario.

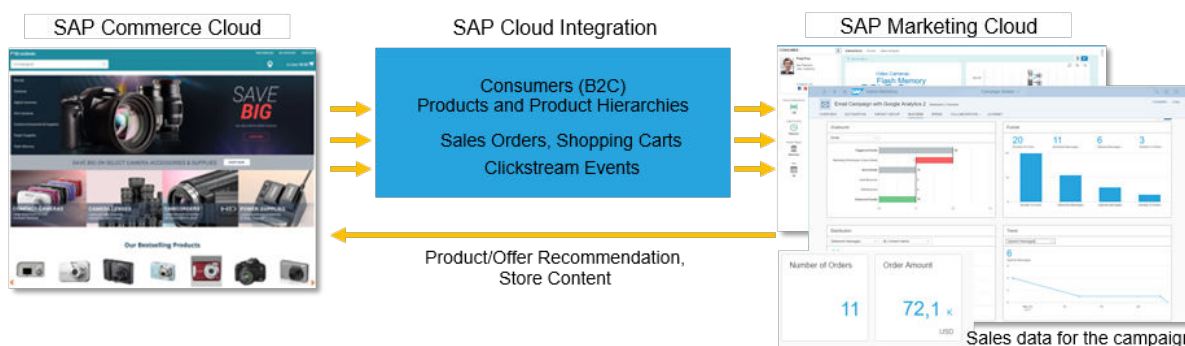
### [Integration with Data Management Platforms \[page 88\]](#)

With this integration scenario, you can capture and replicate Data Management Platform (DMP) IDs from DMP providers, such as Adform. A DMP ID is mapped to a commerce contact ID and stored inside SAP Marketing Cloud.

### 4.2.1.1 Integration with SAP Commerce Cloud

Support omnichannel activities by integrating SAP Marketing Cloud with SAP Commerce Cloud.

This integration leverages the value of commerce stores by personalization of customer engagement on the one hand as it allows you to display SAP Marketing Cloud content such as recommendations and personalized campaign content in SAP Commerce Cloud. On the other hand, the integration drives the customer retention by gathering data from SAP Commerce Cloud for your marketing activities in SAP Marketing Cloud.



You perform installation and configuration activities for the integration entirely in SAP Commerce Cloud.

For information about setting up the integration, see [Configuring SAP Cloud Integration](#) and [Configuring SAP Marketing Cloud](#). Documentation for SAP Commerce Cloud is accessible to SAP customers and partners.

### i Note

You can also integrate with the on-premise version of SAP Commerce.

For all integration options, see [SAP Marketing Cloud Integration Module](#) .

## System Requirements

The system requirements for the integration are as follows:

Feature	SAP Commerce Cloud (in the Public Cloud)	SAP Marketing Cloud
Product recommendations	1905 and higher	1908
Offer recommendations	1905 and higher	1908
Personalized campaign content (segmentation)	1905 and higher	1908
Master and transactional data	1905 and higher	1908
Clickstream data	1905 and higher	1908

For an overview of the features introduced with each release of SAP Commerce Cloud, see the release notes for SAP Commerce Cloud. The release notes are available from the SAP Commerce Cloud product page on the SAP Help Portal at [SAP Commerce](#), under [What's New](#).

To view the documentation for a given release, use the version drop-down list.

## Outbound: Product and Offer Recommendations, Personalized Storefront Content (Segmentation)

Display product and offer recommendations based on the latest information submitted from the customer, such as their cart contents or recently viewed items. SAP Commerce Cloud tracks the visibility and success of your recommendations, and sends this data to SAP Marketing Cloud for analysis.

You can define restrictions to drive the display of individual page components based on real-time campaign lookups in SAP Marketing Cloud. You can also drive the display of entire page variants using marketing data.

For detailed information on the required communication settings, see [Configuring SAP Marketing Cloud](#).

## Inbound: Master Data, Transactional Data, and Clickstream Data

Various types of master and transactional data are collected by SAP Commerce Cloud. For example, customer, product, saved shopping cart, abandoned shopping cart, order, and review data. This data is then sent to SAP Marketing Cloud.

For detailed information on the required communication settings, see [Configuring SAP Marketing Cloud](#).

With clickstream integration, user events on the storefront are aggregated in SAP Commerce Cloud, and then sent to SAP Marketing Cloud for follow-up marketing activities. Various types of user events on the storefront can be sent, for example, product views, category views, and keyword searches.

Clickstream data can be replicated to SAP Marketing Cloud using one of the following integrations:

1. SAP Commerce Cloud, Context-Driven Services using SAP Cloud Integration

For more information, see the following:

- [Customer Interaction Replication from SAP Commerce Cloud, Context-Driven Services](#)
- [SAP Commerce Cloud, Context-Driven Services Integration with SAP Marketing Cloud](#)
- [Context-Driven Services Foundation Integration](#)

### i Note

The system automatically deletes the contacts for anonymous users that only have a `SAP_CDS_PROFILE` origin, if the users have remained anonymous for more than 90 days. This is independent from the licensed number of contacts. For more information on the handling of contacts, see [Contacts](#)

2. Google Analytics using SAP Cloud Integration

For more information, see [Overview of Create Interactions Scenario](#) and [Google Analytics Integration with SAP Marketing Cloud](#).

## 4.2.1.2 SAP Jam Communities

Provides user profiles and product reviews.

The integration option provides user information of consumers or contacts on a commerce store, and product reviews from SAP JAM Communities for the use in SAP Marketing Cloud. If known users read product reviews, interactions are created.

The integration is based on the capabilities of SAP JAM Communities when used in commerce context to facilitate discussions on a product, asking and answering questions on a product, and creating product reviews.

## Integration Setup

To enable the connection with SAP JAM Communities, create the following communication settings:

- Communication user

- Communication system
- Two communication arrangements: One selecting *Communication Scenario* SAP\_COM\_0003, and a second selecting *Communication Scenario* SAP\_COM\_0004.

For information about how to use the communication management apps in general, see [Communication Management](#).

## SAP JAM Communities Data in SAP Marketing Cloud

Find the replicated user data along with profile picture, email address, user ID, user ID of the SAP Commerce shop (if integrated with SAP JAM Communities) in the *Consumer Profile*.

Reviews are treated as a specific interaction type that captures the review score (1-5 stars) in the valuation field, which is also used by sentiments (1 = strong negative to 5 = strong positive). Product data is added to the product node of the interaction. The product name is a tag of the interaction. Interactions of this type run through the SAP HANA text analysis identifying additional tags that can be used for further processing, or interest assignment.

For information about how to set up the integration, see the product page [SAP Jam Collaboration](#) and choose [▶ Administrator Guide ▶ Integrations ▶ Integrate an SAP S/4HANA application ▶](#).

### 4.2.1.3 WeChat Integration

With this integration, you can synchronize the followers of your WeChat official accounts as well as the follower interactions to SAP Marketing Cloud. What's more, you can create and carry out WeChat campaigns through SAP Marketing Cloud. Analytical reports about WeChat followers and interactions are available as well.

#### i Note

The WeChat integration supports **WeChat subscription accounts and WeChat service accounts only**.

The following is a detailed list of the business functions that come with the WeChat integration:

- Synchronization of WeChat followers and interactions to SAP Marketing Cloud
- Creation and execution of WeChat campaigns through SAP Marketing Cloud
- Analytical reports about the acquisition of WeChat followers, including reports predefined by SAP and custom reports that you can build with a CDS view
- Analytical reports about WeChat interactions, including reports predefined by SAP and custom reports that you can build with a CDS view

For information about setting up and administering the WeChat integration, see the following documents:

- [Setting Up the WeChat Integration \[page 66\]](#)
- [Administering the WeChat Integration \[page 73\]](#)

For extensibility options, see [Extensibility \[page 76\]](#).

For the descriptions of the business functions, see the following documents:

- [Followers of Digital Accounts](#)
- [Attributes Related to Followers of Digital Accounts](#)
- [WeChat Campaigns](#)

### 4.2.1.3.1 Setting Up the WeChat Integration

Set up the connection between SAP Marketing Cloud and your WeChat official account.

#### Prerequisites

You have registered an official account through the WeChat Official Account Admin Platform.

#### Procedure

1. Import the WeChat certificate.

For more information, see [Importing the WeChat Certificate \[page 67\]](#).

2. Create communication configurations for the inbound communication and outbound communication, respectively.

For more information, see [Creating Communication Configurations \[page 68\]](#).

3. Create your official account.

For more information, see [Creating a WeChat Official Account \[page 72\]](#).

4. Create and schedule application jobs.

You must create a job based on the template *Digital Accounts: Process Inbound Messages*, which is required for the automatic synchronization of followers and interactions to SAP Marketing Cloud.

If your WeChat official account already had followers before the WeChat integration goes live, you must create a job based on the template *Digital Accounts: Synchronize WeChat Users* to synchronize the existing followers to SAP Marketing Cloud.

There are other jobs that are required for specific functions only. For more information, see [Creating and Scheduling Application Jobs \[page 73\]](#).

#### Results

The system synchronizes followers and follower interactions from the WeChat server to SAP Marketing Cloud automatically. Depending on the application jobs you have run, the system synchronizes other types of data (for example, campaign content) to SAP Marketing Cloud.

## 4.2.1.3.1.1 Importing the WeChat Certificate

In the standard delivery, the system gets and posts WeChat data directly through the WeChat server. If you adopt this approach, import the WeChat certificate so that SAP Marketing Cloud will be trusted by the WeChat server. If you have your own logic for getting and posting WeChat data through a different server, which uses the HTTPS communication protocol, then import the certificate of that server instead.

### Prerequisites

A business role that contains the **Security** (SAP\_CORE\_BC\_SEC) business catalog is required. You can use the standard business role **Administrator** (SAP\_BR\_ADMINISTRATOR), which contains the **Security** business catalog and other administration-related catalogs. Alternatively, you can create custom business roles using the *Maintain Business Roles* app.

### Procedure

1. Go to <https://open.weixin.qq.com/> .
2. Locate the WeChat certificate and export it to a file.
3. Log into SAP Fiori launchpad with a business role that contains the **Security** (SAP\_CORE\_BC\_SEC) business catalog.
4. Open the *Maintain Certificate Trust List* app.
5. Choose + (*Add*).

The *Upload Certificate* window appears.

6. Upload the WeChat certificate file.

### Next Steps

[Creating Communication Configurations \[page 68\]](#)

## 4.2.1.3.1.2 Creating Communication Configurations

Create the configurations required for the communication between SAP Marketing Cloud and the WeChat server.

### Prerequisites

A business role that contains the **Communication Management** (SAP\_CORE\_BC\_COM) business catalog is required. You can use the standard business role **Administrator** (SAP\_BR\_ADMINISTRATOR), which contains the **Communication Management** business catalog and other administration-related catalogs. Alternatively, you can create custom business roles using the *Maintain Business Roles* app.

### Context

The WeChat integration involves communication in both the inbound and outbound directions. In the inbound communication, the WeChat server pushes to a customer-defined servlet WeChat events as well as messages that followers send to WeChat official accounts. Then the servlet forwards the WeChat events and messages to SAP Marketing Cloud by calling the private ICF service <host name>/sap/cuan/ntwrk/. The outbound communication involves synchronizing data (for example, basic follower information and campaign content) from the WeChat server and executing WeChat campaigns through SAP Marketing Cloud.

The inbound communication and outbound communication require their respective communication system/communication arrangement pair. In addition, you need to create a communication user for the inbound communication. When calling the private ICF service to forward the WeChat events and messages to SAP Marketing Cloud, your servlet must authenticate itself with this communication user first.

The following communication scenarios are relevant to the WeChat integration:

- Marketing – Network Channel Events Integration (SAP\_COM\_0174)
- Marketing – Campaign Execution – WeChat Integration (SAP\_COM\_0085)

For detailed instructions, see [Configuring the Inbound Communication \[page 69\]](#) and [Configuring the Outbound Communication \[page 70\]](#).

For general information about communication management, see [Communication Management](#).



## 4.2.1.3.1.2.1 Configuring the Inbound Communication

Create the communication user, communication system, and communication arrangement required for the inbound communication.

### Customer Implementation

You must define a servlet for the inbound communication. This servlet receives the WeChat events and messages pushed by the WeChat server and then forwards them to SAP Marketing Cloud by calling the private ICF service `<host name>/sap/cuan/ntwrk/`. For more information, see the blog [Inbound Connection from WeChat or LINE to SAP Marketing Cloud](#).

### Creating the Communication User

Proceed as follows:

1. Log into SAP Fiori launchpad with a business role that contains the **Communication Management** (SAP\_CORE\_BC\_COM) business catalog.
2. Open the *Maintain Communication Users* app.
3. Choose *New*.  
The *Create Communication User* dialog box appears.
4. Fill in the following fields:
  - *User Name* and *Description* (for example, WECHAT\_EVENT and WeChat Event)
  - *Password*
5. Save the communication user.  
A communication user ID is generated automatically.

#### i Note

When calling the private ICF service, your servlet should authenticate itself with the **communication user ID** instead of the user name.

Do not exit SAP Fiori launchpad.

### Creating the Communication System

This communication system is a dummy one. The purpose of it is to bind the communication user that you created earlier with the communication arrangement that you will create later.

To create the communication system, proceed as follows:

1. Open the *Communication Systems* app.
2. Choose *New*.

The *New Communication System* dialog box appears.

3. Enter a system ID and its name, for example, WECHAT\_EVENT and WeChat Event. Choose *Create*.  
The editing screen for the communication system appears.
4. A host is irrelevant to the inbound communication. Enter **dummy** in the *Host Name* field to assign a dummy host.
5. Assign the communication user created earlier to this communication system, as follows:
  1. In the *User for Inbound Communication* section, choose + (Add).  
The *New Inbound Communication User* dialog box appears.
  2. Enter the user created earlier and select the authentication method **User Name and Password**.
6. Save and activate the communication system.  
Do not exit SAP Fiori launchpad.

## Creating the Communication Arrangement

Proceed as follows:

1. Open the *Communication Arrangements* app.
2. Choose *New*.  
The *New Communication Arrangement* dialog box appears.
3. Enter scenario SAP\_COM\_0174 and an arrangement name. Choose *Create*.  
The editing screen for the communication arrangement appears.
4. In the *Communication System* field, enter the communication system created earlier.
5. Save and activate the communication arrangement.

## Next Steps

[Configuring the Outbound Communication \[page 70\]](#)

### 4.2.1.3.1.2.2 Configuring the Outbound Communication

Create the communication system and communication arrangement required for the outbound communication.

## Creating the Communication System

Proceed as follows:

1. Log into SAP Fiori launchpad with a business role that contains the **Communication Management** (SAP\_CORE\_BC\_COM) business catalog.

2. Open the *Communication Systems* app.
3. Choose *New*.  
The *New Communication System* dialog box appears.
4. Enter a system ID and its name, for example, WECHAT\_API and WeChat API. Choose *Create*.  
The editing screen for the communication system appears.
5. In the *Host Name* field in the *Technical Data* section, enter `api.weixin.qq.com`, which is the host name of the WeChat server. Choose *Save*.

#### **i** Note

If you have your own logic for getting and posting WeChat data through another server, then enter the host name of that server instead. For more information, see [Extensibility \[page 76\]](#).

6. Set the authentication method to **None**, as follows:
  1. In the *User for Outbound Communication* section, choose + (Add).  
The *New Outbound User* dialog box appears.
  2. Select the authentication method **None**. Choose *Create*.
7. Save and activate the communication system.  
Do not exit SAP Fiori launchpad.

## Creating the Communication Arrangement

Proceed as follows:

1. Open the *Communication Arrangements* app.
2. Choose *New*.  
The *New Communication Arrangement* dialog box appears.
3. Enter scenario `SAP_COM_0085` and an arrangement name. Choose *Create*.  
The editing screen for the communication arrangement appears.
4. In the *Communication System* field, enter the communication system that you have created.
5. Activate all the outbound services by selecting the *Active* checkboxes.
6. Save and activate the communication arrangement.

## Next Steps

[Creating a WeChat Official Account \[page 72\]](#)

## 4.2.1.3.1.3 Creating a WeChat Official Account

Create a WeChat official account in SAP Marketing Cloud.

### Prerequisites

A business role that contains the **Marketing - Data** (SAP\_CEC\_BC\_MKT\_PRD\_PC) business catalog is required. You can use the standard business role **Marketing Expert** (SAP\_BR\_MARKETING\_EXPERT), which contains the **Marketing - Data** business catalog and other marketing-related catalogs. Alternatively, you can create custom business roles using the *Maintain Business Roles* app.

### Procedure

1. Log into SAP Fiori launchpad with a business role that contains the **Marketing - Master Data** (SAP\_CEC\_BC\_MKT\_PRD\_PC) business catalog.
2. Open the *Digital Accounts* app.
3. On the landing page, choose *Create Digital Account*.  
The *Select Digital Account Type* dialog box appears.
4. Select *WeChat Official Account*.  
The editing screen for the official account appears.
5. Select the marketing area and enter the required information. Choose *Create*.  
The required information includes:
  - ID and name of the official account that are registered through the WeChat Official Account Admin Platform
  - Handshake token  
You create your own handshake token. The token entered here must be the same as the one specified on the *Basic Configurations* page on the WeChat Official Account Admin Platform.
  - Original ID
  - Credentials for the official account, including the following:
    - App ID
    - App Secret  
You can find these credentials from the *Basic Configurations* page on the WeChat Official Account Admin Platform.
6. On the WeChat Official Account Admin Platform, enter your handshake token and the URI of your servlet.

### Results

The official account appears with the *Active* status on the landing page of the *Digital Accounts* app. By default, the system sets the image assigned to the *WeChat* communication medium in the *Manage Images* app as the

profile picture of the official account. Clicking the account displays various tabs that contain different kinds of information about the official account. The *Information* tab contains basic information about the official account and the credentials. You can change the profile picture, credentials, and so on, by clicking the *Edit* button.

## Next Steps

[Creating and Scheduling Application Jobs \[page 73\]](#)

### 4.2.1.3.2 Administering the WeChat Integration

Learn about the system administration activities relevant to the WeChat integration.

[Creating and Scheduling Application Jobs \[page 73\]](#)

Learn about the application job templates relevant to the WeChat integration. Some are required for the WeChat integration in general, while others are required for specific functions.

[Activating, Deactivating, and Restricting a WeChat Official Account \[page 75\]](#)

You can set a WeChat official account to any of these statuses in the *Digital Accounts* app: *Active*, *Inactive*, and *Restricted*.

[Overview of Business Catalogs Required for Different Business Functions \[page 76\]](#)

Different business functions require different business catalogs. Learn about the business catalogs that are required for the functions related to the WeChat integration and assign business roles to business users appropriately.

#### 4.2.1.3.2.1 Creating and Scheduling Application Jobs

Learn about the application job templates relevant to the WeChat integration. Some are required for the WeChat integration in general, while others are required for specific functions.

### Overview of Related Application Job Templates

Application Job Template	Description
<a href="#">Digital Accounts: Process Inbound Messages</a>	A job created using this template creates contacts and interactions from digital accounts, such as WeChat official accounts, in SAP Marketing Cloud. For more information, see <a href="#">Digital Accounts: Process Inbound Messages</a> .

Application Job Template	Description
<a href="#">Digital Accounts: Synchronize WeChat Users</a>	You have set up the WeChat integration and thus the followers of a WeChat official account can be synchronized to SAP Marketing Cloud automatically. However, there are certain situations where you must synchronize WeChat followers by running a job that is created with this template. For more information, see <a href="#">Digital Accounts: Synchronize WeChat Users</a> .
<a href="#">Digital Accounts: Synchronize Campaign Content from WeChat</a>	You must create a job using this template if you want to create and carry out WeChat campaigns through SAP Marketing Cloud. Business users maintain campaign content on the WeChat Official Account Admin Platform. The application job synchronizes the campaign content from the WeChat Official Account Admin Platform to SAP Marketing Cloud. For more information, see <a href="#">Digital Accounts: Synchronize Campaign Content from WeChat</a> .

## Checking the Application Log

You can find a log of all these application jobs centrally from the [Application Logs](#) app. The filters that you can use for the application jobs are as follows:

Application Job	Filter
<a href="#">Digital Accounts: Process Inbound Messages</a>	Category CUAN, subcategory CUAN_NTWRK
<a href="#">Digital Accounts: Synchronize WeChat Users</a>	Category CUAN, subcategory CUAN_WECHAT
<a href="#">Digital Accounts: Synchronize Campaign Content from WeChat</a>	

Alternatively, you can find the log of a particular application job directly from the [Marketing Application Jobs](#) app. From the application job list, click the **i** icon next to an application job.

## Required Business Role

A business role that contains the **Marketing - Business Administration** (SAP\_CEC\_BC\_MKT\_ADM\_PC) business catalog is required for scheduling application jobs and checking logs. You can use the standard business role **Administrator - Marketing** (SAP\_BR\_ADMINISTRATOR\_MKT), which already contains this business catalog. Alternatively, you can create custom business roles using the [Maintain Business Roles](#) app.

## 4.2.1.3.2.2 Activating, Deactivating, and Restricting a WeChat Official Account

You can set a WeChat official account to any of these statuses in the *Digital Accounts* app: *Active*, *Inactive*, and *Restricted*.

### Activating a WeChat Official Account

To use the full functionality of the WeChat integration, you must set an official account to *Active* status. When completing creating an official account, the status of the official account is set to *Active* automatically.

To set an official account from *Restricted* status to *Active* status, choose the *Activate* button.

To set an official account from *Inactive* status to *Active* status, choose the *Switch to Restricted Mode* button and then the *Activate* button.

### Deactivating a WeChat Official Account

If you do not want to connect to an official account, for example, because the official account is no longer in use, deactivate it by choosing the *Deactivate* button.

### Restricting a WeChat Official Account

When an official account is restricted, the outbound connection to the official account stops working. Therefore, the following functions become unavailable:

- Execution of WeChat campaigns through SAP Marketing Cloud
- Synchronization of basic information about new followers from WeChat, such as nickname and gender
- Synchronization of WeChat followers using an application job created based on the job template *Digital Accounts: Synchronize WeChat Users*
- Synchronization of campaign content from WeChat using an application job created based on the job template *Digital Accounts: Synchronize Campaign Content from WeChat*

When an official account is restricted, there is no impact on the inbound connection from an official account. The synchronization of interactions between followers and the official account still works and the messages that followers send to the official account are still synchronized.

To set an official account to *Restricted* status, choose the *Switch to Restricted Mode* button.

### 4.2.1.3.2.3 Overview of Business Catalogs Required for Different Business Functions

Different business functions require different business catalogs. Learn about the business catalogs that are required for the functions related to the WeChat integration and assign business roles to business users appropriately.

The following table lists the business catalogs that are required for different functions:

Business Function	Required Business Catalog	Standard Business Role That Can Be Used
Functions Available with the <i>Digital Accounts</i> App	Marketing - Data (SAP_CEC_BC_MKT_PRD_PC)	<b>Marketing Expert</b> (SAP_BR_MARKETING_EXPERT)
Contact Profiles of WeChat Followers	Marketing - Contacts and Profiles Base (SAP_CEC_BC_MKT_DMB_PC)  Marketing - Contacts and Profiles Standard (SAP_CEC_BC_MKT_DMS_PC)	
Segmentation	Marketing – Segmentation (SAP_CEC_BC_MKT_SEG_PC)	
WeChat Campaigns	Marketing - Campaign Management (SAP_CEC_BC_MKT_CPM1_PC)	
Custom analytical reports in the <i>Query Browser</i> app	Marketing - Data (SAP_CEC_BC_MKT_PRD_PC)  Query Browser (SAP_CA_BC_VDM_BROWSE)	
Quick Launch	Marketing - Quick Launch (SAP_CEC_BC_MKT_COM_PC (Deprecated as of 2011))	

### 4.2.1.3.3 Extensibility

Customize the way that you use the WeChat integration.

#### Custom Fields

You can add custom fields to the *Digital Accounts* app using the *Custom Fields* app. When creating custom fields in that app, use business context `MKT_DIGITAL_ACCOUNT`.



Due to ABAP DDIC restrictions, only a defined number of fields and characters can be created for each business context.

Business Context	Description	Maximum Number of Fields	Maximum Number of Characters
MKT_DIGITAL_ACCOUNT	Marketing: Digital Account	100	1000

For general information about creating and enabling custom fields, see [Custom Fields](#).

## Custom Logic for Getting WeChat Access Tokens

In the standard delivery, the system requests WeChat access tokens directly from the WeChat server. However, due to business requirements, you may have multiple servers that have outbound connections to the same official account. You use one of them as a primary server, which is responsible for getting and storing access tokens. In this situation, you can set up the system to get access tokens through the primary server by creating an enhancement implementation in the *Custom Logic* app. When creating your enhancement implementation in that app, use the *Marketing: Digital Account* business context and *Getting of Access Token* enhancement option.

For more information, see [Custom Logic](#).

## Required Business Role

A business role that contains the **Extensibility** (SAP\_CORE\_BC\_EXT) business catalog is required for creating custom fields or custom logic. You can use the standard business role **Administrator** (BR\_ADMINISTRATOR), which already contains this business catalog. Alternatively, you can create custom business roles using the *Maintain Business Roles* app.

### 4.2.1.4 LINE Integration

With this integration, you can synchronize the followers of your LINE accounts as well as the follower interactions to SAP Marketing Cloud. What's more, you can create and carry out LINE campaigns through SAP Marketing Cloud. Analytical reports about LINE followers and interactions are available as well.

#### i Note

The LINE integration supports **LINE@ accounts only**.

The following is a detailed list of the business functions that come with the LINE integration:

- Synchronization of LINE followers and interactions to SAP Marketing Cloud

## i Note

Not all LINE followers of a LINE account are necessarily synchronized to SAP Marketing Cloud. Only the following followers are synchronized:

- Those who followed the LINE account before the LINE integration went live and have initiated an interaction with the LINE account after the LINE integration went live
- Those who follow the LINE account after the LINE integration went live

- Creation and execution of LINE campaigns through SAP Marketing Cloud
- Analytical reports about the acquisition of LINE followers, including reports predefined by SAP and custom reports that you can build with a CDS view
- Analytical reports about LINE interactions, including reports predefined by SAP and custom reports that you can build with a CDS view

For information about setting up and administering the LINE integration, see the following documents:

- [Setting Up the LINE Integration \[page 78\]](#)
- [Administering the LINE Integration \[page 84\]](#)

For extensibility options, see [Extensibility \[page 86\]](#).

For the descriptions of the business functions, see the following documents:

- [Followers of Digital Accounts](#)
- [Attributes Related to Followers of Digital Accounts](#)
- [LINE Campaigns](#)

## 4.2.1.4.1 Setting Up the LINE Integration

Set up the connection between SAP Marketing Cloud and your LINE account.

### Procedure

1. Create communication configurations for the inbound communication and outbound communication, respectively.

For more information, see [Creating Communication Configurations \[page 79\]](#).

2. Create your LINE account in SAP Marketing Cloud.

For more information, see [Creating a LINE Account \[page 83\]](#).

3. Create a job based on the template *Digital Accounts: Process Inbound Messages*.

This job is required for the synchronization of followers and interactions. For more information, see [Creating and Scheduling Application Jobs \[page 84\]](#).

## Results

The system synchronizes followers and follower interactions from the LINE server to SAP Marketing Cloud automatically.

### i Note

Not all LINE followers of a LINE account are necessarily synchronized to SAP Marketing Cloud. Only the following followers are synchronized:

- Those who followed the LINE account before the LINE integration went live, but have initiated an interaction with the LINE account since the LINE integration went live
- Those who follow the LINE account after the LINE integration went live

### 4.2.1.4.1.1 Creating Communication Configurations

Create the configurations required for the communication between SAP Marketing Cloud and the LINE server.

## Prerequisites

A business role that contains the **Communication Management** (SAP\_CORE\_BC\_COM) business catalog is required. You can use the standard business role **Administrator** (SAP\_BR\_ADMINISTRATOR), which contains the **Communication Management** business catalog and other administration-related catalogs. Alternatively, you can create custom business roles using the [Maintain Business Roles](#) app.

## Context

The LINE integration involves communication in both the inbound and outbound directions.

### Inbound Communication

You must define a servlet for the inbound communication. In the inbound communication, the LINE server pushes to your servlet LINE events as well as messages that followers send to a LINE account. Then the servlet forwards the LINE events and messages to SAP Marketing Cloud by calling the private ICF service `<host name>/sap/cuan/ntwrk/`.

The inbound communication requires a communication system/communication arrangement pair. In addition, you need to create a communication user for the inbound communication. When calling the private ICF service to forward the LINE events and messages to SAP Marketing Cloud, your servlet must authenticate itself with this communication user first.

## Outbound Communication

The outbound communication involves synchronizing data (for example, basic follower information) from the LINE server and executing LINE campaigns through SAP Marketing Cloud. The outbound communication requires a communication system/communication arrangement pair.

The following communication scenarios are relevant to the LINE integration:

- Marketing – Network Channel Events Integration (SAP\_COM\_0174)
- Marketing – Campaign Execution – LINE Integration (SAP\_COM\_0218)

For detailed instructions, see [Configuring the Inbound Communication \[page 80\]](#) and [Configuring the Outbound Communication \[page 82\]](#).

For general information about communication management, see [Communication Management](#).

### 4.2.1.4.1.1.1 Configuring the Inbound Communication

Create the communication user, communication system, and communication arrangement required for the inbound communication.

## Customer Implementation

You must define a servlet for the inbound communication. This servlet receives the LINE events and messages pushed by the LINE server and then forwards them to SAP Marketing Cloud by calling the private ICF service `<host name>/sap/cuan/ntwrk/`.

## Creating the Communication User

Proceed as follows:

1. Log into SAP Fiori launchpad with a business role that contains the **Communication Management** (SAP\_CORE\_BC\_COM) business catalog.
2. Open the *Maintain Communication Users* app.
3. Choose *New*.  
The *Create Communication User* dialog box appears.
4. Fill in the following fields:
  - *User Name* and *Description* (for example, LINE\_EVENT and LINE Event)
  - *Password*
5. Save the communication user.  
A communication user ID is generated automatically.

### i Note

When calling the private ICF service, your servlet should authenticate itself with the **communication user ID** instead of the user name.

Do not exit SAP Fiori launchpad.

## Creating the Communication System

This communication system is a dummy one. The purpose of it is to bind the communication user that you created earlier with the communication arrangement that you will create later.

To create the communication system, proceed as follows:

1. Open the *Communication Systems* app.
2. Choose *New*.  
The *New Communication System* dialog box appears.
3. Enter a system ID and its name, for example, LINE\_EVENT and LINE Event. Choose *Create*.  
The editing screen for the communication system appears.
4. A host is irrelevant to the inbound communication. Enter **dummy** in the *Host Name* field to assign a dummy host.
5. Assign the communication user created earlier to this communication system, as follows:
  1. In the *User for Inbound Communication* section, choose + (Add).  
The *New Inbound Communication User* dialog box appears.
  2. Enter the user created earlier and select the authentication method *User Name and Password*.
6. Save and activate the communication system.  
Do not exit SAP Fiori launchpad.

## Creating the Communication Arrangement

Proceed as follows:

1. Open the *Communication Arrangements* app.
2. Choose *New*.  
The *New Communication Arrangement* dialog box appears.
3. Enter scenario SAP\_COM\_0174 and an arrangement name. Choose *Create*.  
The editing screen for the communication arrangement appears.
4. In the *Communication System* field, enter the communication system created earlier.
5. Save and activate the communication arrangement.

## Next Steps

[Configuring the Outbound Communication \[page 82\]](#)

## 4.2.1.4.1.1.2 Configuring the Outbound Communication

Create the communication system and communication arrangement required for the outbound communication.

### Creating the Communication System

Proceed as follows:

1. Open the *Communication Systems* app.
2. Choose *New*.  
The *New Communication System* dialog box appears.
3. Enter a system ID and its name, for example, LINE\_API and LINE API. Choose *Create*.  
The editing screen for the communication system appears.
4. Under **▶ Technical Data ▶ General ▶**, enter **api.line.me** in the *Host Name* field, which is the host name of the LINE server. Choose *Save*.

#### **i** Note

If you have your own logic for getting and posting LINE data through another server, then enter the host name of that server instead.

5. Set the authentication method to *None*, as follows:
  1. In the *User for Outbound Communication* section, choose + (Add).  
The *New Outbound User* dialog box appears.
  2. Select the authentication method *None*.
  3. Choose *Create*.
6. Save and activate the communication system.  
Do not exit SAP Fiori launchpad.

### Creating the Communication Arrangement

Proceed as follows:

1. Open the *Communication Arrangements* app.
2. Choose *New*.  
The *New Communication Arrangement* dialog box appears.
3. Enter scenario SAP\_COM\_0218 and an arrangement name. Choose *Create*.  
The editing screen for the communication arrangement appears.
4. In the *Communication System* field, enter the communication system that you created earlier.
5. Activate all the outbound services by selecting the *Active* checkboxes.
6. Save and activate the communication arrangement.

## Next Steps

[Creating a LINE Account \[page 83\]](#)

### 4.2.1.4.1.2 Creating a LINE Account

Create a LINE account in SAP Marketing Cloud.

## Prerequisites

A business role that contains the **Marketing - Master Data** (SAP\_CEC\_BC\_MKT\_PRD\_PC) business catalog is required. You can use the standard business role **Marketing Expert** (SAP\_BR\_MARKETING\_EXPERT), which contains the **Marketing - Master Data** business catalog and other marketing-related catalogs. Alternatively, you can create custom business roles using the [Maintain Business Roles](#) app.

## Procedure

1. Log into SAP Fiori launchpad with a business role that contains the **Marketing - Master Data** (SAP\_CEC\_BC\_MKT\_PRD\_PC) business catalog.
2. Open the [Digital Accounts](#) app.
3. On the landing page, choose [Create Digital Account](#).  
The [Select Digital Account Type](#) dialog box appears.
4. Select [LINE Account](#).  
The editing screen for the LINE account appears.
5. Select the marketing area and enter the required information. Choose [Create](#).
6. In the Channel Console, enter the URI of your servlet on the relevant configuration page.

## Results

The LINE account appears with the [Active](#) status on the landing page of the [Digital Accounts](#) app. By default, the LINE account inherits the profile picture from the LINE platform. If there is no profile picture set on the LINE platform, the system sets the image assigned to the [LINE](#) communication medium in the [Manage Images](#) app as the profile picture of the LINE account. Clicking the account displays various tabs that contain different kinds of information about the official account. The [Information](#) tab contains basic information about the LINE account and the credentials. You can change the profile picture, credentials, and so on, by clicking the [Edit](#) button.

## Next Steps

[Creating and Scheduling Application Jobs \[page 84\]](#)

### 4.2.1.4.2 Administering the LINE Integration

Learn about the system administration activities relevant to the LINE integration.

#### 4.2.1.4.2.1 Creating and Scheduling Application Jobs

Learn about the application jobs required for the LINE integration.

#### Required Application Job

To use the LINE integration, you must create an application job using the following template:

Application Job Template	Description
<a href="#">Digital Accounts: Process Inbound Messages</a>	A job created using this template creates contacts and interactions from digital accounts, including LINE official accounts, in SAP Marketing Cloud. For more information, see <a href="#">Digital Accounts: Process Inbound Messages</a> .

#### Checking the Application Log

You can find a log of the application job from the [Application Logs](#) app using category CUAN and subcategory CUAN\_NTWRK as filters.

Alternatively, you can find the log directly from the [Marketing Application Jobs](#) app. From the application job list, click the  icon next to the application job.

#### Required Business Role

A business role that contains the **Marketing - Business Administration** (SAP\_CEC\_BC\_MKT\_ADM\_PC) business catalog is required for scheduling application jobs and checking logs. You can use the standard business role



**Administrator - Marketing** (SAP\_BR\_ADMINISTRATOR\_MKT), which already contains this business catalog. Alternatively, you can create custom business roles using the *Maintain Business Roles* app.

## 4.2.1.4.2.2 Activating, Deactivating, and Restricting a LINE Account

You can set a LINE account to any of these statuses in the *Digital Accounts* app: *Active*, *Inactive*, and *Restricted*.

### Activating a LINE Account

To use the full functionality of the LINE integration, you must set a LINE account to *Active* status. When completing creating a LINE account, the status of the account is set to *Active* automatically.

To set a LINE account from *Restricted* status to *Active* status, choose the *Activate* button.

To set a LINE account from *Inactive* status to *Active* status, choose the *Switch to Restricted Mode* button and then the *Activate* button.

### Deactivating a LINE Account

If you do not want to connect to a LINE account, for example, because the account is no longer in use, deactivate it by choosing the *Deactivate* button.

### Restricting a LINE Account

When a LINE account is restricted, the outbound connection to the account stops working. Therefore, the following functions become unavailable:

- Execution of LINE campaigns through SAP Marketing Cloud
- Synchronization of basic follower information

When a LINE account is restricted, there is no impact on the inbound connection from the account. The synchronization of interactions between followers and the account still works and the messages that followers send to the account are still synchronized.

To set a LINE account to *Restricted* status, choose the *Switch to Restricted Mode* button.

### 4.2.1.4.2.3 Overview of Business Catalogs Required for Different Business Functions

Different business functions require different business catalogs. Learn about the business catalogs that are required for the functions related to the LINE integration and assign business roles to business users appropriately.

The following table lists the business catalogs that are required for different functions:

Business Function	Required Business Catalog	Standard Business Role That Can Be Used
Functions Available with the <i>Digital Accounts</i> App	<b>Marketing - Data (fka Products)</b> (SAP_CEC_BC_MKT_PRD_PC)	<b>Marketing Expert</b> (SAP_BR_MARKETING_EXPERT)
Contact Profiles of LINE Followers	<b>Marketing - Contacts and Profiles Base</b> (SAP_CEC_BC_MKT_DMB_PC)  <b>Marketing - Contacts and Profiles Standard</b> (SAP_CEC_BC_MKT_DMS_PC)	
Segmentation	<b>Marketing – Segmentation</b> (SAP_CEC_BC_MKT_SEG_PC)	
LINE Campaigns	<b>Marketing - Campaign Management</b> (SAP_CEC_BC_MKT_CPM1_PC)	
Custom analytical reports in the <i>Query Browser</i> app	<b>Marketing - Data (fka Products)</b> (SAP_CEC_BC_MKT_PRD_PC)  <b>Query Browser</b> (SAP_CA_BC_VDM_BROWSE)	
Quick Launch	<b>Marketing - Quick Launch</b> (SAP_CEC_BC_MKT_COM_PC, Deprecated as of 2011)	

### 4.2.1.4.3 Extensibility

Customize the way that you use the LINE integration.

#### Custom Fields

You can add custom fields to the *Digital Accounts* app using the *Custom Fields* app. When creating custom fields in that app, use business context `MKT_DIGITAL_ACCOUNT`.

Due to ABAP DDIC restrictions, only a defined number of fields and characters can be created for each business context.

Business Context	Description	Maximum Number of Fields	Maximum Number of Characters
MKT_DIGITAL_ACCOUNT	Marketing: Digital Account	100	1000

For general information about creating and enabling custom fields, see [Custom Fields](#).

## Required Business Role

A business role that contains the **Extensibility** (SAP\_CORE\_BC\_EXT) business catalog is required for creating custom fields or custom logic. You can use the standard business role **Administrator** (BR\_ADMINISTRATOR), which already contains this business catalog. Alternatively, you can create custom business roles using the [Maintain Business Roles](#) app.

### 4.2.1.5 Integration with Google Analytics

Overview of the integration scenario.

The integration with Google Analytics allows you to do the following:

- Enrich sales order interactions of type SALES\_ORDER with Google Analytics data.  
You can enrich a sales order interaction with its source campaign and device category information. For more information, see [Interactions: Enrich Sales Orders with Google Analytics](#).
- Create interactions with web tracking data from Google Analytics or Google BigQuery.  
You create query configurations that identify the set of web hits data that you want to retrieve from Google Analytics or Google BigQuery. You define mapping values that identify how to map the data retrieved from Google to the interaction data in SAP Marketing Cloud. For more information, see [Overview of Create Interactions Scenario](#).
- Match and merge additional contact ID information in SAP Marketing Cloud with data retrieved from Google Analytics or Google BigQuery.  
You create query configurations that identify the contact ID information that you want to retrieve. You import contact ID data which goes through the match and merge process. For more information, see [Overview of Match and Merge Scenario](#).

#### i Note

You can set up one or more of the scenarios, depending on your needs.

## Configuration Settings

For a complete description of the configuration settings required for the integration scenario, see the [Integration Guide](#).

## Integration Package

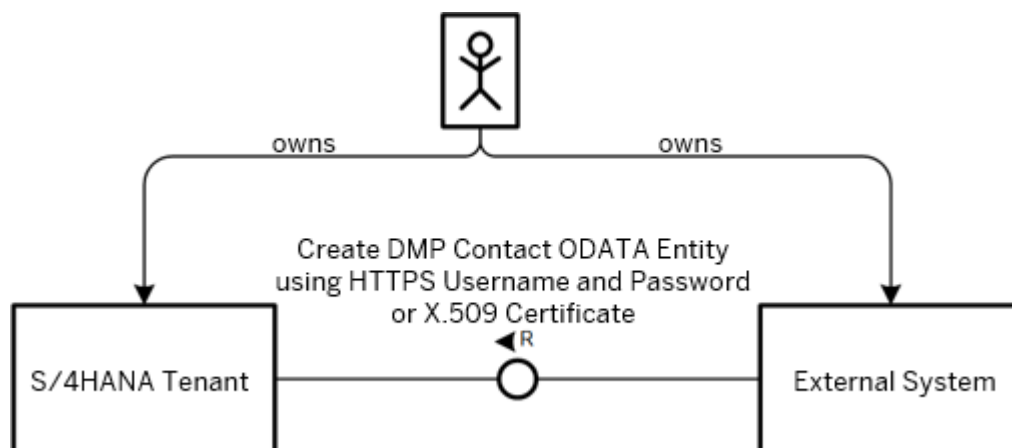
For more information about the *Google Analytics Integration with SAP Marketing Cloud/SAP Marketing* integration package, see [the SAP API Business Hub](#).

### 4.2.1.6 Integration with Data Management Platforms

With this integration scenario, you can capture and replicate Data Management Platform (DMP) IDs from DMP providers, such as Adform. A DMP ID is mapped to a commerce contact ID and stored inside SAP Marketing Cloud.

For more information about Adform, see [Advertiser Edge by Adform A/S](#).

The following graphic shows the overall process:



To use this integration, you must configure SAP Marketing Cloud.

#### 4.2.1.6.1 Configuring SAP Marketing Cloud

To establish communication with the OData service, you perform procedures in SAP Marketing Cloud.

The overall process is as follows:

1. [Define a Communication User \[page 89\]](#)

You can use an existing communication user, or create a new one.

2. [Set Up the Communication System \[page 89\]](#)  
After defining your communication user, set up a communication system for the DMP integration scenario.
3. [Set Up the Communication Arrangement \[page 90\]](#)  
After setting up the communication system, set up the communication arrangement for the DMP integration scenario.

### 4.2.1.6.1.1 Define a Communication User

You can use an existing communication user, or create a new one.

#### Procedure

1. Log on to your SAP Marketing Cloud system with a user that has administrator authorizations.
2. Launch the *Maintain Communication Users* app and choose *New*.
3. Enter the required *User Name*, *Description*, and *Password*.
4. Create and save your user.

Note down the user data for further processes.

**Task overview:** [Configuring SAP Marketing Cloud \[page 88\]](#)

**Next task:** [Set Up the Communication System \[page 89\]](#)

### 4.2.1.6.1.2 Set Up the Communication System

After defining your communication user, set up a communication system for the DMP integration scenario.

#### Prerequisites

To set up a communication system and communication arrangement, you require the Communication Management (SAP\_CORE\_BC\_COM) business catalog role.

#### Procedure

1. Log on to your SAP Marketing Cloud system with a user that has administrator authorizations.

2. From the SAP Fiori launchpad, choose the *Communication Systems* app.
3. Choose *New*.
4. Enter a system ID and name for your communication system.
5. Choose *Create*.
6. On the *Communication System* page, enter the following:
  - a. Under *Technical Data*, enter **dummy** as the *Host Name* to assign a dummy host.

This is a dummy communication system as its only purpose is to bind the communication user that you previously created to the communication arrangement that you will create in the next step.
  - b. Under *User for Inbound Communication*, choose (+) and enter your communication user name.
  - c. For *Authentication Method*, select *User Name and Password*.
7. Save your changes and exit the app.

**Task overview:** [Configuring SAP Marketing Cloud \[page 88\]](#)

**Previous task:** [Define a Communication User \[page 89\]](#)

**Next task:** [Set Up the Communication Arrangement \[page 90\]](#)

### 4.2.1.6.1.3 Set Up the Communication Arrangement

After setting up the communication system, set up the communication arrangement for the DMP integration scenario.

#### Prerequisites

To set up a communication system and communication arrangement, you require the Communication Management (SAP\_CORE\_BC\_COM) business catalog role.

#### Procedure

1. Log on to your SAP Marketing Cloud system with a user that has administrator authorizations.
2. From the SAP Fiori launchpad, choose the *Communication Arrangements* app.
3. Create a new communication arrangement.
4. Select SAP\_COM\_0343 (*Marketing – DMP Integration*).
5. Choose *Create*.
6. In the *Communication Arrangements* screen, do the following:
  - a. Under *Common Data*, choose the communication system that you created previously.

- b. Under *Inbound Communication*, choose the communication user name for inbound services that you assigned to the communication system.
7. Save your changes and exit the app.

**Task overview:** [Configuring SAP Marketing Cloud \[page 88\]](#)

**Previous task:** [Set Up the Communication System \[page 89\]](#)

## 4.2.2 Landing Pages and Forms

Integration options for landing pages and forms.

### Landing Pages

Find out how to get data entered on external landing pages into SAP Marketing Cloud. For more information, see [Landing Pages and Forms \[page 91\]](#).

#### i Note

If you create a landing page (with or without a form) using the landing page editor in the *Content Studio*, you can publish your page immediately. There is no configuration necessary. For more information, see [Landing Page Design](#).

### Forms

If you use forms standalone, the following documentation is relevant:

- Custom integration allows you to host forms on your own Web server and connect them to your SAP Marketing Cloud system. For more information, see [Custom Integration of Forms \[page 92\]](#).

#### i Note

If you use forms as part of landing pages, custom integration is not relevant. You need only the documentation for standard integration and form publication. Please also note that a form does not need to be published in order to use it in a landing page.

- Standard integration allows you to host forms on your own Web server and to connect them to your SAP Marketing Cloud system using an SAP integration service on the SAP Business Technology Platform. For more information, see [Standard Integration of Forms \[page 102\]](#).
- You can automate the process of making the form files available on customer web servers. For more information, see [Form Publication \[page 105\]](#).
- You can captcha configuration to enhance the security of your forms and decrease vulnerability to malicious attacks by bots that send fraudulent contact data into your system. For more information, see [Setting Up Captcha Configuration for Forms \[page 264\]](#).

## 4.2.2.1 Custom Integration of Forms

Custom integration of forms supports the technical user with specific integration tasks when deploying forms in the customer's Web server. You can alternatively integrate a form into a landing page and publish it immediately out-of-the-box.

### Use

This document provides details for the technical user to support with implementing an integration service between the HTML-based frontend and the OData-based backend when deploying forms in the customer's Web server.

### Setting up Forms

In SAP Marketing Cloud , you are provided with the *Forms* content type in the *Content Studio* app. This allows you to design forms to collect interaction contact and marketing permission data.

For security and performance reasons, you must deploy the forms you create on your Web server to make them available to the internet. The Web server must send the collected data to the server of SAP Marketing Cloud using the public OData service `CUAN_CONTENT_PAGE_RESULT_SRV`. The service saves the data and triggers follow-on actions.

Implementation of forms includes of the following:

- Deploying the files onto your Web server
- Implementing the Web server in order to forward the results to SAP Marketing Cloud

### Related Information

[Forms](#)

[Form Publication \[page 105\]](#)



## 4.2.2.1.1 Creating the Communication Arrangement

To prepare for the technical implementation of forms, the administrator must create a user, a communication system, and a communication arrangement.

### Create a Communication User

1. Open the *Maintain Communication Users* app and click *New*.
2. Enter a username, for example **MKT\_FORM\_RESULT\_USER**, and a description, for example **Marketing - Form Result User**.
3. Enter a password.
4. Click *Create*.

### Create a Communication System

1. Open the *Communication Systems* app and click *New*.
2. Enter a system ID and Name, for example **MKT\_FORM\_TEST\_SERVER**.
3. Enter the domain name of the server you host your form on.
4. Add the previously created communication user under *User for Inbound Communication*.
5. Click *Save*.

### Create a Communication Arrangement

1. Open the *Communication Arrangements* app and click *New*.
2. Select the integration scenario **SAP\_COM\_0023** (Marketing - Form Integration).
3. If you like, you can adjust the proposed Arrangement name before clicking *Create*.
4. Select the previously created communication system under *Common Data*.
5. The previously created communication user should appear automatically under *Inbound Communication*. If one does not appear, return to the communication system and make sure that a user was added under *User for Inbound Communication*.
6. Click *Save*.

The communication user you created has the integration role assigned from the arrangement, and is ready for use in your integration.

## 4.2.2.1.2 Deploying the Form

The source code (HTML) for every form you design must be downloaded using the user interface and deployed on your Web server. The HTML file that is generated describes the design and the content of the specific form.

In addition, you must download a style sheet (CSS) and a JavaScript file and adapt them according to your requirements and your system setup. This step is required for your initial system setup and allows you to deploy the CSS and JavaScript files.

#### **i Note**

You do not need to adapt these files again until you perform an upgrade to a new release. Adapting the files when you upgrade ensures that you can avail of new features.

By default, the HTML file tries to load both files with the names `sapContentPage.css` and `sapContentPage.js` from the same folder in which the HTML file is located. If you want to adjust those names or the file paths you must change the HTML file content.

### **4.2.2.1.3 Adjusting the JavaScript File**

After downloading the JavaScript file you must adjust the Web service path written in the file.

The default base path is the path for the OData service on the SAP Marketing Cloud server:

- `/sap/opu/odata/sap/CUAN_CONTENT_PAGE_RESULT_SRV`

You must adjust this path according to the Web server implementation (see section [Implementing the Result OData Service \[page 94\]](#)).

When a user opens a form, the path is loaded using an `HTTP HEAD` request to fetch a CSRF token. Further data requests are sent to the result path, which is added to the base path. The result path can be adjusted or left empty in the JavaScript file. The default path `ResultHeaders` describes the OData service entity used for the results.

You are not required to perform any implementation for the front end. The JavaScript that is delivered collects the user input independently. It is possible to adapt and enhance the form HTML file that is generated, but you must preserve the integrity of the standard structure.

### **4.2.2.1.4 Implementing the Result Service**

You must implement your Web server to enable it to receive the requests of the form JavaScript files and forward them to the SAP Marketing Cloud server.

By default, the requests are ready for the result service `CUAN_CONTENT_PAGE_RESULT_SRV` and do not need to be adjusted.

#### **i Note**

In some usage scenarios, the request data must be enhanced to enable all features. To do this, you must decode the JSON payload string and add the appropriate attributes before encoding the JSON string again for the result service.

## ⚠ Caution

The actual implementation depends on the technology and development language that you use in your company. The sample code provided below is an example PHP implementation. SAP does not take responsibility if you use it in your productive system.

To use this PHP implementation, you must adjust the `BasePath` and `ResultHeadersPath` at the beginning of your Javascript file to read as follows:

```
(function () {
    "use strict";
    var C = {
        BasePath: "./myLandingPageIntegrationScript.php",
        ResultHeadersPath: "",
        ...
    }
})
```

Here `myLandingPageIntegrationScript.php` stands for the name of the PHP script, and must be replaced by the name of your PHP script.

## PHP Example

### ☰ Sample Code

```
<?php
/**
 * This class is an example implementation
 * of a PHP based form integration.
 * Note: This is a template, which is used at your own risk.
 */
class LandingPageIntegration
{
    /**
     * The BasePath is the URL for the system
     * including the form result service.
     *
     * @var string
     */
    const BASE_PATH = "https://<server>:<port>/sap/opu/odata/sap/
CUAN_CONTENT_PAGE_RESULT_SRV/";
    /**
     * The ResultHeadersPath is the name of the ResultHeaders entity
     * which is used for processing the form results.
     *
     * @var string
     */
    const RESULT_HEADERS_PATH = "ResultHeaders";
    /**
     * The credentials are used for authenticating on the system.
     * This is usually a dedicated system or communication user
     * with the integration role assigned.
     *
     * @var string
     */
    const CREDENTIALS = "USERNAME:PASSWORD";
    /**
     * The cookies are remembered between consecutive OData requests
     * to implement the session handling
     * and security measures of the SAP Gateway.
     *
     */
}
```

```

* @var string
*/
private $cookies = "";
/**
 * The CSRF-Token is required for the OData service communication
 * and must be fetched before it is possible
 * to perform any changing requests such as 'POST'.
 *
 * @var string
 */
private $csrfToken = null;
/**
 * This method is the main entry point
 * for processing the requests received from forms.
 */
public function execute()
{
    switch ($_SERVER["REQUEST_METHOD"]) {
        case "POST":
            $this->handlePostRequest();
            break;
    }
}
/**
 * POST requests must be forwarded to the system
 * and the responses must be passed to the client
 * to ensure correct form integration.
 */
private function handlePostRequest()
{
    // first fetch the csrf-token
    $this->fetchCsrfToken();

    // read the POST data sent by the form
    $requestBody = @file_get_contents("php://input");
    $requestData = json_decode($requestBody);

    // optional: enhance the request data with the IP address for
tracking purposes
    $requestData->IpAddress = $_SERVER["REMOTE_ADDR"];

    // optional: add the campaign id to connect all form interactions to
your campaign
    // $requestData->CampaignId = "your-campaign-id";

    // send the prepared request data to the system
    $requestString = json_encode($requestData);
    $response = $this->sendHttpRequest("POST", $this::BASE_PATH .
$this::RESULT_HEADERS_PATH, $requestString);

    // print the response
    echo $response;
}
/**
 * Send a 'HEAD' request to fetch
 * the required CSRF-Token from the OData service.
 * If the HEAD request fails, a 'GET' request is performed.
 */
private function fetchCsrfToken()
{
    $this->sendHttpRequest("HEAD", $this::BASE_PATH, null);
    if (!$this->csrfToken) {
        // HEAD request failed -> fallback using GET
        $this->sendHttpRequest("GET", $this::BASE_PATH, null);
    }
}
/**
 * This method performs a synchronous HTTP request

```

```

* and returns its response.
*
* @param string $method
*         The HTTP method (e.g. 'HEAD', 'POST')
* @param string $path
*         The URL for the request
* @param string $body
*         The request payload (POST data)
* @return string The response
*/
private function sendHttpRequest($method, $path, $body)
{
    // first create stream context
    $context = $this->createStreamContext($method, $body);

    // perform http request
    $response = file_get_contents($path, false, $context);

    if ($response === false) {
        // request failed - print error for analysis
        $error = error_get_last();
        if (is_array($error)) {
            echo $error["message"];
        } else {
            echo $error;
        }
    }

    // process response headers
    $this->readResponseHeaders($http_response_header);

    // return response
    return $response;
}
/**
 * This method creates a stream context, which is used for the HTTP
request.
 * It configures the context for
 * the authorization, content-type, cookies, and csrf-token.
 *
 * @param string $method
 *         The HTTP method
 * @param string $body
 *         The request payload (POST data)
 * @return resource The stream context
 */
private function createStreamContext($method, $body)
{
    // basic authorization uses base64 encoded credentials
    $credentials = base64_encode($this::CREDENTIALS);

    // build http request headers
    $headers = array(
        "Authorization: Basic " . $credentials,
        "Accept: application/json",
        "Content-Type: application/json"
    );

    if ($this->cookies) {
        // add remembered cookies
        array_push($headers, "Cookie: " . $this->cookies);
    }

    // add x-csrf-token header for fetching or using the already fetched
token
    $csrfToken = ($this->csrfToken ? : "Fetch");
    array_push($headers, "x-csrf-token: " . $csrfToken);
}

```

```

// build complete options array
$options = array(
    "http" => array(
        "header" => $headers,
        "method" => $method,
        "content" => $body,
        "ignore_errors" => true,
        "max_redirects" => 0
    )
);

// return stream context using the built options
return stream_context_create($options);
}
/**
 * This method processes the HTTP response headers
 * in order to read the fetched CSRF-Token and cookies.
 *
 * @param array $responseHeaders
 */
private function readResponseHeaders($responseHeaders)
{
    // loop response headers
    foreach ($responseHeaders as $responseHeader) {
        // split header name from value
        $parts = explode(" ", $responseHeader);

        // handle response header based on name
        switch (strtolower($parts[0])) {
            case "HTTP/1.0":
                // status code
                http_response_code($parts[1]);
                break;
            case "x-csrf-token:":
                // save fetched csrf-token
                $this->csrfToken = $parts[1];
                break;
            case "set-cookie:":
                // set received cookies
                $this->cookies .= $parts[1];
                break;
        }
    }
}
// initialize the integration class and start the processing
$LandingPageIntegration = new LandingPageIntegration();
$LandingPageIntegration->execute();

```

## Java Example

### Sample Code

```

package com.sap.hpa.cei.cntpg.man.integration;
/**
 * This class is an example implementation
 * of a Java-based form integration.
 * Note: This is a template, which is used at your own risk.
 */
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;

```

```

import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.CookieHandler;
import java.net.CookieManager;
import java.net.HttpURLConnection;
import java.net.URL;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import sun.misc.BASE64Encoder;
public class IntegrationServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    /**
     * The BasePath is the URL for the system including the form result
     * OData service. Note: It needs to include the trailing slash (/).
     *
     * @var String
     */
    private static final String BASE_PATH = "https://<server>:<port>/sap/opu/
odata/sap/CUAN_CONTENT_PAGE_RESULT_SRV/";
    /**
     * The ResultHeadersPath is the name of the ResultHeaders entity which is
     used
     * for processing the form results.
     *
     * @var String
     */
    private static final String RESULT_HEADERS_PATH = "ResultHeaders";
    /**
     * The credentials are used for authenticating on the system. This is
     usually a
     * dedicated system or communication user with the integration role
     assigned.
     *
     * @var String
     */
    private static final String CREDENTIALS = "USERNAME:PASSWORD";
    /**
     * The cookie manager remembers cookies between consecutive OData
     requests to
     * implement the session handling and security measures of the SAP
     Gateway.
     *
     * @var CookieManager
     */
    private CookieManager cookieManager = null;
    @Override
    protected void doHead(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    }
    /**
     * Handle POST requests containing the results of forms
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
        String content = this.readContent(request.getReader());
        String csrfToken = this.fetchCsrfToken();
        String responseText = this.postData(content, csrfToken);
        response.getWriter().write(responseText);
    }
    /**
     * Read request body
     */
    private String readContent(BufferedReader reader) throws IOException {
        StringBuffer stringBuffer = new StringBuffer();

```

```

        String line = null;
        while ((line = reader.readLine()) != null) {
            stringBuffer.append(line);
        }
        return stringBuffer.toString();
    }
    /**
     * Send a HEAD request to fetch the CSRF token
     */
    private String fetchCsrfToken() throws IOException {
        HttpURLConnection connection = this.createConnection(BASE_PATH,
"HEAD", null);
        connection.connect();
        String csrfToken = connection.getHeaderField("x-csrf-token");
        connection.disconnect();
        return csrfToken;
    }
    /**
     * Send POST request to forward the form result to the backend system
     */
    private String postData(String data, String csrfToken) throws IOException
    {
        // open HTTP connection and send body
        HttpURLConnection connection = this.createConnection(BASE_PATH +
RESULT_HEADERS_PATH, "POST", csrfToken);
        connection.setDoOutput(true);
        OutputStream outputStream = connection.getOutputStream();
        outputStream.write(data.getBytes());
        connection.connect();
        // read success or error response
        InputStream inputStream;
        if (200 <= connection.getResponseCode() &&
connection.getResponseCode() <= 299) {
            inputStream = connection.getInputStream();
        } else {
            inputStream = connection.getErrorStream();
        }
        InputStreamReader inputStreamReader = new
InputStreamReader(inputStream);
        BufferedReader reader = new BufferedReader(inputStreamReader);
        String response = this.readContent(reader);
        connection.disconnect();
        return response;
    }
    private HttpURLConnection createConnection(String path, String method,
String csrfToken) throws IOException {
        if (this.cookieManager == null) {
            // create the cookie manager
            this.cookieManager = new CookieManager();
            CookieHandler.setDefault(this.cookieManager);
        }
        // open HTTP connection and set relevant headers
        HttpURLConnection connection = (HttpURLConnection) new
URL(path).openConnection();
        connection.setRequestMethod(method);
        connection.setRequestProperty("Accept", "application/json");
        connection.setRequestProperty("Content-Type", "application/json");
        connection.setRequestProperty("Connection", "keep-alive");
        // set base64-encoded authorization header
        BASE64Encoder encoder = new BASE64Encoder();
        String credentials = encoder.encode(CREDENTIALS.getBytes());
        connection.setRequestProperty("Authorization", "Basic " +
credentials);
        // set CSRF token header to 'Fetch' or to the actual token value if
available
        if (csrfToken != null) {
            connection.setRequestProperty("x-csrf-token", csrfToken);
        } else {

```



```
        connection.setRequestProperty("x-csrf-token", "Fetch");
    }
    return connection;
}
}
```

## System User Authentication

The result service `CUAN_CONTENT_PAGE_RESULT_SRV` can only be called by users with the corresponding authorization. You must use the user created for integration scenario `SAP_COM_0023`. The example PHP script shows the authentication using an `HTTP` header named **Authorization using Basic Authentication** (user and password).

## Contact Identification

The forms integration offers different ways to identify the Web user who visits the form. The following usage scenarios are supported:

- **Scenario A:** The Web user is anonymous (unknown)  
In this scenario, the user cannot be identified on the form.
- **Scenario B:** The Web user has accessed the form using a tracking link in an SAP Marketing Cloud email.  
Scenario B does not require any additional implementation effort. The form script performs the required actions autonomously. If the form is accessed using a SAP Marketing Cloud email, the link contains a tracking ID that is sent along with the data requests. This ID is used to identify the user that received the email.

## Prefill Contact Data

When a Web user who accesses a form is identified, it is possible to prefill data for the *Input* and *Permission* elements in the form.. Selecting the *Prefill Contact Data* checkbox allows the form elements to be filled with data for the identified contact, which is maintained in the SAP Marketing Cloud system.

To support the prefill of contact data, the Web server implementation needs to pass the response data from the SAP Marketing Cloud system to the form (web client) that initiated the request. The SAP Marketing Cloud system provides all necessary data for forms with the *Prefill Contact Data* setting. There is no additional effort for the implementation, apart from the forwarding of response data.

## Optional Attributes

In order to complete the form integration, you can enhance the OData requests with the following optional attributes:

- *IPAddress*  
The IP address of the web client visiting a form can be saved in order to have additional evidence that the user submitted the form, and gave marketing permissions and contact data.
- *CampaignId*  
The campaign ID can be supplied to connect the interactions created out of the form to a specific SAP Marketing Cloud campaign. If the form is opened with a URL parameter `sap-campaign-id` with its value set to the ID, it is automatically added to all form requests. This connection will also be created if the form is opened out of a SAP Marketing Cloud email sent as part of a campaign.

### 4.2.2.2 Standard Integration of Forms

The SAP Marketing Cloud offers a built-in integration service, which allows you to use forms on your web server without having to implement a custom integration of forms service as described in [Custom Integration of Forms \[page 92\]](#).

#### Use

This document provides details for the technical user to support with implementing an integration between the HTML-based frontend and the OData-based backend when deploying forms in the customer's Web server.

## Setting Up Forms

In SAP Marketing Cloud, you are provided with the *Forms* content type in the *Content Studio* app.. This allows you to design forms to collect interaction contact and marketing permission data.

For security and performance reasons, you must deploy the forms you create on your Web server to make them available to the internet. The form sends the collected data to an elastic service on the SAP Cloud Integration, which forwards the data to the SAP Marketing Cloud using the public OData service `CUAN_CONTENT_PAGE_RESULT_SRV`. The service saves the data and triggers follow-on actions.

Implementation of forms includes of the following:

- Preparing the form files
- Deploying the files onto your Web server

#### 4.2.2.2.1 Deploying the Form

The source code (HTML) for every form you design must be downloaded using the user interface and deployed on your Web server. The HTML file that is generated describes the design and the content of the specific form.

In addition, you must download a style sheet (CSS) and a JavaScript file and adapt them according to your requirements and your system setup. This step is required for your initial system setup and allows you to deploy the CSS and JavaScript files.

### **i Note**

You do not need to adapt these files again until you perform an upgrade to a new release. Adapting the files when you upgrade ensures that you can avail of new features.

By default, the HTML file tries to load both files with the names `sapContentPage.css` and `sapContentPage.js` from the same folder in which the HTML file is located. If you want to adjust those names or the file paths you must change the HTML file content.

## **4.2.2.2.2 Adjusting the JavaScript File**

After downloading the JavaScript file, you must adjust the configuration variables that are included. These variables define how the landing pages communicate with the backend system. Please be aware that the JavaScript file is only delivered in a minified version, which saves resources and increases performance for end users.

The following variables must be adjusted to use the standard landing page integration:

- **BasePath**
  - The default base path is the path for the OData service on the SAP Marketing Cloud server: `/sap/opu/odata/sap/CUAN_CONTENT_PAGE_RESULT_SRV/`
  - You must adjust this path to the respective SAP Cloud Integration service URL depending on the data center of your SAP Marketing Cloud system:

SAP Cloud Integration Service URLs

<b>Data Center</b>	<b>URL</b>
Sydney (AP)	<code>"https://s4cloudlpicb1aab197.ap1.hana.ondemand.com/elastic-access/sap/lpi/"</code>
Shanghai (CN)	<code>"https://s4cloudlpiz4055ed249.cn1.hana.ondemand.com/elastic-access/sap/lpi/"</code>
Rot (EU)	<code>"https://s4cloudlpia9f27a988.hana.ondemand.com/elastic-access/sap/lpi/"</code>
Tokyo (JP)	<code>"https://s4cloudlpib423c25653.jp1.hana.ondemand.com/elastic-access/sap/lpi/"</code>
Moscow (RU)	<code>"https://s4cloudlpin6265058ca.ru1.hana.ondemand.com/elastic-access/sap/lpi/"</code>

Data Center	URL
Sterling (US)	"https://s4cloudlpihe4b6c67a.us3.hana.ondemand.com/elastic-access/sap/lpi/"

- **CORS**

The default value is *false* and must be changed to *true*. This change ensures that the landing page sends its requests in the correct manner to support Cross-Origin-Resource-Sharing (CORS).

#### **i** Note

The value is a boolean value, so you must write false and true without using apostrophes.

- **CSRFTokenHeader**

The default value "X-CSRF-Token" can be removed by changing it to an empty string "".

- **AppendScenarioParameter**

The default empty value "" must be changed to "\_L54AD1F204\_", which ensures that the landing page sends the technical parameter as part of its requests. This provides the elastic service on the SAP Cloud Integration with the required scenario information.

- **Tenant**

The default empty value "" must be filled with the domain name of your SAP Marketing Cloud system. The name value must respect the following format: "my123456-api.s4hana.ondemand.com".

It's not necessary to change any other parts of the JavaScript file to use the standard functionality.

## Example Configuration

### Sample Code

The following is an example of the JavaScript file after you have implemented the above changes:

```
[...] (function() {"use strict";var
C={BasePath:"https://s4cloudlpi9f27a988.hana.ondemand.com/elastic
-access/sap/
lpi/",ResultHeadersPath:"ResultHeaders",CORS:true,CSRFTokenHeader:"",AppendSce
narioPa
rameter:"_L54AD1F204_",Tenant:"my123456-
api.s4hana.ondemand.com",Version:"1.2.3"}; [...]
```

## 4.2.2.3 Form Publication

The automated form publication describes a scenario that makes it possible to automate the process of making the form files available on customer web servers.

This scenario replaces the manual activities of downloading the files and uploading them to the web server.

### Implementing the Publication Service

A standard solution can't take care of storing the files on the customer web server considering that there are many technologies available and customers might need to adjust the files and their locations to their needs. Therefore, SAP Marketing Cloud provides logic that calls a custom service on the desired target system making it possible to automatically store the files using your own implementation.

#### Request

This publication service needs to be implemented to handle the requests sent by SAP Marketing Cloud. These requests use the HTTP method 'POST' with a JSON payload. The format of the JSON body looks as follows:

#### Sample Code

```
{
  "landingPageKey": "<Form key>",
  "landingPageName": "<Form name>",
  "systemId": "<System ID>",
  "targetId": "<Publication target ID>",
  "baseDirectory": "<Base directory/folder>",
  "htmlFileName": "<File name entered by the business user>",
  "html": "<Form HTML file content>",
  "css": "<Form CSS file content>",
  "js": "<Form JS file content>"
}
```

#### Note

The placeholder texts in the above code sample are updated to mention forms. The technical names however still mention landing pages for compatibility reasons.

The publication service handling these requests needs to store the form files contained in the JSON attributes "html", "css" and "js" on the file system according to your needs. For this purpose, the implementation can use additional details like the form name, system ID, target ID, or base directory to distinguish between multiple possible locations or projects.

Before the request is sent, SAP Marketing Cloud adjusts the file contents using the parameters provided for the publication target. This involves changes the paths to the CSS and JavaScript files in the HTML file content and the service paths in the JavaScript file content.

The service implementation needs to store the files in the correct location for the selected publication target. Otherwise the references in the files won't be correct and the form can't be used. If necessary, the implementation can also adjust the files on your side. However, there is also a Business Add-In (BAI) available to do this in SAP Marketing Cloud.

## Response

SAP Marketing Cloud expects a response from the publication service containing details about the form address. This address will be displayed to the business user for further usage.

The response body needs to conform to a JSON format that looks as follows:

### Sample Code

```
{
  "landingPageUrl": "<Public form URL>"
}
```

## Creating the Communication System and Maintaining the Communication User

The communication between SAP Marketing Cloud and your web server is based on a communication arrangement. Therefore, it's necessary to create a related communication system for the desired target system.

1. Open the *Communication Systems* app and select *New*.
2. Enter a system ID and name and click *Create*.
3. Enter the host name of your web server and any additional details needed.
4. If necessary, create an outbound communication user to be used for the publication service on the web server. You will need to select the appropriate authentication method and enter the related details, for example, a user name and password as well as credentials of a valid user.
5. Be sure to save before creating a communication arrangement.

## Creating the Communication Arrangement

1. Open the *Communication Arrangements* app and select *New*.
2. Select the scenario `SAP_COM_0148`, *Marketing - Form Publication Integration*.
3. Enter a name for the publication target, which will appear on the business user interface, and click *Create*.
4. Select the previously created communication system.
5. Enter the following additional properties:
  - The paths to your form integration service and result headers that you prepared during your form integration
  - The desired folder for the form files as the base directory
  - The paths to the stylesheet (CSS) and JavaScript (JS) files
6. If necessary, select the outbound communication user.
7. Enter the publication path in the section *Outbound Services* under the entry *Deployment Service*.

Once the communication arrangement has been created successfully and is active, the *Forms* content type in the *Content Studio* app will offer the option to publish a form automatically.

## Optional: Implementing Custom Logic for Form Publication

In case the standard publication logic doesn't completely fit your needs, you can adjust it using a custom logic extension point, or Business Add-In (BAI).

### Change File Contents Before Form Publication

The BAI definition *Change File Contents Before Form Publication* makes it possible to automatically adjust the contents of the HTML, CSS, and JS files.

Business Context: *Marketing: Form*

Enhancement Spot: CUAN\_ODATA\_CONTENT\_PAGE

BAI Definition: CUAN\_CP\_DEPLOY\_ADJUST\_FILES

## Related Information

[Custom Integration of Forms \[page 92\]](#)

## 4.2.3 Survey

The documentation for survey explains the following:

- How to integrate survey metadata and survey responses from third-party survey tools into SAP Marketing Cloud using an OData service. For more information, see [Survey OData API \[page 890\]](#).
- How to integrate survey metadata and survey responses from third-party survey tools into SAP Marketing Cloud using integration flows. For more information, see [Integration with Third-Party Survey Providers \[page 107\]](#).

### 4.2.3.1 Integration with Third-Party Survey Providers

Integration of Survey Data with SAP Marketing Cloud using SAP Cloud Integration.

For more information about the Survey Data Integration with SAP Marketing Cloud, see [Third Party Survey Data Integration with SAP Marketing Cloud](#).

By supporting the integration of survey data with SAP Marketing Cloud, customers can benefit from the features of third-party tools, such as Qualtrics, Clicktools, SurveyMonkey, SurveyGizmo, and so on. This integration fetches and stores data easily from the third-party tools into SAP Marketing Cloud system.

To achieve this integration, the following iFlows are provided:

- Create Survey Data in SAP Marketing Cloud.
- Retry Loading Buffered Survey Data to SAP Marketing Cloud.

For more information, see [Integrating Survey Data with SAP Marketing Cloud](#).

## 4.2.4 Extensions

### 4.2.4.1 Import CSV Using SAP Cloud Integration (Deprecated)

With this integration you can do file-based data load to your SAP Marketing Cloud system. The data is fetched from an SFTP server or, alternatively, posted via HTTP request, and pushed to your system using an OData service.

The package enables you to load the following message types and also provides sample CSV templates:

- Interactions
- Accounts
- Contacts

For more information, see the integration package in the SAP API Business Hub under [SAP Marketing Cloud - File-Based Data Load](#).

## 4.3 Outbound

### [Sending Emails and Text Messages \[page 109\]](#)

The integration enables you to send emails and text messages using service providers, such as Sinch.

### [Setting Up External Campaign Execution \[page 155\]](#)

SAP Marketing Cloud allows you to execute campaigns in an external system, and to request the success data for further processing in SAP Marketing Cloud.

### [Open Channel Integration \[page 194\]](#)

With this integration you create own actions that send data for further processing to an external system, such as SAP Business Technology Platform, when the campaign has been executed. But you can also just implement the inbound side of this integration to get external data in your campaigns.

### [Mobile, Social, and Digital Channel \[page 242\]](#)

### [Setting Up Captcha Configuration for Forms \[page 264\]](#)

Use captcha configuration to enhance the security of your forms and decrease vulnerability to malicious attacks by bots that send fraudulent contact data into your system.



## 4.3.1 Sending Emails and Text Messages

The integration enables you to send emails and text messages using service providers, such as Sinch.

You find the list of integration options under [Service Provider and Available Features \[page 111\]](#).

For information about how to set up the integration with a service provider, see [Setting Up Service Provider for Emails and Text Messages \[page 110\]](#).

### Campaign Processing in Detail

In the following you find an example of detailed processing steps for an automated campaign in the system. The campaign uses a dynamic target group and has the actions *Send Email* and *Open Channel* assigned.

When the user starts the campaign, the activation runs some consistency checks, for example, the system checks whether all actions and their parameters correct. The activation also schedules one or more background jobs which are started at the defined execution date and time.

Because the campaign uses a dynamic target group the activation schedules the following background jobs:

- The first job rebuilds the dynamic target group. It is scheduled for the defined execution date and time.
- After the first background job has been finished the two other jobs are processing the actions *Send Email* and *Open Channel* in parallel as successor.

Action job **Send Email** does the following steps:

- The job reads the members of target group and split them into smaller packages, for example, into target groups of 500 members.
- Packages are processed in parallel tasks in order to reduce the runtime of the complete action and use the available resources of the system in an optimal way.  
Steps within package:
  - Check permission
  - Create personalized emails
  - Send out emails by handing over them to the email service provider
  - Create interactions for all target group members, also for those where email cannot be sent, because, for example, marketing permission is missing.
- After all packages have been executed, the execution status is updated and the background job ends.

Action job **Open Channel** does the following steps:

- The job reads the members of target group and split them into smaller packages, for example, into target groups of 500 members.
- Usually packages are processed in parallel tasks in order to reduce the runtime of the complete action and use the available resources of the system in an optimal way.  
Steps within package:
  - Check permission (if activated by customer extension)
  - Read values of attributes defined in export definition
  - Transfer personalized payload to SAP Cloud Integration
  - Create interactions for all target group members (if activated by customer extension)
- After all packages have been executed, the execution status is updated and the background job ends.

After all background jobs have been finished the campaign's execution status is updated.

See also [Open Channel Integration \[page 194\]](#).

### 4.3.1.1 Setting Up Service Provider for Emails and Text Messages

In the following you will find information about how to conduct your system for SAP Marketing Cloud with the required service providers, such as Sinch, for sending out emails and text messages directly out of the system.

Read the following chapters to set up the connection with an email or text message provider.

#### i Note

A system of SAP Marketing Cloud enables you to design and organize marketing campaigns. But to reach your customers, you also need email and text message service provider who take over the data from the SAP system and finally send the text messages and emails.

This service provider must be able to send **mass emails and text messages** for **marketing campaigns**, and should also collect bounces and unsubscribes. These providers are also called **marketing service providers (MSPs)**.

Don't mix them up with your email and cell phone providers for normal communication.

#### [Service Provider and Available Features \[page 111\]](#)

The table gives you an overview about the features available for each service provider.

#### [Setting Up Sinch \[page 112\]](#)

With Sinch as service provider, you send mass emails and text messages to your customers and inform them, for example, about your new developments. With this setup you are also enabled to get bounces and complaints for emails, and receive bounces and unsubscribes for text messages.

#### [Setting Up a Generic Email and Text Message Interface \[page 116\]](#)

With this generic email and text message interface, you can conduct any email or text message service provider to a system of SAP Marketing Cloud to send mass messages. With this setup, you're also enabled to get bounces.

#### [Amazon Setup \[page 136\]](#)

The integration with Amazon is very powerful and covers a wide span of functionality for email and text message campaigns.

#### [Usage of Multiple Service Provider Instances \[page 146\]](#)

In case you want to run campaigns for different customers you can use several instances of the same service provider to gain a better overview about your figures and costs.

#### [Campaign Execution Inclusion List \[page 149\]](#)

You use the *Campaign Execution Inclusion List* app to maintain allowed email addresses and telephone numbers.

#### [Unsubscribe for Emails and Text Messages \[page 150\]](#)

In the following, you learn about the possibilities to unsubscribe from emails and text messages.

#### [Complaints for Emails \[page 152\]](#)

Complaints for email means that an email recipient classifies emails from dedicated senders as spam. For classifying emails as spam, the email recipient either drops the email to the spam folder of the email provider or declares the email as spam. This technology is also known as email feedback loops.

[Troubleshooting for Campaigns \[page 154\]](#)

In case you have issues with the execution of your campaigns, we recommend to read also the troubleshooting in the **Administration Guide**.

[Sender Profiles \[page 155\]](#)

A sender profile allows you to carry out campaigns for different channels in different markets. You can maintain sender profiles for channels, such as email, text message, and mobile push notifications.

### 4.3.1.1.1 Service Provider and Available Features

The table gives you an overview about the features available for each service provider.

#### i Note

A system of SAP Marketing Cloud enables you to design and organize marketing campaigns. But to reach your customers, you also need email and text message service provider who take over the data from the SAP system and finally send the text messages and emails.

This service provider must be able to send **mass emails and text messages** for **marketing campaigns**, and should also collect bounces and unsubscribes. These providers are also called **marketing service providers (MSPs)**.

Don't mix them up with your email and cell phone providers for normal communication.

Feature	Sinch	Amazon	Generic Interface
Documentation	<a href="#">Setting Up Sinch [page 112]</a>	<a href="#">Setting Up Amazon [page 138]</a>	<a href="#">Setting Up a Generic Email and Text Message Interface [page 116]</a>
<b>Sending Emails</b>	Yes Scenario ID: SAP_COM_0040	Yes Scenario ID: SAP_COM_0016	Yes Scenario ID: SAP_COM_0234
Bounces for Emails	Yes Scenario ID: SAP_COM_0040	Yes Scenario ID: SAP_COM_0039	Yes Scenario ID: SAP_COM_0234
Complaints for Emails	Yes Scenario ID: SAP_COM_0040	Yes Scenario ID: SAP_COM_0039	Yes Scenario ID: SAP_COM_0234
Unsubscribe for Emails	Yes Scenario ID: SAP_COM_0040	Yes Scenario ID: SAP_COM_0289	Yes Scenario ID: SAP_COM_0234

Feature	Sinch	Amazon	Generic Interface
Best Sending Time (for emails)	Yes	No	Yes (if supported by connected email service provider)
<b>Sending Text Messages</b>	Yes Scenario ID: SAP_COM_0041	No	Yes Scenario ID: SAP_COM_0258
Bounces for Text Messages	Yes Scenario ID: SAP_COM_0299	No	Yes Scenario ID: SAP_COM_0258
Unsubscribes for Text Messages	Yes Scenario ID: SAP_COM_0299	No	Yes Scenario ID: SAP_COM_0258

## Related Information

[Unsubscribe for Emails and Text Messages \[page 150\]](#)

### 4.3.1.1.2 Setting Up Sinch

With Sinch as service provider, you send mass emails and text messages to your customers and inform them, for example, about your new developments. With this setup you are also enabled to get bounces and complaints for emails, and receive bounces and unsubscribes for text messages.

## Context

As a preparation we recommend to read the following documents and recommendations in order to not get listed as a spammer [Deliverability Best Practices](#) .

To be prepared for the onboarding for Sinch as email provider, see [Sinch E-Mail 365 - Onboarding Guide \(SAP Marketing Cloud customers\)](#) and fill out the form provided by Sinch on this page: [Sinch E-Mail 365 – Provisioning Form](#) .

To further ensure proper email deliverability, it is also important to consider IP warm-up. For more on this topic, see [CX Works | E-mail Marketing - Trust with a Solid Warm-up Plan](#) .

### i Note

For already existing customers: If you are changing something in the settings for your system connection after the upgrade from a lower release to release 1902 or higher, you must re-enter the following data:

- Credentials in the [Communication Systems](#) app

- *Path* in the *Communication Arrangements* app under *Outbound Services*

This is valid for the communication arrangement with scenario ID `SAP_COM_0040` and `SAP_COM_0041`.

## Procedure

Follow the steps below to get your service provider up and running:

1. Set up *Sinch*:
  - You have a *Sinch* account for:
    - sending emails using http service call
    - sending text messages
    - receiving bounces and unsubscribes for text messages

If you need more details or have questions on this solution, send an email to *Sinch* at [DI.email365.onboarding@sinch.com](mailto:DI.email365.onboarding@sinch.com).
  - Check that you got a *User* and a *Password* as well as a *Host* and a *Path Prefix* from *Sinch*.
2. To be able to establish the system connection, check that the apps *Communication System* and *Communication Arrangements* are assigned to your user.
3. Set up the communication system in the *Communication System* app.

### i Note

Use the data from the onboarding material for a proper set-up.

The technical details below may differ from the onboarding material you got from Sinch.

- **Communication System for Emails**  
*System ID*: Enter a system ID, such as `SINCH_EMAIL`.  
*System Name*: Enter a system name, such as `Sinch Email`.  
*Host Name*: `email-eu1.sapdigitalinterconnect.com`  
*User for Outbound Communication*: Enter the user name and password you got from Sinch.
  - **Communication System for Text Messages**  
*System ID*: Enter a system ID, such as `SINCH_TEXT_MESSAGE`  
*System Name*: Enter a system name, such as `Sinch Text Message`.  
*Host Name*: `sms-pp.sapmobileservices.com`  
*User for Outbound Communication*: Enter the user name and password you got from Sinch.
  - **Communication System Details Required for Bounces and Unsubscribes for Text Messages**  
*System ID*: Enter a system ID, such as `SINCH_TEXT_MESSAGE_BOUNCE_UNSUBSCRIBE`  
*System Name*: Enter a system name, such as `Sinch Text Message: Bounce and Unsubscribe`.  
*Host Name*: `livelink.sapmobileservices.com`  
*User for Outbound Communication*: Enter the App Key as *User Name* and the Secret as *Password*.
- In case you need to change user and password again later, you will do this also in the *Communication System* app.
4. Establish the communication arrangement in the *Communication Arrangement* app.
    1. To create a new arrangement choose *New*, select the required *Scenario* and enter an *Arrangement Name*.

Technical details for the communication arrangement:

- **Communication Arrangement for Emails**  
*Scenario:* **SAP\_COM\_0040**
- **Communication Arrangement for Text Messages**  
*Scenario:* **SAP\_COM\_0041**
- **Communication Arrangement for Bounces and Unsubscribes for Text Messages**  
*Scenario:* **SAP\_COM\_0299**

2. Select the previously created *Communication System*.
3. Depending on the used setup enter the following *Additional Properties*: Enter a *Provider ID* and *Sender Profile ID*, and assign a suitable *Marketing Area ID*.

Note that you use for the setup of **SAP\_COM\_0041** and **SAP\_COM\_0299** the same *Provider ID*.

4. Under *Outbound Communication* select the *User Name*, you created in the *Communication System* app, you got from Sinch.
5. Under *Outbound Services* check that the *Service Status* is activated, and *Port 443* is used.

In addition, maintain the *Path* as followed:

- *Scenario:* **SAP\_COM\_0040**  
*Path* depending on the information you got from Sinch: **/in365-api/<accountID>/notifications** or **/email/<accountID>/notifications**
- *Scenario:* **SAP\_COM\_0041**  
*Path:* such as **/cmn/<accountID>/<accountID>.sms**
- *Scenario:* **SAP\_COM\_0299**  
*Path:* no path

6. Now save your entries.

During the save the system establishes the required system connections and creates a provider and sender profile.

5. Finally maintain your sender profiles. For more information, see [Sender Profiles \[page 155\]](#).

## Using Several Accounts

If you want to use several accounts, you must do the steps above for each account separately. To get more information about the dependencies in the setup, see [Usage of Multiple Service Provider Instances \[page 146\]](#).

## Related Information

[Consuming the Integration APIs \[page 395\]](#)

## 4.3.1.1.2.1 Bounces and Unsubscribe for Text Messages

When you want to use unsubscribe and bounces offered by Sinch, you need a connection between your SAP system and Sinch.

### Prerequisites

You set up the connection to Sinch. For more information, see [Setting Up Sinch \[page 112\]](#).

In addition, you have also set up the scenario [SAP\\_COM\\_0041](#) and [SAP\\_COM\\_0299](#).

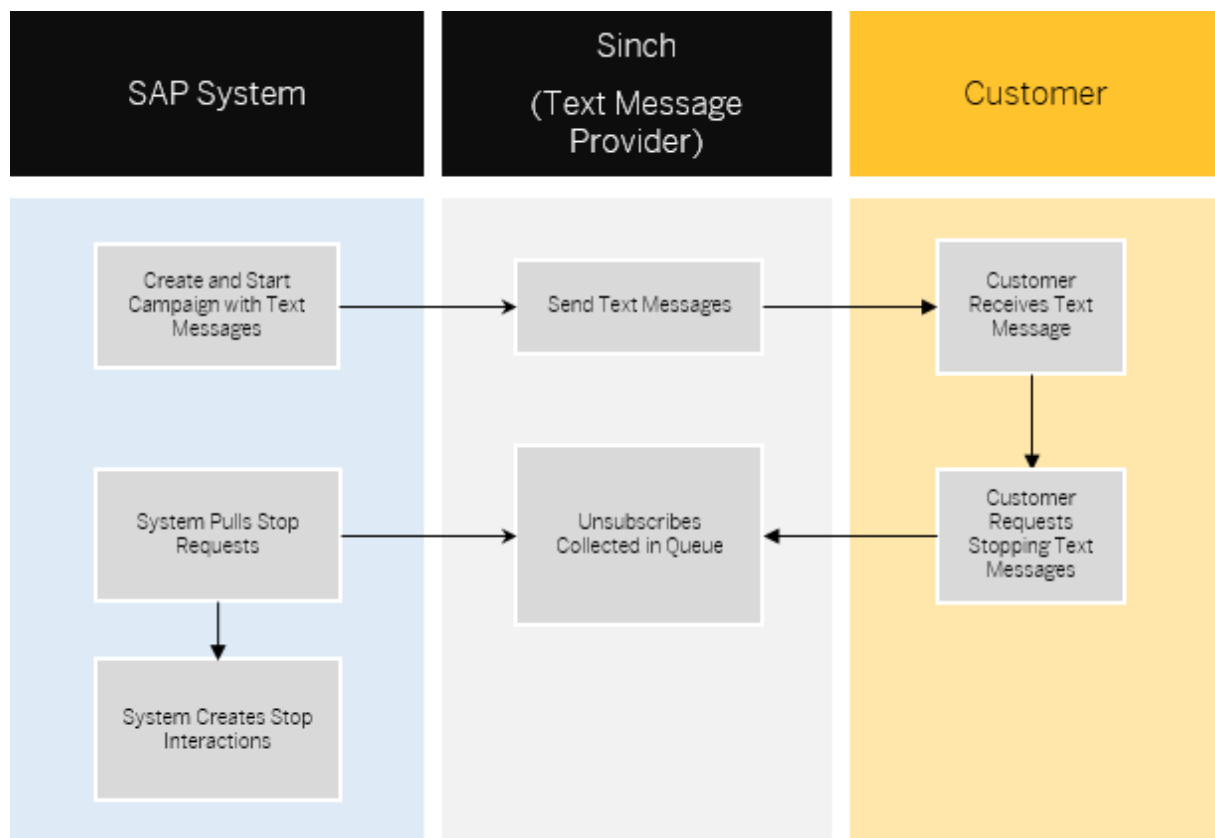
### How It Works

After you did all the settings, the recipients of text messages can unsubscribe and you can collect bounces for text messages.

#### Unsubscribe

If the contact does not want to get further text messages, she or he has to send a text message with the word **STOP** as reply to the received text message. These unsubscriptions can happen at any time.

The unsubscribe requests are collected on Sinch's side in a queue. A background job then pulls the unsubscribe requests from Sinch and creates corresponding interactions in the system. The system evaluates the interactions and updates marketing permissions for the contact.



In detail the following steps happen:

1. The marketing expert executes a text message campaign.
2. The system sends out the marketing text messages.
3. A recipient is getting the text message on the mobile.
4. The recipient unsubscribes by sending the word **STOP** as reply to the received text message. Optionally, the recipient can send back the word **STOP** plus the campaign ID to unsubscribe from a specific campaign with a specific marketing area. Prerequisite is that the marketing area separation is active and the campaign ID is part of the sent text message, ideally using personalization attributes in the *Content Studio*.  
At the end an interaction with type `MKT_PERM_OPTOUT` and with the marketing area of this campaign is created.
5. The mobile service provider sends the text message with the unsubscribe request to Sinch.
6. Sinch collects unsubscribe requests and bounces in a queue.
7. SAP pulls the unsubscribe requests and bounces, and creates interactions.
8. Based on the interactions the system updates marketing permissions.

## Bounces

To see the number of hard and soft bounces, open the corresponding campaign in the *Campaigns* app. On the *Performance* tab, you see the actuals under *Outbound*.

For more information regarding how to handle bounces, see [Handling Bounces](#).

For more information regarding bounce classification, see [Email-360 - Bounce Classification - Sinch Community- 3532](#) .

## Related Information

[Sinch Live Link 365](#) .

[Marketing Areas](#)

[Setting Up Sinch \[page 112\]](#)

### 4.3.1.1.3 Setting Up a Generic Email and Text Message Interface

With this generic email and text message interface, you can conduct any email or text message service provider to a system of SAP Marketing Cloud to send mass messages. With this setup, you're also enabled to get bounces.

SAP Marketing Cloud offers a generic interface that transfers the message header and body data in a JSON format.

The system calls

- either the service provider directly, if it supports the defined JSON interface natively,
- or an integration engine like SAP Business Technology Platform (SAP BTP) to transform the message from SAP format into any kind of legacy formats. The integration engine is optional in this scenario even though it's the most likely use case.



## i Note

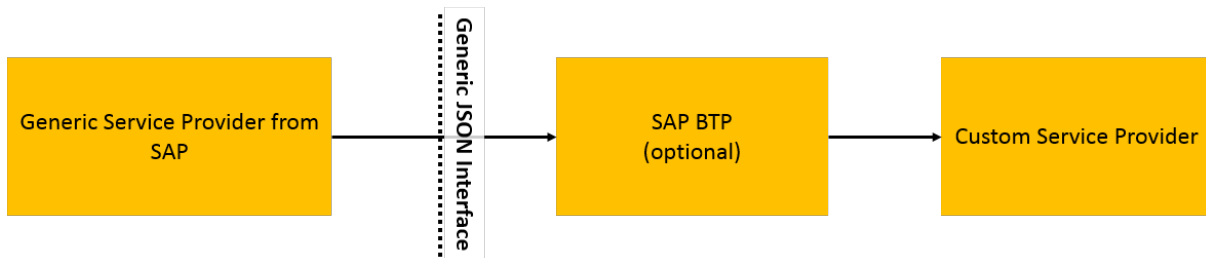
A system of SAP Marketing Cloud enables you to design and organize marketing campaigns. But to reach your customers, you also need email and text message service provider who take over the data from the SAP system and finally send the text messages and emails.

This service provider must be able to send **mass emails and text messages** for **marketing campaigns**, and should also collect bounces and unsubscribes. These providers are also called **marketing service providers (MSPs)**.

Don't mix them up with your email and cell phone providers for normal communication.

You've prepared the following data:

- Connection data to connect SAP Marketing Cloud with SAP BTP:
  - hostname of SAP BTP
  - logon credentials such as user and password for the outbound communication to SAP BTP  
Instead of user and password, you can also work with certificates. You activate the certificates in the [Communication Systems](#) app during the setup.
- Connection data to connect SAP BTP with your service provider:
  - hostname of your service provider
  - logon credentials such as user and password
- **Optional:** Depended on the service provider, you must upload the provider's certificates to the SAP BTP. To upload the certificates on the platform, choose [Operations View](#) > [Manage Keystore](#) > [Add](#) > [Certificate](#) and upload the certificate.



In addition, you have access to the following apps:

- [Communication Systems](#)
- [Communication Arrangements](#)
- [Sender Profiles](#)

## Set Up with User and Password

### i Note

The following steps describe the setup by usage of SAP BTP.

1. In the [Communication Systems](#) app, you create the system entry by entering:
  - host name of SAP BTP

- logon credentials such as user and password under *User for Outbound Communication*
2. In the *Communication Arrangements* app, create a new communication arrangement with the Scenario ID SAP\_COM\_0234 or SAP\_COM\_0258 and a name.

Add the communication system from the previous step and make sure that

- you activated the *Service Status* and
- you entered the *Path* under *Outbound Services*.

### i Note

The path doesn't need any path enhancement, such as `/send`, `/bounces`, `/complaints`, or `/verifiedSenders`.

### ❁ Example

In SAP BTP you've defined an iFlow with the following settings:

Type of the adapter = HTTPS

Connection address = `/sap_mkt_cloud/send`

In the communication arrangement, you must enter the *Path* `/http/sap_mkt_cloud` without the path enhancement `/send`.

The *Service URL* looks, for example, like `https://<hostname of SAP Cloud Integration>:443/http/sap_mkt_cloud`.

3. After the activation of your communication arrangement, choose the *Sender Profiles* app and complete the incomplete entries:
  - *Profile ID* GNML for emails
  - *Profile ID* GENS for text messages
 Open the profile GNML and enter a valid email address as *Sender Address*.

## Set Up with Certificates

### i Note

The following steps describe the setup by usage of SAP BTP.

1. In the *Communication Systems* app, you create the system entry by entering host name of SAP BTP and choose the *Authentication Method* **SSL Client Certificate** under *User for Outbound Communication*.
2. In the *Communication Arrangements* app, create a new communication arrangement with the Scenario ID SAP\_COM\_0234 or SAP\_COM\_0258 and a name. Add the communication system from the previous step and make sure that
  - you activated the *Service Status* and
  - you entered the *Path* under *Outbound Services*.

## i Note

The path doesn't need any path enhancement, such as `/send`, `/bounces`, `/complaints`, or `/verifiedSenders`.

## ❁ Example

In SAP BTP you've defined an iFlow with the following settings:

Type of the adapter = HTTPS

Connection address = `/sap_mkt_cloud/send`

In the communication arrangement, you must enter the *Path* `/http/sap_mkt_cloud` without the path enhancement `/send`.

The *Service URL* looks, for example, like `https://<hostname of SAP Cloud Integration>:443/http/sap_mkt_cloud`.

3. To use the certificates, you have to download them in the *Communication Arrangements* app for the corresponding communication arrangement under **▶ Outbound Communication ▶ Download ▶ Download Authentication Certificate ▶**.
4. Then switch to the SAP BTP, choose **▶ Design ▶ Artifacts ▶**. Then open the *iFlow Adapter* and select the outbound communication channel that needs the certificate.  
Under **▶ Connection ▶ Authorization ▶**, select **Client Certificate**, choose **▶ Add ▶ Select ▶**, and upload the certificate from the *Communication Arrangements* app.  
Don't forget to save and deploy the iFlow.
5. After the activation of your communication arrangement, choose the *Sender Profiles* app and complete the incomplete entries:
  - *Profile ID* GNML for emails
  - *Profile ID* GENS for text messagesOpen the profile GNML and enter a valid email address as *Sender Address*.

## More Information

- [Consuming the Integration APIs \[page 395\]](#)
- [SAP Marketing Cloud – Connect any email service provider to SAP Marketing Cloud](#)
- [SAP Marketing Cloud – Connect any text message service provider to SAP Marketing Cloud](#)
- [Connecting a Customer System to Cloud Integration](#)
- [Cloud Integration – How to Setup Secure HTTP Inbound Connection with Client Certificates](#)

### 4.3.1.1.3.1 Generic Email and Text Message Integration

With this REST service and methods you integrate any email and text message service provider (SP) with a system of SAP Marketing Cloud. For email you use scenario ID [SAP\\_COM\\_0234](#) and for text message you use [SAP\\_COM\\_0258](#).

Using the campaign automation, you can send emails or text messages to your customers.

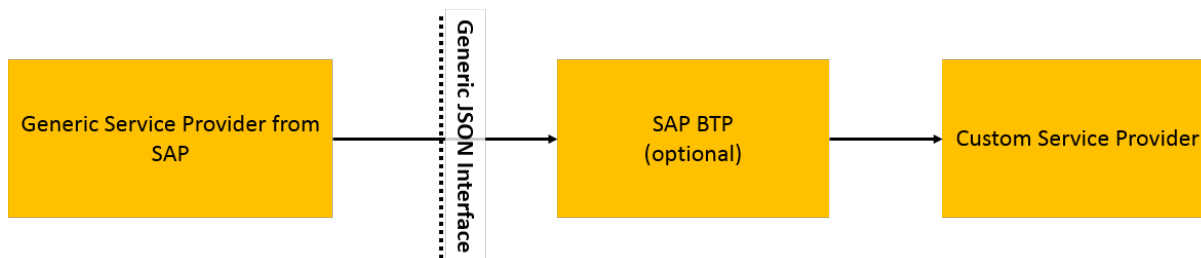
The email bodies contain personalized content and trackable links. Countable interactions are, for example, email opened, link clicked, and email hard bounce. Emails are sent using a REST service to an email SP.

The text messages contain personalized content. Text messages are sent using a REST service API to an SP for text messages.

The integration is a pure outbound scenario. You require an account and license on the email or text message SP's side.

#### ⚠ Caution

When you connect an email or text message service provider, it's your responsibility to establish the connection by implementing the methods defined by SAP. This is also valid for required enhancements on the SAP Business Technology Platform (SAP BTP), such as doing the mapping or persisting data.



## Methods for Email Integration

HTTP Method	Action	Path Enhancement (that must be the same in SAP BTP)
Send Emails <i>POST</i>	<b>Mandatory</b> method <a href="#">Email: Send Emails [page 121]</a>	/send
Get Bounces <i>GET</i>	<a href="#">Email: Get Bounces [page 125]</a>	/bounces
Get Complaints <i>GET</i>	<a href="#">Email: Get Complaints [page 126]</a>	/complaints
Get Unsubscribes <i>GET</i>	<a href="#">Email: Get Unsubscribes [page 127]</a>	/unsubscribes

HTTP Method	Action	Path Enhancement (that must be the same in SAP BTP)
Get Verified Senders <i>GET</i>	<b>Mandatory</b> method <a href="#">Email: Get Verified Senders [page 129]</a>	/verifiedSenders

## Methods for Text Message Integration

HTTP Method	Action	Path Enhancement (that must be the same in SAP BTP)
Send <i>POST</i>	<b>Mandatory</b> method <a href="#">Text Message: Send [page 130]</a>	/send
Collect Delivery Status <i>GET</i>	<a href="#">Text Message: Collect Delivery Status [page 132]</a>	/status
Get Unsubscribes <i>GET</i>	<a href="#">Text Message: Get Unsubscribes [page 134]</a>	/unsubscribes

### 4.3.1.1.3.1.1 Email: Send Emails

With this method you send the emails to your email service provider (ESP).

#### Note

This method is **mandatory** for the integration.

## Request

URI: /send

HTTP Method: *POST*

## Request Parameters

Parameter	Required	Data Type	Description
bodyContentHTML	Yes	String	Body Content. Format: HTML, JSON encoded

Parameter	Required	Data Type	Description
bodyContentPlainText	Yes	String	Body Content (for multipart or alternative email MIME). Format: Plain Text, JSON encoded
campaignId	No	String	<i>Campaign ID</i> of the campaign that generates this email. Can be empty for send tests in campaign content and sender profile. Helpful for support.
listUnsubscribe	No	String	Header for list unsubscribe in raw format. Possible entries: <code>mailto:&lt;email address&gt;</code> , <code>https://&lt;URL&gt;</code>
outboundId	No	String	Unique identifier of outbound message generated by SAP Marketing Cloud
recipient	Yes	String	Recipient ( <i>To</i> field)
recipientName	No	String	Recipient name. Not yet supported. Format: Plain Text, JSON encoded
replyTo	Yes	String	Reply-To Address ( <i>Reply-To</i> field)
replyToName	Yes	String	Reply-To Name. Format: Plain Text, JSON encoded
sendAt	No	String	Timestamp for scheduled sends. Format: YYYYMMDDHHMMSS
sender	Yes	String	Sender Address ( <i>From</i> field)
senderName	Yes	String	Sender Name. Format: Plain Text, JSON encoded
sourceSystem	No	String	Logical System (Netweaver). Required to get corresponding bounces and complaints related to outbound messages.
subjectContentPlainText	Yes	String	Subject; Format: Plain Text, JSON encoded
type	No	String	Indicates for the middleware which integration flow for which type of ESP should be processed

## Request Example

### Sample Code

```
/send
HTTP method POST

Content-Type:
application/json
Encoding: UTF-8

Body:
{
  "type" : "email",
  "outboundId" : "33fds34534r4",
  "campaignId" : "123456789",
  "sourceSystem" : "XYZCLINT100",
  "sendAt" : "20170328080000",
  "sender" : "john.miller@example.com",
  "senderName" : "John Miller",
  "replyTo" : "news@example.com",
  "replyToName" : "SAP News",
  "recipient" : "recipient@example.com",
  "recipientName" : "Recipient",
  "listUnsubscribe" : "<mailto:yyy>, <http::zzz>",
  "subjectContentPlainText" : "Hello Recipient",
  "bodyContentHtml" : "<b>Hello, this email body is HTML<b>",
  "bodyContentPlainText" : "Hello, this email body is plain text"
}
```

## Response

### Response Parameters

Parameter	Required	Data Type	Description
errorCategory	No	String	Permanently appearing errors lead to a stop of the campaign execution. Retriable errors result in multiple retries to resolve the issue before the campaign stops. Throttling reduces the throughput that is generated by the backend. Possible values are Retriable, Permanent, or Throttling.
errorText	No	String	Error text is written to the log and shown to the end user. Format: Plain Text
messageId	Yes	String	Unique identifier for outbound message provided by ESP. Could be 'outboundId' if supported by ESP, but not necessarily.

## Response Codes

- The response refers only to the email sent using the connected service provider or SAP Business Technology Platform. You can't get bounces, such as email address is not valid, with the response. For more information, see [Email: Get Bounces \[page 125\]](#).
- Note that the success code must start with 2 followed by two digits, for example, 202.
- For erroneous responses the following generic codes are used:
  - 401 Unauthorized: This code stops the running campaign.
  - 403 Forbidden: This code stops the running campaign.
  - 429 Too Many Requests: With this code the system throttles the email delivery, such as in the case of the throttling for Amazon: [How the System Reacts on Amazon's Throttling \[page 145\]](#).

### i Note

Keep in mind that these codes will only work when the `errorCategory` is empty. The `errorCategory` has always a higher priority then the error codes.

## Response Example

### Sample Code

```
Success Code 202

Content-Type:
application/json
{
  "messageId": "33fds34534r4"
}
```

## Error Example

### Sample Code

```
Error Codes 4xx, 5xx:
{
  "errorCategory" : "Retriable"
  "errorText"    : "Sending messages failed and can be retried."
}
```

### Sample Code

```
Error Codes 4xx, 5xx:
{
  "errorCategory" : "Permanent"
  "errorText"    : "Messages cannot be sent."
}
```

### Sample Code

```
Error Codes 4xx, 5xx:
{
  "errorCategory" : "Throttling"
  "errorText"    : "Throughput for sending messages is too high. Sending
messages can be throttled."
```



```
}
```

### 4.3.1.1.3.1.2 Email: Get Bounces

With this method you request the bounces from your connected email service provider (ESP).

#### Request

URI: `/bounces`

HTTP Method: `GET`

#### Request Parameters

Parameter	Required	Data Type	Description
<code>sourceSystem</code>	No	String	Logical system that is required to get bounces and complaints related to outbound messages.
<code>startTimeUTC</code>	Yes	String	Timestamp to begin with query collected bounces on ESP side. Format: YYYYMMDDHHMMSS
<code>endTimeUTC</code>	Yes	String	Timestamp to end with query collected bounces on ESP side. Format: YYYYMMDDHHMMSS
<code>page</code>	No	Integer	Indicates the result page in case of multiple pages. Possible values are 0 to n

#### Request Example

```
/bounces
GET
sourceSystem=XYZCLNT100
startTimeUTC=20181115221500
endTimeUTC=20181115223000
Page=2
```

#### Response

##### Response Parameters

Parameter	Required	Data Type	Description
<code>page</code>	No	Integer	Indicates the result page in case of multiple pages. Possible values are 0 to n
<code>lastPage</code>	Yes	Boolean	Indicates the last page of the result.

Parameter	Required	Data Type	Description
bounces	Yes	JSON Array	Contains the bounce details.
messageId	No	String	Reference to outbound message, see sending interface.
recipient	Yes	String	The email address that bounced or created a complaint. Value: <email address>
type	Yes	String	Type of feedback, depending on bounce or complaint use case. Possible values are: Hard, Soft, abuse, fraud, virus, other, not-spam
timestamp	No	String	Timestamp when bounce occurred. Format: YYYYMMDDHHMMSS
errorCode	No	String	Error code for bounces. Possible entries are: DSN error code (X.Y.Z), SMTP error code (XYZ)
errorText	No	String	Error text for bounce message.

### Response Example

```
{
  "page": "2",
  "lastPage": "false",
  "bounces": [
    {
      "messageId" : "12343243243413",
      "recipient" : "bounce@example.com",
      "errorCode" : "5.1.1.",
      "errorText" : "Address does not exist",
      "Type" : "Hard",
      "Timestamp" : "20181116093500"
    }
  ]
}
```

## 4.3.1.1.3.1.3 Email: Get Complaints

With this method you request the complaints from the connected email service provider (ESP).

For the parameter values please refer to [Email: Get Bounces \[page 125\]](#).

### Request

URI: /complaints

HTTP Method: *GET*

## Request Example

```
/complaints
GET
sourceSystem=XYZCLNT100
startTimeUTC=20180925211500
endTimeUTC=20180925213000
Page=2
```

## Response

### Response Example

```
{
  "page": "2",
  "lastPage": "false",
  "complaints": [
    {
      "messageId" : "12343243243413",
      "recipient" : "spam@example.com",

      "Type"      : "abuse",
      "Timestamp" : "20180926093500"
    }
  ]
}
```

## 4.3.1.1.3.1.4 Email: Get Unsubscribes

With this method you request the unsubscribes from your connected email service provider (ESP).

## Request

URI: /unsubscribes

HTTP Method: *GET*

### Request Parameters

Parameter	Required	Data Type	Description
sourceSystem	No	String	Logical system that is required to get unsubscribes related to outbound messages.
startTimeUTC	No	String	Timestamp in UTC to begin with query collected unsubscribes on ESP side. Not required for bounce queue. Format: YYYYMMDDHHMMSS

Parameter	Required	Data Type	Description
endTimeUTC	No	String	Timestamp in UTC to end with query collected unsubscribes on ESP side. Not required for bounce queue. Format: YYYYMMDDHHMMSS
page	No	Integer	Indicates the result page in case of multiple pages. Possible values are 0 to n

## Request Example

```
/unsubscribes
GET
sourceSystem=XYZCLNT100
startTimeUTC=20180815064512
endTimeUTC=20180815073422
Page=2
```

## Response

### Response Parameters

Parameter	Required	Data Type	Description
page	No	Integer	Indicates the result page in case of multiple pages. Possible values are 0 to n
lastPage	Yes	Boolean	Indicates the last page of the result.
unsubscribes	Yes	JSON Array	Contains the unsubscribe details.
outboundId	No	String	Reference to outbound message, see sending interface. Unique identifier of outbound message generated by SAP Marketing Cloud .
messageId	No	String	This is the ESP-specific message ID provided by the generic provider.
recipient	Yes	String	The email address that unsubscribed. Value: <email address>
timestamp	Yes	String	Timestamp in UTC when unsubscribe occurred. Format: YYYYMMDDHHMMSS

### Response Example

```
{
  "page": "2",
  "lastPage": "false",
  "unsubscribes": [
    {
      "messageId" : "12343243243413",
      "recipient" : "unsubscribe@example.com"
    },
    {
      "timestamp" : "20180817163255"
    }
  ]
}
```

In the SAP system, a multi-level approach is implemented and, for example, the `outboundId` is evaluated first. But in case the `outboundId` is not provided and is, for example, initial, the `messageId` is evaluated. And in case, the `messageId` is not provided, too, email address is evaluated (recipient).

#### i Note

Note: If one of the regarding values (`outboundId`, `messageId`) is not initial, the system takes the entries as valid.

A fall back on other levels only happens when the previous level values are initial.

### 4.3.1.1.3.1.5 Email: Get Verified Senders

With this method you get the verified senders from your connected email service provider (ESP).

#### i Note

This method is **mandatory** for the integration.

Only with this method implemented, you can:

- maintain sender profiles
- send test emails
- send emails out of a campaign

#### → Recommendation

We recommend to use your customer domain as `senderDomains` instead of `*`.

## Request

URI: `/verifiedSenders`

HTTP Method: `GET`

### Request Example

```
/verifiedSenders
GET
```

## Response

### Response Example

```
Response
{
  "senders"      : [ "sender1@example.com", "sender2@example.com" ],
  "senderDomains" : [ "news.sap.com", "sap.com", "example.com" ]
}
```

## 4.3.1.1.3.1.6 Text Message: Send

With this method you send text messages to your connected text messaging service provider.

## Request

URI: /send

HTTP Method: *POST*

### Request Parameters

Parameter	Required	Data Type	Description
type	No	String	Indicates for the middleware which integration flow for which type of service provider should be processed; entry 'sms'
outboundId	No	String	Unique identifier of outbound message generated by SAP Marketing Cloud
campaignId	No	String	Campaign ID of SAP Marketing Cloud that generates this email. Can be empty for send tests in campaign content and sender profile. Helpful for support.
sourceSystem	No	String	Logical System (SAP Net-Weaver). Required to get corresponding bounces or complaints related to outbound messages.
sender	Yes	String	Sender address; name or phone number
recipient	Yes	String	Recipient; phone number

Parameter	Required	Data Type	Description
bodyContentPlainText	Yes	String	Body content as plain text and JSON encoded.

## Request Example

### Sample Code

```

Path /send
HTTP method POST

Content-Type: application/json
Encoding: UTF-8

Body:
{
  "type" : "sms",
  "outboundId" : "33fds34534r4",
  "campaignId" : "123456789",
  "sourceSystem" : "ANACLNT100",
  "sender" : "SAP News",
  "recipient" : "+49123456789",
  "bodyContentPlainText" : "Hello, this is plain text"
}

```

## Response

### Response Parameters

Parameter	Required	Data Type	Description
messageId	Yes	String	Unique identifier for outbound message provided by service provider. Could be outboundId if supported by service provider.
errorCategory	No	String	Permanent errors lead to a stop of the campaign execution. Errors that can be fixed by a retry result in multiple retries to resolve the issue before the campaign stops. Throttling reduces the throughput that is generated by the backend. Possible values are <code>Retriable</code> , <code>Permanent</code> , or <code>Throttling</code> .
errorText	No	String	Error text is written to the log and shown to the end user in plain text.

### i Note

- The response refers only to the text message sent using the connected service provider or SAP Business Technology Platform. You can't get bounces, such as phone number is not valid, with the response. For more information, see [Text Message: Collect Delivery Status \[page 132\]](#).
- Note that the success code must start with 2 followed by two digits, for example, 202.

### Response Example

#### Sample Code

```
Response:
Success Code 202
Content-Type: application/json

{
  "messageId": "33fds34534r4"
}
```

### Error Example

#### Sample Code

```
Error Codes 4xx, 5xx:

{
  "errorCategory" : "Retriable"
  "errorText"     : "Internal Server Error"
}
```

## 4.3.1.1.3.1.7 Text Message: Collect Delivery Status

With this method you get back the status of your connected text messages sent.

### Request

URI: /status

HTTP Method: *GET*



## Request Parameters

Parameter	Required	Data Type	Description
sourceSystem	No	String	Logical System (SAP NetWeaver). Required to get corresponding bounces or complaints related to outbound messages.
startTimeUTC	No	String	Timestamp to begin with query collected bounces on service provider side. Not needed in case of a bounce queue; format: YYYYMMDDHHMMSS
endTimeUTC	No	String	Timestamp to end with query collected bounces on service provider side. Not needed in case of a bounce queue; format: YYYYMMDDHHMMSS
page	No	Integer	Indicates the result page in case of multiple pages; possible values: 0..n

## Request Example

```
/status  
  
GET  
sourceSystem=ANACLNT100  
startTimeUTC=20170912144813  
endTimeUTC=20170913144813  
Page=2
```

## Response

### Response Parameters

Parameter	Required	Data Type	Description
page	No	Integer	Indicates the result page in case of multiple pages. Possible values are 0 to n
lastPage	Yes	Boolean	Indicates the last page of the result.
status	Yes	JSON Array	Contains the bounce details.
messageId	No	String	Reference to outbound message, see sending interface.
recipient	Yes	String	The phone number that bounced or created a complaint. Value: phone number
type	Yes	String	Type of feedback, depending on bounce or complaint use case. Possible values are: <code>Permanent</code> or <code>Temporary</code> . Note that the type is mapped in the SAP system as followed:  <code>Permanent</code> - hard bounce  <code>Temporary</code> - soft bounce

Parameter	Required	Data Type	Description
timestamp	No	String	Timestamp when bounce occurred. Format: YYYYMMDDHHMMSS
statusCode	No	String	The status code that your connected service provider sends back to inform you about the delivery status. Keep in mind that <ul style="list-style-type: none"> <li>status codes can't be mapped to statuses in the SAP system and</li> <li>shall not be longer than 10 characters due to further processing.</li> </ul>
errorText	No	String	Error text for bounce message.

### Response Example

#### Sample Code

```
{
  "page": 2,
  "lastPage": false,
  "status": [
    {
      "messageId" : "12343243243413",
      "recipient" : "+49123456789",
      "statusCode" : "10",
      "statusText" : "Number does not exist",
      "type" : "Permanent",
      "timestamp" : "20170913144813"
    }
  ]
}
```

## 4.3.1.1.3.1.8 Text Message: Get Unsubscribes

With this method you request the unsubscribes (also known as 'Stop Trigger') from your connected text message provider.

### Request

URI: /unsubscribes

HTTP Method: *GET*

## Request Parameters

Parameter	Required	Data Type	Description
sourceSystem	No	String	Logical system that is required to get unsubscribes related to outbound messages.
startTimeUTC	Yes	String	Timestamp to begin with query collected unsubscribes on service provider side. Format: YYYYMMDDHHMMSS
endTimeUTC	Yes	String	Timestamp to end with query collected unsubscribes on ESP side. Format: YYYYMMDDHHMMSS
page	No	Integer	Indicates the result page in case of multiple pages; possible values: 0..n

## Request Example

```
/unsubscribes
GET
sourceSystem=ABCCLNT100
startTimeUTC=20180815064512
endTimeUTC=20180815073422
Page=2
```

## Response

### Response Parameters

Parameter	Required	Data Type	Description
page	No	Integer	Indicates the result page in case of multiple pages. Possible values are 0 to n
lastPage	Yes	Boolean	Indicates the last page of the result.
unsubscribes	Yes	JSON Array	Contains the unsubscribe details.
outboundId	No	String	Reference to outbound message, see sending interface. Unique identifier of outbound message generated by SAP Marketing Cloud .
messageId	No	String	Reference to an outbound ID provide by the generic ESP.
recipient	Yes	String	Phone number of the original text message that replied with an unsubscribe (stop trigger). Value: phone number
sender	No	String	Phone number to which the unsubscribe (stop trigger) was sent. Value: phone number
timestamp	Yes	String	Timestamp when unsubscribe occurred. Format: YYYYMMDDHHMMSS
messageText	No	String	Message text that was sent with the unsubscribe, for example, the campaign ID.

## Response Example

### Sample Code

```
{
  "page": 2,
  "lastPage": false,
  "unsubscribes": [
    {
      "outboundId" : "AHGB789345",
      "recipient" : "+49123456789",
      "messageText" : "Stop 0815",
      "timestamp" : "20180913144813"
    }
  ]
}
```

In the SAP system, a multi-level approach is implemented and, for example, evaluates

- `outboundId` first, in case that it is not provided
- `messageId`, in case it is not provided
- mobile number (recipient) only.  
For this level the system evaluates the message text if it contains a campaign ID. In this case the `campaignID` is considered with regards to marketing area separation.

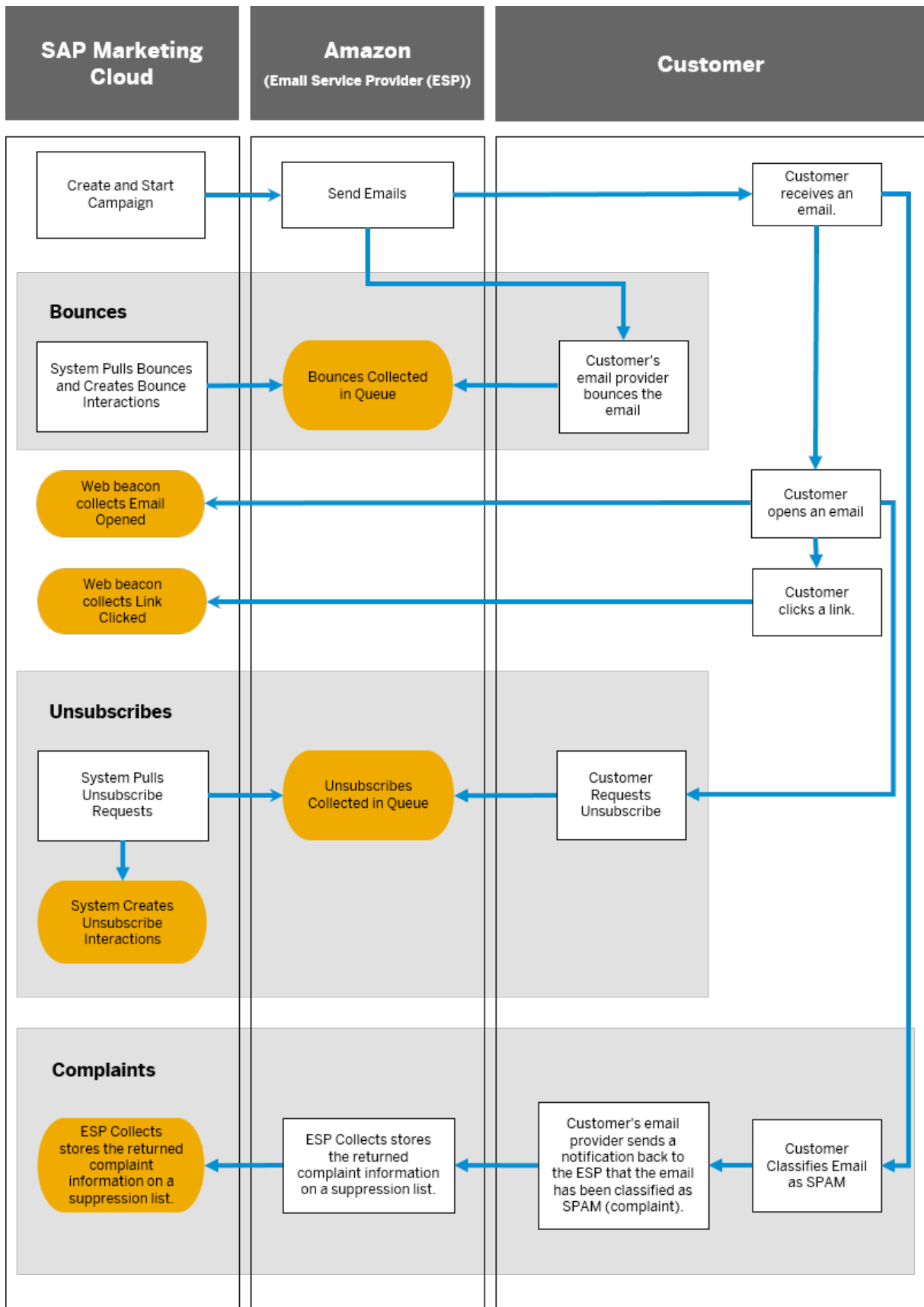
Note: If one of the regarding values (`outboundId`, `messageId`) is not initial, the system takes the values as valid ones.

A fall back on other levels only happens, when the previous level values are initial.

### 4.3.1.1.4 Amazon Setup

The integration with Amazon is very powerful and covers a wide span of functionality for email and text message campaigns.

The following graphic gives you an overview how the integration roughly works and how the data flows. In addition there are links, where suitable for the setup or other useful information.



- [Enabling Automatic Unsubscribe for Emails by Amazon \[page 143\]](#)
- [Setting Up Amazon \[page 138\]](#)
- [Bounces, Unsubscribes, and Complaints](#)
- [Bounces, Unsubscribes, and Complaints](#)

## Related Information

[Service Provider and Available Features \[page 111\]](#)

### 4.3.1.1.4.1 Setting Up Amazon

To establish the connection to Amazon's Simple Email Service (SES) for email and bounce handling, you must do several steps at Amazon and at SAP.

## Prerequisites

The following prerequisites exist for setting up Amazon as an email service provider:

- Use the *Campaign Execution Inclusion List* app during the system setup and test phase. The settings in the app prevent that target group members (contacts) receive emails when you execute an email campaign for test purposes.  
For more information, see [Campaign Execution Inclusion List \[page 149\]](#).
- You are familiar with the basics of the *Amazon Service for Emails, Notifications and Queues*:
  - [aws.amazon.com/de/documentation/ses/](https://aws.amazon.com/de/documentation/ses/)
  - [aws.amazon.com/de/documentation/sns/](https://aws.amazon.com/de/documentation/sns/)
  - [aws.amazon.com/de/documentation/sqs/](https://aws.amazon.com/de/documentation/sqs/)
- You have access granted on Amazon for the following API methods:
  - `SendRawEmail` (SES)
  - `ListIdentities` (SES)
  - `GetIdentityVerificationAttributes` (SES)
  - `GetSendQuota` (SES)
  - `ReceiveMessage` (SQS)
  - `DeleteMessageBatch` (SQS)

For more information, see [Controlling Access to Amazon SES](#) .

## Procedure

1. To send marketing emails to your customers, you have to configure Amazon Web Services (AWS). For more information, see [Setting up Amazon \[page 139\]](#).

2. After you did the configuration at Amazon, you go further with the configuration at SAP. Download the certificates and import them to the SAP system. For more information, see [Importing Certificates \[page 141\]](#).
3. Establish the system connection between Amazon and SAP. For more information, see [Establishing System Connection \[page 141\]](#).

## Next Steps

### i Note

The following blog is not part of the official documentation of SAP Marketing Cloud and some of the information may be outdated.

## Related Information

<http://docs.aws.amazon.com/ses/latest/DeveloperGuide/before-you-begin.html> ↗

<http://docs.aws.amazon.com/ses/latest/DeveloperGuide/configure-sns-notifications.html> ↗

<http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqssubscribe.html> ↗

[How the System Reacts on Amazon's Throttling \[page 145\]](#)

### 4.3.1.1.4.1.1 Setting up Amazon

Here you create an Amazon Web Services (AWS) account and verified email addresses. Then you configure your SNS topics and bounce queues, create Identity and Access Management (IAM) users, generate your credentials for the Simple Email Service (SES), and set up group administration for your users to assign policies.

## Procedure

1. Create an Amazon Web Services (AWS) account to get an account ID and password.
2. Log in to <https://console.aws.amazon.com> ↗ with your credentials.
3. Ensure, to select the correct region you intend to use. You see the region beside your account ID in the AWS Console itself and in the URL of your browser, for example, `https://eu-west-1.console.aws.amazon.com/ses/home?region=eu-west-1#`.
4. In the AWS Console under *Identity Management*, create the required verified sender email addresses under *Email Addresses*.

You need the verified sender address, when you define a sender profile later on.

Keep in mind that the *MAIL FROM Domain* of your verified email addresses must in the same region as the *Host Name* entered in the system connection later on.

For more information, see [Regions and Amazon SES](#) .

5. Now edit the notification configuration for the verified sender email address by choosing ► [Notifications](#) ► [Edit Configuration](#) ► [Click here to create a ne Amazon SNS topic.](#) ►.

An *Edit Notification Configuration* popup opens.

Create a new Amazon SNS topic, for example, for bounces and complaints. Give the new topic the same name as the feedback queue, such as **AMAZON\_BOUNCE**.

6. Then under *SNS Topic Configuration* select the previously created topics for bounces and complaints.

Note that you don't maintain a topic for deliveries. The system is not able to handle the delivery notification.

7. Also in the *Edit Notification Configuration* popup under *Email Feedback Forwarding*, disable the email feedback forwarding.
8. Save your configuration.

After the saving you can find your Amazon Resource Names (ARN) under *Notifications*.

For more information, see also [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) .

9. In the *SQS* console, create a new queue with the same name you gave the feedback queue in technical configuration.

To do so, choose ► [SQS](#) ► [Create New Queue](#) ► and enter a topic name from the previous steps. Keep all other values as default and save your entries.

10. Subscribe to new queue to the SNS topic you created earlier.
  - a. Go to your AWS-SES account and choose *Security Credentials*.
  - b. Under *Your Security Credentials*, you create an IAM user required to send or used in *Sender Profiles*. We recommend to use IAM users to send emails as you can control with them the permissions and authorizations.
11. To start, choose *Get Started with IAM Users*:
  - a. Choose *Create New Users*.
  - b. Enter the user names.
  - c. Select *Generate an access key for each user*.
  - d. Choose *Create*.

A confirmation message should appear that the users have been created.

12. Now you have set up an account at Amazon for and you receive the following parameters during account set-up:

- *Amazon Access Key ID* (hash string)
- *Secret Access Key* (hash string)
- *Feedback Queue Path*

The *Feedback Queue Path* is the last individual portion of the queue URL at Amazon. For example, the path from `https://sqs.eu-west-1.amazonaws.com/NNNNNNNNNNNN/ABC` is `/NNNNNNNNNNNN/ABC`.

Note that you need these parameters again in the apps *Communication Systems* and *Communication Arrangements*.



13. In the AWS Dashboard under *Groups*, create a new group to assign policies to users. For example, create a group named **Administrator** and assign all the admin policies in the *Attach Policies* step.
14. To assign users to the created group, choose ► *Group Actions* ► *Add Users to Group* ►.

## 4.3.1.1.4.1.2 Importing Certificates

### i Note

The following blogs are not part of the official documentation of SAP Marketing Cloud and some of the information may be outdated.

The following blogs describe how to download and import the certificates to SAP Marketing Cloud:

[Heads-up: Amazon Simple Email Services \(SES\) change Certificates](#) 📄

## 4.3.1.1.4.1.3 Establishing System Connection

After you have downloaded and imported the certificated, you now have to establish the connection between Amazon and the SAP system.

### Context

For already existing customers: If you are changing something in the settings for your system connection after the upgrade from a lower release to release 1902 or higher, you must re-enter data that you have maintained previously in the *Provider Credentials* app:

- For scenario ID `SAP_COM_0016`, you have maintained the credentials (access key and secret key) in the *Communication Systems* app.
- For scenario IDs `SAP_COM_0039` and `SAP_COM_0289`, you have maintained the corresponding paths in the *Communication Arrangements* app under *Outbound Services*.

### Using Several Accounts

If you want to use several accounts, you must do the steps above for each account separately. To get more information about the dependencies in the setup, see [Usage of Multiple Service Provider Instances \[page 146\]](#).

### Procedure

1. To be able to establish the system connection, check that the apps *Communication System* and *Communication Arrangements* are assigned to your user.

2. In the *Communication Systems* app you create the connection with the SAP system and define a communication user with user and password.
  - a. Enter a *System ID* and *System Name*.
  - b. Enter the host names, you got from Amazon. You require different hosts, depending whether you want to create a communication system for sending emails or for bounce and unsubscribe handling.

*Host Name* for emails (Amazon SES): **email.eu-west-1.amazonaws.com**

*Host Name* for bounce and unsubscribe handling (Amazon SQS): **sqs.eu-west-1.amazonaws.com**

Note that you have to use different host names when you are using Amazon in another region than, for example, **EU-WEST**.

- c. Under *User for Outbound Communication* create a user with your provider credentials by choosing *Add*.

You use for emails (Amazon SES) your real credentials, called *Access Key (User)* and *Secret Key (Password)*.

Whereas for bounce and unsubscribe handling (Amazon SQS), you use a dummy user with a generic password.

3. Establish the communication arrangement in the *Communication Arrangement* app.
  - a. To create a new arrangement choose *New*, select the required *Scenario* and enter an *Arrangement Name*.

Technical details for the communication arrangement:

- **Communication Arrangement for Emails**  
*Scenario*: **SAP\_COM\_0016**
  - **Communication Arrangement for Bounce Handling**  
*Scenario*: **SAP\_COM\_0039**
  - **Communication Arrangement for Unsubscribe**  
*Scenario*: **SAP\_COM\_0289**
- b. Select the previously created *Communication System* which fits to the scenario.
  - c. Depending on which communication arrangement you have set up, add the following *Additional Properties*:
    - For emails (**SAP\_COM\_0016**) enter a *Provider ID*, *Sender Profile ID*, and assign a relevant *Marketing Area ID*.
    - For bounce handling (**SAP\_COM\_0039**) and unsubscribe handling (**SAP\_COM\_0289**) you must enter the **EXACT SAME (!)** *Provider ID* that is used for the related email instance (**SAP\_COM\_0016**).  
Note that if you want to use multiple Amazon instances, you must use the *Provider ID* of the related email instance (**SAP\_COM\_0016**) that you want to connect to the bounce and unsubscribe instances.
  - d. Under *Outbound Services* check that the *Service Status* is activated, and *Port 443* is used.

In addition, check the paths:

- *Scenario*: **SAP\_COM\_0016**  
*Path*: no path
- *Scenario*: **SAP\_COM\_0039**

*Path*: The path (also known as *Feedback Queue Path*) is the last individual portion of the queue URL at Amazon. For example, the path from `https://sqs.eu-west-1.amazonaws.com/NNNNNNNNNNNN/ABC` is `/NNNNNNNNNNNN/ABC`.

- *Scenario*: [SAP\\_COM\\_0289](#)

*Path*: The path is the last individual portion of the queue URL at Amazon for unsubscribe. For example, the path from `https://sqs.eu-west-1.amazonaws.com/NNNNNNNNNNNN/ABC` is `/NNNNNNNNNNNN/ABC`.

For more information, see [Enabling Automatic Unsubscribe for Emails by Amazon \[page 143\]](#).

- e. Now save your entries.

During the save the system establishes the required system connection and creates a provider and sender profile.

Note that if you change an already existing communication arrangement in release 1902 and higher, you must

- maintain your user credentials, called *Access Key (User)* and *Secret Key (Password)*, in the *Communication Systems* app.
- maintain the paths in the *Communication Arrangements* app.

- 4. Finally maintain your sender profiles. For more information, see [Sender Profiles \[page 155\]](#).

## Related Information

[Consuming the Integration APIs \[page 395\]](#)

### 4.3.1.1.4.1.4 Useful Blog

#### i Note

The following blog is not part of the official documentation of SAP Marketing Cloud and some of the information may be outdated: .

[How to Set Up Amazon SES as Email Service Provider](#) 

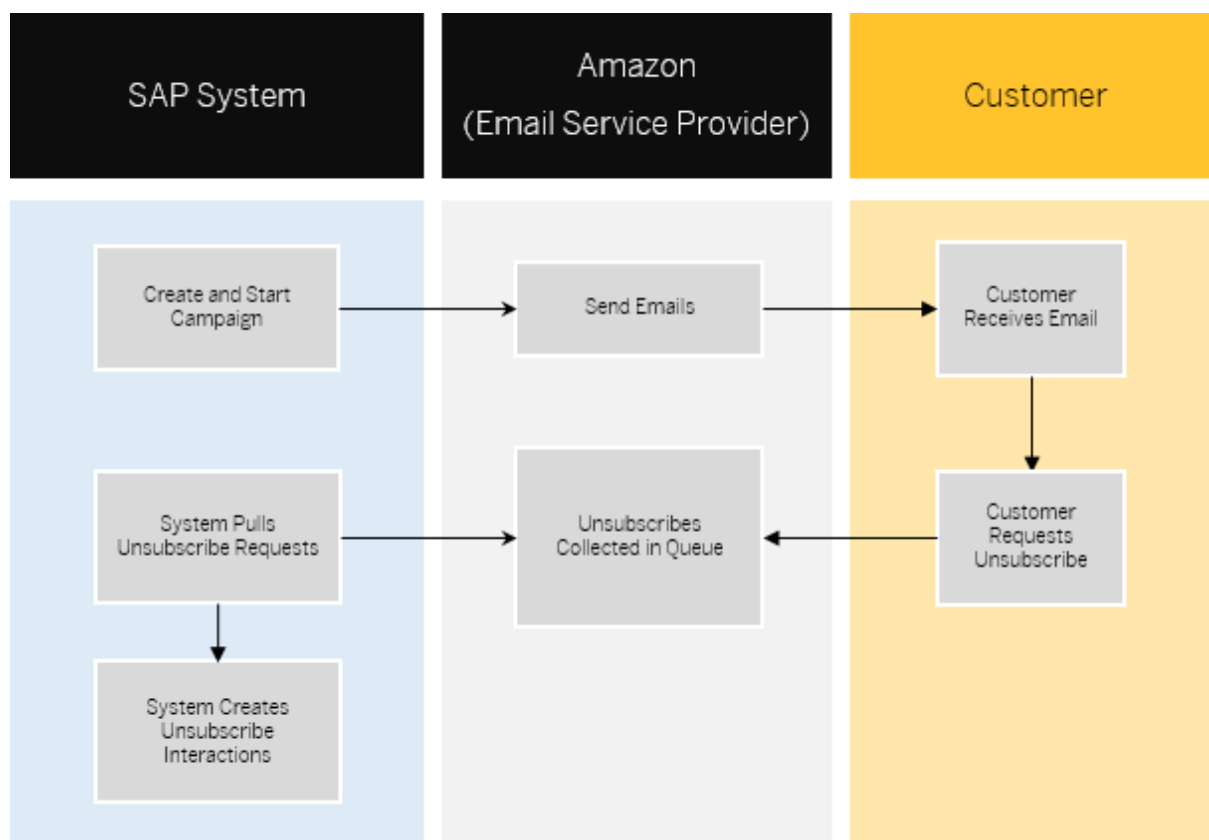
### 4.3.1.1.4.2 Enabling Automatic Unsubscribe for Emails by Amazon

When you want to use the unsubscribe offered by Amazon, you need to do settings at Amazon and at SAP.

After you did all the settings, the header of the recipient's email contains the possibility to unsubscribe.

If the contact does not want to get further emails, she or he sends an unsubscribe request to the unsubscribe email address entered in the sender profile. The unsubscribe requests are collected on Amazon side in a queue. A background job then pulls the unsubscribe requests from Amazon and creates corresponding interactions in

the system. The system evaluates the interactions and updates marketing permissions or list subscriptions for the marketing contact.



In detail the following steps will happen:

1. The marketing expert executes an email campaign.
2. The system sends out the marketing emails.
3. A recipient is getting an email in the inbox.
4. The recipient unsubscribes by clicking on the option in the email header.
5. Email client sends an unsubscribe request to unsubscribe email address of Amazon.
6. Amazon collects unsubscribe requests in a queue.
7. SAP pulls the unsubscribe requests and creates interactions.
8. Based on the interactions the system updates marketing permissions and list subscriptions.

## Prerequisites

You need the communication scenarios [Marketing - Campaign Execution - Amazon E-Mail Integration](#) (SAP\_COM\_0016) and optionally [Marketing - Campaign Execution - Amazon E-Mail Bounce Integration](#) (SAP\_COM\_0039).

## Setup

### At Amazon

1. Create and configure an AWS account. For more information, see [Before You Begin](#) .
2. [Registering a New Domain](#) .
3. Check and verify your domain. For more information, see [Verify your Domain](#) , [Amazon SES Domain Verification TXT Records](#) , and [Publishing an MX Record for Amazon SES Email Receiving](#) .
4. Now you can create your rule set. For more information, see [Set up a Receipt Rule](#) .  
Note that when you create a rule, choose action type **SNS** (instead of **S3** mentioned in the documentation) and give the SNS topic a meaningful name such as **unsubscribe**.
5. Then create a queue with Amazon Simple Queue Service (SQS) by choosing *Subscribe Queue to SNS Topic* from the dropdown menu. Important here to know is that you must connect this queue with the previously created SNS topic. For more information, see [Create a Queue in Amazon Simple Queue Service](#) .

### At SAP

1. Open the *Communication Systems* app and create, if not yet existing, an Amazon SQS system by maintaining the according data.
2. Open the *Communication Arrangements* app and create a communication arrangement for the communication system that was created in the previous step. Enter a / (slash) in the *Path* field. Use for the setup the *Communication Scenario* SAP\_COM\_0289.
3. Open the *Sender Profiles* app and add the *Email Address for Unsubscribing*. For more information, see [Opting-Out and Unsubscribing by Email](#).  
Create either a new domain or register an existing one using Amazon Route 53 for the Amazon Email sender profiles you use in the messages for your campaign execution.

### 4.3.1.1.4.3 How the System Reacts on Amazon's Throttling

When Amazon runs into throttling, Amazon returns an error message. The SAP system reacts on it by reducing the send rate and processing the failed messages.

When Amazon SES runs into throttling, it returns an error message with the following text: `API error: Code "400", Reason: "Bad Request", Message: "Throttling Maximum sending rate exceeded."`

This happens if the maximum send rate is exceeded.

The campaign execution reacts on it by reducing the number of parallel sent request to Amazon SES. Messages that failed due to the Amazon SES error are automatically reprocessed by the campaign execution.

As result of throttling the campaign execution needs more time to process all requests because it sends less in parallel to avoid that the maximum send rate is exceeded again.

#### → Recommendation

If the issue still persists, we recommend to exceed the quota at Amazon.

For more information, see also information given by Amazon: [What Happens When You Reach Your Sending Limits?](#) .

## 4.3.1.1.5 Usage of Multiple Service Provider Instances

In case you want to run campaigns for different customers you can use several instances of the same service provider to gain a better overview about your figures and costs.

But when you plan to create several instances of one service provider, you must keep the following in mind:

- For each service provider account, you create a communication system, a communication arrangement, a provider ID, and a sender profile ID.

### → Recommendation

We recommend that you define all names and IDs upfront, **BEFORE** you start with the creation of the system connection.

### ❁ Example

In the following example you got 2 accounts from Sinch: Both have the same host, but different users and passwords.

You define upfront that the provider IDs shall be **sinch\_01** and **sinch\_02** and the IDs for the generated sender profiles shall be **sinch01** and **sinch02**.

You define upfront that the IDs of the communication systems shall be **DI\_01** and **DI\_02** with the corresponding names **Digital Interconnect 1** and **Digital Interconnect 2**.

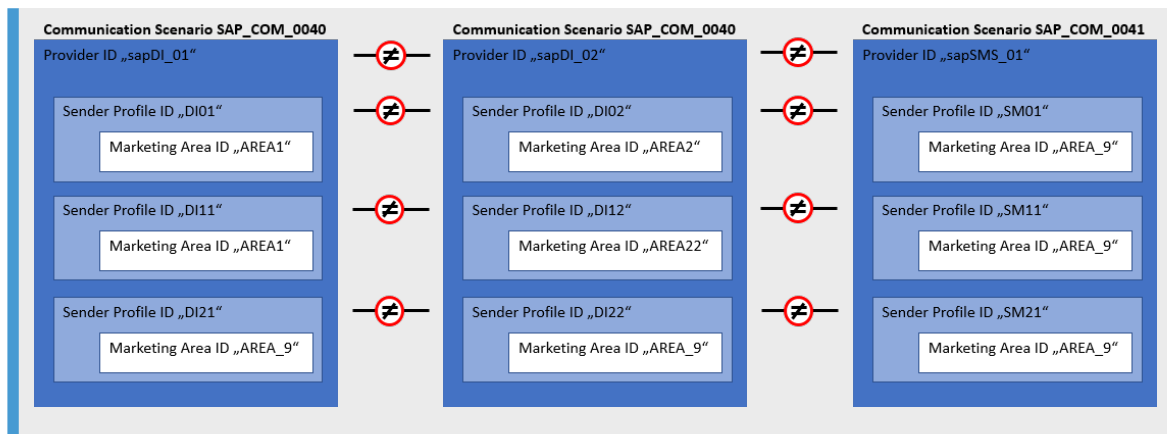
You define upfront that the name of the communication arrangement shall be **Digital Interconnect 0040-1** and **Digital Interconnect 0040-2**.

	Example Account 1	Example Account 2
<b>Account of Service Provider</b>		
<b>Given by Provider</b>		
Host	email-eu1.sapdigitalinterconnect.com	email-eu1.sapdigitalinterconnect.com
User	abc_def1234	xyz_def9876
Password	6T5z)f\$45d\$	98(6/idRt\$m
<b>Defined by Customer</b>		
Provider ID	sinch_01	sinch_02
Sender Profile ID	sinch01	sinch02
<b>Communication System</b>		
<b>Defined by Customer</b>		
System ID	DI_01	DI_02

System Name	Digital Interconnect 1	Digital Interconnect 2
<b>Assigned by Customer</b>		
Host (under Outbound Communication)	email-eu1.sapdigitalinterconnect.com	email-eu1.sapdigitalinterconnect.com
User (under Outbound Communication)	abc_def1234	xyz_def9876
Password	6T5z)f\$45d\$	98(6/idRt\$m
<b>Communication Arrangement for Scenario SAP_COM_0040</b>		
<b>Defined by Customer</b>		
Arrangement Name	Digital Interconnect 0040-1	Digital Interconnect 0040-2
<b>Assigned by Customer</b>		
Communication System	DI_01	DI_02
Provider ID	sinch_01	sinch_02
Sender Profile ID	sinch01	sinch02
Path	/in365-api/abc_def1234/notifications	/in365-api/xyz_def9876/notifications

For every communication arrangement you have to define an unique *Provider ID* and an unique *Sender Profile ID*. Unique means:

If you are defining the sending of emails and the collecting of the bounces, for example, using Amazon as service provider, you must use the same *Provider ID* in all of the related communication scenarios that you want to use: SAP\_COM\_0016 for sending emails, SAP\_COM\_0039 for collecting the bounces, and for enabling the unsubscribe using SAP\_COM\_0289. The same is valid for sending text messages and collecting the text message bounces using Sinch with the scenario IDs SAP\_COM\_0041 and SAP\_COM\_0299.



- In the *Communication Systems* app, you create your system instance with the host given by your service provider.  
For every account of any service provider, you must create a communication system, where you assign the account credentials such as user and password. Depending on the service provider, it can be possible that you use for every account of the same provider the same host.
- In the *Communication Arrangements* app, you create as many communication arrangements for service provider as required. In addition, select also the user you created in the *Communication Systems* app.  
For each additional arrangement entry of the same scenario, you must enter also new provider and sender profile IDs. That means that you must, in this case, overwrite the proposed ones.

## i Note

You cannot use IDs that are reserved for other service providers.

The following provider and sender profile IDs are reserved:

Provider ID	Sender Profile ID
sapGeneric	GNML
genSmsAdap	GENS
mobPush	MPN
sapMS1025	MSTS

Note that the *Provider ID* is case-sensitive and that the IDs are also reserving entries that are starting with these IDs as prefix.

For example, *Provider ID* **aliMail** reserves also entries starting with **aliMail\*** (but not **ALIMail\***), whereas *Sender Profile ID* **AM** reserves also entries starting with **AM\***.

The steps for the setup itself are the same as described in the chapters [Setting Up Sinch \[page 112\]](#) and [Setting Up Amazon \[page 138\]](#).

Saving the communication arrangement, the system creates the corresponding providers and sender profiles, and establishes the required system connections.

In the *Sender Profiles* app then you must complete the generated sender profiles, with, for example, a sender address. There you can also change the assigned marketing area and also copy the profiles. But be aware that the copied profile uses the **SAME** provider ID as the source profile.



Service Provider	Scenario ID	Specifics
Sinch: Sinch E-Mail 365	SAP_COM_0040	In the <i>Communication Systems</i> app, you enter the user credentials you got from Sinch: <i>User</i> and <i>Password</i>
Sinch: SAP Intelligent Notification 365, SMS API	SAP_COM_0041	In the <i>Communication Systems</i> app, you enter the user credentials you got from Sinch: <i>User</i> and <i>Password</i>
Sinch: SAP Intelligent Notification 365, SMS API for Bounces and Unsubscribe	SAP_COM_0299	<p>In the <i>Communication Systems</i> app, you enter the user credentials you got from Sinch Live Link 365: <i>User</i> and <i>Password</i></p> <p>In the <i>Communication Arrangements</i> app, you must use the same <i>Provider ID</i> as in the setup for the related communication arrangement for scenario SAP_COM_0041.</p>
Amazon SES for Email	SAP_COM_0016	In the <i>Communication Systems</i> app, you enter the user credentials you got from Amazon, called <i>Access Key (User)</i> and <i>Secret Key (Password)</i> .
Amazon SQS for Bounces	SAP_COM_0039	<p>In the <i>Communication Systems</i> app, you create only a dummy user with a dummy password.</p> <p>In the <i>Communication Arrangements</i> app, you must use the same <i>Provider ID</i> as in the setup for the related communication arrangement for scenario SAP_COM_0016.</p>
Amazon SQS for Unsubscribe	SAP_COM_0289	<p>In the <i>Communication Systems</i> app, you do not need to create a new system. You can re-use the system for SAP_COM_0039.</p> <p>In the <i>Communication Arrangements</i> app, you must use the same <i>Provider ID</i> as in the setup for the related communication arrangement for scenario ID SAP_COM_0016 and SAP_COM_0039. But note that the <i>Path</i> is different from the setup for scenario SAP_COM_0039.</p>

### 4.3.1.1.6 Campaign Execution Inclusion List

You use the *Campaign Execution Inclusion List* app to maintain allowed email addresses and telephone numbers.

You maintain the allowed email addresses and telephone numbers that you want to use, when you create marketing campaigns for test purposes. With the entries, you avoid sending test emails and test text messages to your customers.

#### i Note

- You use this app only in your test system.
- As soon, as you have maintained email addresses or telephone numbers in the app, only those recipients from this list can be contacted.

When you create an inclusion list entry, the system checks whether there's already an existing exclusion list entry with the same email address or email domain or mobile phone number. When an entry exists in the exclusion list, the system rejects this entry in the inclusion list.

This behaviour is also valid for overlapping email domains. For example, if the exclusion list has an email entry `@example.org`, you can't add the email address `joe.public@example.org` to the inclusion list.

### 4.3.1.1.7 Unsubscribe for Emails and Text Messages

In the following, you learn about the possibilities to unsubscribe from emails and text messages.

#### Automatic Unsubscribe for Emails (List Unsubscribe)

With the automatic unsubscribe, you don't need to

- maintain the unsubscribe email
- collect the unsubscribe information provided from email client.
- handle the permissions based on an email unsubscribe

All these steps are automatically done by the system.

To enable the automatic unsubscribe, you have created the required communication arrangements for Sinch (SAP\_COM\_0040) and/or Amazon (SAP\_COM\_0289) for unsubscribe. To get also the bounces, we recommend to also set up the communication scenario (SAP\_COM\_0039).

For the setup, you must enter an *Email Address for Unsubscribing* in the *Sender Profiles* app and the system adds the unsubscribe information to the email header. Most email clients then show an unsubscribe button at the very top of the email.

#### i Note

For Sinch, the email address is preset and can be changed.

For Amazon, the email address must be added manually as the address is account-specific.

After the recipient has unsubscribed, the email client sends an email with the information back to the service provider. The SAP system collects the data from the service provider and maintains the marketing permissions (Opt-out) automatically.

In a newsletter campaign, the unsubscribe also contains the communication category. With the communication category and contact data, the unsubscribe from a newsletter can be realized and the system knows that this unsubscribe is one for a newsletter.

For more information, see:

- [Opting-Out and Unsubscribing by Email](#)
- [Enabling Automatic Unsubscribe for Emails by Amazon \[page 143\]](#)

## Automatic Unsubscribe for Text Messages (STOP Trigger)

To enable the automatic unsubscribe for text messages, you have created the required communication arrangements for Sinch (SAP\_COM\_0041 and SAP\_COM\_0299).

After the recipient has unsubscribed by sending back the word **STOP**, the SAP system collects the data from the service provider and maintains the marketing permissions (opt-out) automatically.

For more information, see [Bounces and Unsubscribe for Text Messages \[page 115\]](#).

## Manual Unsubscribe for Emails

For the manual unsubscribe, you don't need any configuration, beside of adding an *Email Address for Unsubscribing* and/or a *Follow-Up Page for Unsubscribing* to the used sender profile in the *Sender Profiles* app.

Using the *Email Address for Unsubscribing* or the *Follow-Up Page for Unsubscribing* you can collect the required data and update your subscriptions and permissions (opt-out) manually in the *Contacts* app.

## Easy Opting-Out and Easy Unsubscribe for Email

With this option, you add a link to the email body that triggers the unsubscribe.

The option is independent from any service provider.

For more information, see [Opting-Out and Unsubscribing by Email](#).

## Unsubscribe and Marketing Areas

The following explains in more detail how the system behaves, whether marketing areas are activated for campaign execution or not.

- **Sinch: Unsubscribe from Text Messages Sending Back STOP**

From text messages sent by Sinch can be unsubscribed by sending back the word **STOP**, also when marketing areas are activated for campaign execution. The system uses the mobile number to determine the contacts.

If the text message with the word **STOP** contains also a valid campaign ID, the marketing area of this campaign is used for the opt-out of the corresponding contact mobile number. For this scenario it is irrelevant whether the marketing areas are activated for campaign execution or not.

Only for the case no campaign ID has been sent back with the STOP trigger:

- If marketing area is not activated in the configuration, the system looks for all marketing areas that are assigned to sender profiles for sending text messages and creates an opt-out for **one** marketing area of the contact(s) related to the mobile number.
- If marketing area is activated, the system looks for all marketing areas that are assigned to sender profiles for sending text messages and creates opt-outs **for each found** marketing area of the contact(s) related to the mobile number.

- **Email Unsubscribe by Amazon**

Amazon provides an identification with which the outgoing email can be identified, and campaign, contact, and marketing area are determined.

The system creates one opt-out (independent whether the separation is activated or not) with the data determined from the outgoing email.

- **Email Unsubscribe by Sinch**

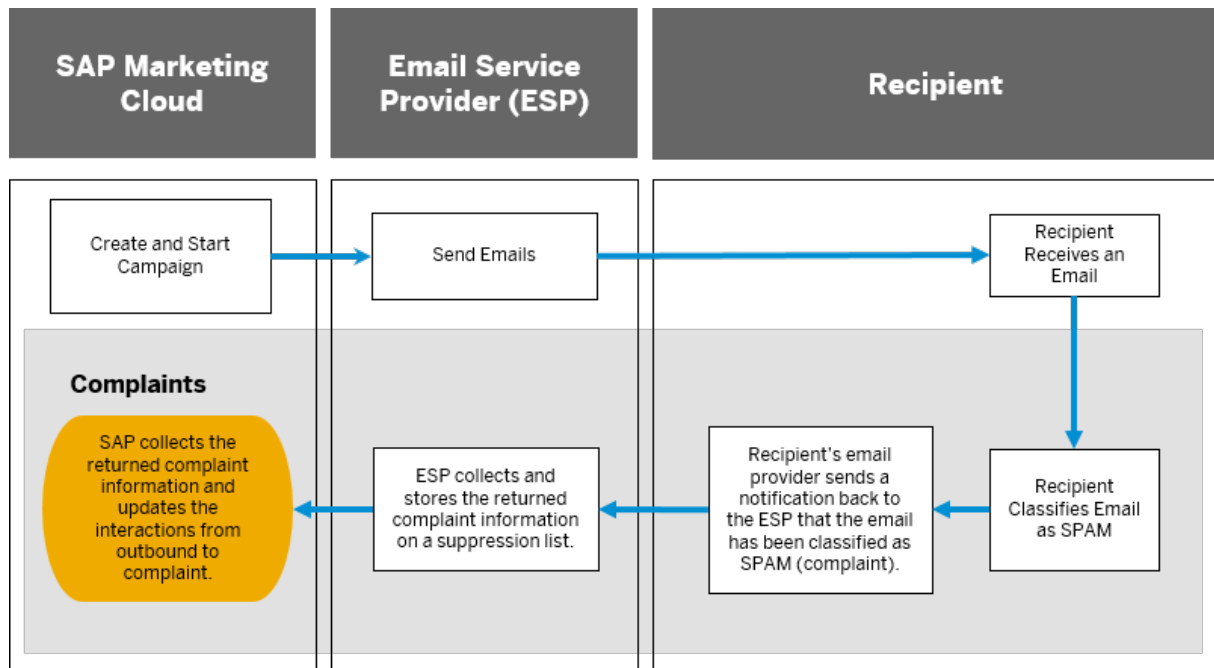
Sinch provides an identification with which the outgoing email can be identified, and campaign, contact, and marketing area are determined.

The system creates one opt-out (independent whether the separation is activated or not) with the data determined from the outgoing email.

For more information, see [Marketing Area for Campaigns](#).

### 4.3.1.1.8 Complaints for Emails

Complaints for email means that an email recipient classifies emails from dedicated senders as spam. For classifying emails as spam, the email recipient either drops the email to the spam folder of the email provider or declares the email as spam. This technology is also known as email feedback loops.



**Process**

1. Email recipient classifies a received campaign email as spam.
2. The classification is sent to the email service provider (ESP).
3. The ESP sends complaint information to the SAP system, and the complaint information is stored in the system and visible for the marketer.
4. Depending on the campaign scenario, a complaint will have the following results:
  - If the email has been sent by a subscription-based campaign, the SAP system does an unsubscribe for the email address of the recipient/contact and for the corresponding communication category. The global opt-in for marketing permission is not changed.

- If the email has been sent by a non-subscription-based campaign, the SAP system does a permission opt-out for the email address of the recipient/contact.

### **i Note**

The described complaint handling is only possible if the email provider of the recipient sends a notification back to SAP, which means the provider supports email feedback loops. Not all email providers support this technology.

The email service provider you used to send the emails out of the SAP system (Sinch, Amazon SES, or the generic email interface) stores the returned complaint information on a suppression list. If you use another campaign to send again an email to the recipient that sent a complaint, the email is not delivered to this recipient even if the marketing subscription or permission is still Opted-In in the SAP system. The email is not delivered by the email service provider because the email address is part of the suppression list of the email service provider.

Note that your used ESP in your generic email adapter can have such a suppression list and collect the emails with complaints. But that's not certain and there can be ESPs without that complaint process.

## **Performance Tab**

To see the number of email complaints, open the corresponding campaign in the *Campaigns* app. On the *Performance* tab, you see the actuals in the *Email Complaints* tile.

## **Related Information**

[Enabling Complaints \[page 153\]](#)

[Removing a Contact from Suppression List \[page 154\]](#)

### **4.3.1.1.8.1 Enabling Complaints**

The steps guide you to enable complaint handling in your SAP system.

## **Prerequisites**

This function is available if you are using the email services from Sinch or Amazon SES. In case you are using the generic email service provider interface, the availability of the functionality depends on the capabilities of the email service provider behind the interface.

## Procedure

Create an incident for the SAP component **CEC-MKT-CPG-EXE** and request the activation for one of the following options:

- Update of marketing permission during processing of complaints for email
- Update of marketing subscription during processing of complaints for email
- Update of marketing permission and subscription during processing of complaints for email

## Results


After the enabling the system updates the marketing permissions or subscriptions during the processing of complaints for email.

### 4.3.1.1.8.2 Removing a Contact from Suppression List

In case, your customer wants to be contacted again, you must remove the email address of the contact again from the suppression list.

## Procedure

To remove a contact again from the service provider's suppression list, you do the following steps.

1. For Sinch, send an email to [essupport.digitalinterconnect@sinch.com](mailto:essupport.digitalinterconnect@sinch.com) and ask Sinch to remove the contact again from the suppression list.
2. For Amazon, log on at your AWS Management Console and remove the customer manually. For more information, see [Removing an Email Address from the Amazon SES Suppression List](#) .

### 4.3.1.1.9 Troubleshooting for Campaigns

In case you have issues with the execution of your campaigns, we recommend to read also the troubleshooting in the **Administration Guide**.

## Related Information

[Troubleshooting Campaigns](#)

### 4.3.1.1.10 Sender Profiles

A sender profile allows you to carry out campaigns for different channels in different markets. You can maintain sender profiles for channels, such as email, text message, and mobile push notifications.

#### Prerequisites

- You have set up the service provider for emails and text messages and you have maintained the communication arrangements before creating sender profiles.  
For more information, see [Setting Up Service Provider for Emails and Text Messages \[page 110\]](#).
- You have registered the *Sender Address* and the *Reply-To Address* at Sinch and/or Amazon.

#### i Note

Note that the registered email address is case-sensitive for Amazon and Sinch.

#### Recommendation: Test Sender Profiles

To test the maintained sender profiles, we recommend to use *Send Test Email* or *Send Test Text Message* to ensure that the settings are working. Otherwise the issues can appear during campaign execution.

#### Related Information

[Mobile Campaigns](#)

[Opting-Out and Unsubscribing by Email](#)

## 4.3.2 Setting Up External Campaign Execution

SAP Marketing Cloud allows you to execute campaigns in an external system, and to request the success data for further processing in SAP Marketing Cloud.

#### Implement Interfaces

Optionally implement a set of interfaces, either directly in the external system, or using a suitable middleware, such as *SAP HANA Cloud Integration*, to map the interfaces in SAP Marketing Cloud to the interfaces of the external system. For the implementation details, see [Implementing Interfaces for External Campaign Execution \[page 157\]](#).

## Set up a Communication Arrangement

Finally, set up a communication arrangement for the external campaign execution, and the success data requests. For the details, see [Communication Arrangement for External Campaign Execution \[page 192\]](#).

## Process

Once you have set up your system for external campaign execution, executing the campaigns occurs as follows:

1. **Plan Campaign**

In SAP Marketing Cloud, plan your campaigns from program down to detailed spend (optional).

2. **Release Campaign**

In SAP Marketing Cloud, create and release your campaign. A corresponding campaign is created automatically in the external system. You can optionally assign a target group to the campaign and schedule the transfer of the target group to the external system. Once the campaign is activated, your target group will be updated periodically according to the schedule you set.

Alternatively, you can link an existing campaign manually from the external system.

3. **Execute Campaign**

In the external system, execute the campaign.

If you choose to use a marketing agency, they can execute the campaign in the external system without needing users in SAP Marketing Cloud. You need technical users in the external system to set up the connectivity.

Targeting can be done in the external system, or you can **transfer a target group** as described in the *Release Campaign* step. Once the target group is in the external system, the marketing agency, if you choose to use one, can use it for targeting.

4. **Track Success**

In SAP Marketing Cloud, you can track the success of your campaign. Success data is automatically retrieved from the external system. You also can upload success data manually as a CSV file.

5. **Delete Contacts**

Once per day a batch report is executed to check if contacts have been deleted, the permissions for the contacts were changed, or the contacts became expired on external system. If one of those conditions is met, then the action for the external campaign execution will be executed. This process will be executed until the campaign is stopped, except if the action is placed directly after the first node in the campaign workflow. In this case, the process will be executed until the end date of the campaign is reached.

## Example

### Baidu Paid Search Campaign

You can use the method described in this chapter to implement Baidu Paid Search Campaigns. For a more detailed explanation on how to accomplish this with Baidu Paid Search Campaigns specifically, see the blog [Handling Baidu Paid Search Campaigns using External Campaigns in SAP Marketing Cloud](#).



## 4.3.2.1 Implementing Interfaces for External Campaign Execution

You implement a set of interfaces to enable the creation of campaigns in the external system, and to obtain the success data for the executed campaigns.

### Overview

Executing campaigns in an external system is set up as follows:

- The user creates a campaign for external execution in SAP Marketing Cloud. As a result, your system calls the external system for a list of parameters (including possible values) required for the creation of the campaign in the external system.
- The application displays the received parameters allowing the user to specify the parameter values.
- The user releases the campaign. As a result, the corresponding campaign is created in the external system and any assigned target group is transferred to the external system.
- Once the campaign is created in the external system, your system requests success data for the campaign. The request is repeated every 4 hours, until the external system indicates that no more success data is expected.

The services available with SAP Marketing Cloud, are based on OData Version 2.0. The OData messages are sent in the JSON format. In the communication, the marketing edition system acts as a client. The server side implementation of the services is done by the external system, or a middleware, such as [SAP Cloud Integration](#).

For the server side implementation, the required methods and entity sets are specified in the following sections, and it is explained how to use OData features, such as filtering, sorting, or paging. For the responses, the metadata ("\_\_metadata") is optional.

The following types are included in the list of required entity types:

Entity Type	Description
Campaign	The campaign, as required to integrate with an external system executing the campaign
CampaignParameter	A generic parameter of a campaign that is specific to an external system, and not known to your system in detail
CampaignParameterCode-ListValue	A code list value of a campaign parameter for parameters that have the type CODE_LIST
CampaignParameterValue	A value of a campaign parameter, for example, the selected code list value for a parameter of the type CODE_LIST
MarketingSuccess	All marketing success KPIs for a given set of characteristics including campaign, date, gender, and age
ExtTargetGroup	The target group, as needed to integrate with an external system executing a campaign

Entity Type	Description
ExtTargetGroupParameterValue	A value of a target group parameter, for example, the selected code list value for a parameter of the type CODE_LIST
ExtTargetGroupSupporterIdOrigin	An ID origin requested by the external system executing a campaign
ExtTargetGroupMemberFacet	A member of the target group
ExtTGContactTransferHeaderSet	The personalized attributes of a target group member

The identification of an externally executed campaign by SAP Marketing Cloud can require additional parameters, for example, when the externally assigned campaign ID is not unique. In this case, the external system can indicate that certain parameters, specifically `ADVERTISER` and `EXT_CAMPAIGN_MANAGING_PARTY`, are included in the campaign key. Then, your system can use the additional parameters when filtering for a set of campaigns.

Example: Advertiser is an additional key, the combination of Advertiser and Campaign ID makes a unique key of an external campaign. During the `GetEntitySet` request of an external campaign, the client sends the additional key in the request filters like below:

```
((Filter/KeyPart2Id eq 'ADVERTISER') and (Filter/KeyPart2Value eq 'SAP_GLOBAL_MARKETING'))
```

Additional Filter Parameters

ServerCampaignIDContextCT (Complex Type)	Used for filtering in GetEntitysetMethods
KeyPart2Id (Edm.String, length-50)	ID of the first parameter marked to be part of the campaign key
KeyPart2Value (Edm.String, length-50)	Value of the first additional parameter
KeyPart3Id (Edm.String, length-50)	ID of the second parameter marked to be part of the campaign key
KeyPart3Value (Edm.String, length-50)	Value of the second additional parameter

For more information about OData Version 2.0, see <http://www.odata.org/documentation/odata-version-2-0/>



## 4.3.2.1.1 Requesting Parameters for Campaign Creation

The request for campaign parameters is triggered when the user creates an externally executed campaign.

For the parameter retrieval, SAP Marketing Cloud calls the `GetEntitySet` method of the `CampaignParameterSet` entity set with an expand to the `CampaignParameterCodeListValueSet` without any filter. As a result, the following list of the parameters is expected:

Campaign Parameters

CampaignParameter (Entity Type)	Usage in GetEntitySet Method
Properties and Data Type	
Id (Edm.String, length-50)	<p>Stable (language independent) ID of the parameter</p> <p>For Type ADVERTISER the ID must be set to ADVERTISER</p> <p>For Type EXT_CAMPAIGN_MANAGING_PARTY the ID must be set to EXT_CAMPAIGN_MANAGING_PARTY</p> <p>ADVERTISER and EXT_CAMPAIGN_MANAGING PARTY are key parameters</p> <p>Mandatory in response</p>
Name (Edm.String, length-255)	<p>Potentially language dependent parameter name displayed in SAP Marketing Cloud</p> <p>Optional in response (ID is displayed when name is missing)</p>
Type (Edm.String, length-30)	<p>Supported types: CODE_LIST, TEXT, AMOUNT, NUMBER, ADVERTISER, EXT_CAMPAIGN_MANAGING_PARTY</p> <p>Mandatory in response</p>
Navigation Properties	
CampaignParamCodeListValues	List of possible values for parameters of type CODE_LIST

For parameters of the types CODE\_LIST, ADVERTISER and EXT\_CAMPAIGN\_MANAGING\_PARTY , the following additional list of possible values is required.

### i Note

For amount parameters, a list of supported currencies can be provided. If no list of currencies is given, the user can select from all currencies available in SAP Marketing Cloud.

Additional Code List Parameters

CampaignParamterCodeListValue (Entity Type)	Usage in GetEntitySet Method
Properties and Data Type	

## CampaignParamterCodeListValue (Entity Type) Usage in GetEntitySet Method

Code (Edm.String, length-50)	Stable (language independent) code Mandatory in response
Description (Edm.String, length-255)	Potentially language dependent description of the code displayed in SAP Marketing Cloud Indicates (true or false) whether the type parameter is included Optional in response – Code is displayed when description is missing.
CampaignParameterId (Edm.String, length-50)	Reference to the campaign parameter Mandatory in response

An example as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/CampaignParameterSet? $expand=CampaignParamCodeListValues&$format=json .`
- HTTP Method: GET

**Sample Response, as sent by the external system:**

### Sample Code

```
{
  "d": {
    "results": [{
      "Id": "ADVERTISER",
      "Name": "Advertiser",
      "Type": "ADVERTISER",
      "CampaignParamCodeListValues": {
        "results": [{
          "CampaignParameterId": "ADVERTISER",
          "Code": "SAP_GLOBAL_MARKETING ",
          "Description": "SAP Global Marketing"
        },
        {
          "CampaignParameterId": "ADVERTISER",
          "Code": "SAP_GERMANY",
          "Description": "SAP Germany"
        }
      ]
    }
  ],
  {
    "Id": "METRIC",
    "Name": "Metric",
    "Type": "CODE_LIST",
    "CampaignParamCodeListValues": {
      "results": [{
        "CampaignParameterId": "METRIC",
        "Code": "CLICKS",
        "Description": "Number of Clicks"
      },
      {
        "CampaignParameterId": "METRIC",
        "Code": "IMPRESSIONS",
        "Description": "Number of Impressions"
      }
    ]
  }
},
```

```

{
  "Id": "CAMPAIGN_DESC",
  "Name": "Campaign description",
  "Type": "TEXT",
},
{
  "Id": "DAILY_BUDGET",
  "Name": "Daily Budget",
  "Type": "AMOUNT",
  "CampaignParamCodeListValues": {
    "results": [{
      "CampaignParameterId": "DAILY_BUDGET",
      "Code": "EUR",
      "Description": "Euro"
    },
    {
      "CampaignParameterId": "DAILY_BUDGET",
      "Code": "USD",
      "Description": "American Dollars"
    }
  ]
},
{
  "Id": "DAILY_IMPRESSIONS",
  "Name": "Target Daily Impressions",
  "Type": "NUMBER",
}
}

```

### 4.3.2.1.2 Handling Campaigns in the External System

The user releases the externally executed campaign. As a result, SAP Marketing Cloud calls the `Create` method of the `CampaignSet` entity set including a deep create of the `CampaignParameterValueSet`.

For the campaign ID value help, available in the *Campaigns* app, the `GET` method is used to retrieve the relevant information. `CampaignName` must be enabled for filtering.

Create Campaign Method

Campaign (Entity Type)	Usage in Create Method
Properties and Data Type	
ClientCampaignId (Edm.String, length-10)	ID of the campaign as provided by SAP Marketing Cloud Ignored in response
CampaignName (Edm.String, length-40)	Name of the campaign as provided by SAP Marketing Cloud Needs to be enabled for filtering Ignored in response
StartDate (Edm.DateTime)	Start date of the campaign as provided by SAP Marketing Cloud Ignored in response

Campaign (Entity Type)	Usage in Create Method
EndDate (Edm.DateTime)	End date of the campaign as provided by SAP Marketing Cloud Ignored in response EndDate >= StartDate; no further constraints
ServerCampaignId (Edm.String, length-32)	ID of the externally created campaign as provided by the external system Mandatory in response
ServerCampaignUrl (Edm.String)	Link to the external campaign, optionally provided by the external system Enables navigation from SAP Marketing Cloud to the campaign Optional in response
MainKPI (Edm.String, length-50)	Main KPI displayed in SAP Marketing Cloud, such as <i>IMPRESSIONS</i> , or <i>CLICKS</i> Optional in response Can be provided with success data
SuccessDataEndDate (Edm.DateTime)	Last date by which SAP Marketing Cloud requests administrative, and success data for the campaign Can be changed until the date is reached, and can be provided with success data retrieval Optional in response If not provided in response, the campaign end date is used
SuccessDataTimeZone (Edm.String, length-6)	Not used, ignored in response Not provided by SAP Marketing Cloud
Navigation Properties	
CampaignParameterValues	List of parameters with corresponding values

## Campaign Creation OData Request

An example of a campaign creation request , as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/CampaignSet`
- HTTP Method: POST

### Sample Request Payload:

```
{
  "ClientCampaignId" : "12345",
  "CampaignName" : "My Test Campaign",
  "StartDate" : "\/Date(1452470400000)\/",
  "EndDate" : "\/Date(1454112000000)\/",
  "CampaignParameterValues" :
```

```
[
  {
    "ClientCampaignId" : "12345",
    "Id" : "ADVERTISER",
    "Value" : "SAP_GLOBAL_MARKETING"
  },
  {
    "ClientCampaignId" : "12345",
    "Id" : "METRIC",
    "Value" : "IMPRESSIONS"
  },
  {
    "ClientCampaignId" : "12345",
    "Id" : "DAILY_BUDGET",
    "AmountValue" : "100.00"
    "CurrencyValue" : "EUR"
  },
  {
    "ClientCampaignId" : "12345",
    "Id" : "DAILY_IMPRESSIONS",
    "NumberValue" : 100000
  }
]
```

### Sample Response Payload:

```
{
  "d" : {
    "ServerCampaignId" : "54321",
    "ServerCampaignUrl" : "www.example.com/54321",
    "MainKPI" : "CLICKS",
    "SuccessDataEndDate" : "\/Date(1455494400000)\/"
  }
}
```

For each parameter required by the external system, a value is sent with the creation of the external campaign:

#### Create Campaign Method

CampaignParameterValue (Entity Type)	Usage in Create Method
Properties and Data Type	
ClientCampaignId (Edm.String, length-10)	Reference to the campaign as provided by SAP Marketing Cloud Ignored in response
Id (Edm.String, length-50)	Stable ID of the parameter as provided by SAP Marketing Cloud Ignored in response
Value (Edm.String, length-256)	Value the user specifies in SAP Marketing Cloud; provided by SAP Marketing Cloud (all parameters assumed to be mandatory) Ignored in response For code lists the code is provided, but not the description.

CampaignParameterValue (Entity Type)	Usage in Create Method
NumberValue (Edm.Int32, length 10)	<p>Value entered by the user in SAP Marketing Cloud for the parameter of the type CODE_LIST or TEXT</p> <p>For code lists, the code is provided but not the description.</p> <p>Provided by SAP Marketing Cloud</p> <p>Ignored in response</p>
AmountValue (Edm.Decimal, Precision- 31, scale -2)	<p>Value entered by the user in SAP Marketing Cloud for the parameter of the type AMOUNT</p> <p>Provided by SAP Marketing Cloud</p> <p>Ignored in response</p>
CurrencyValue (Edm.String, length 5)	<p>Currency selected by the user in SAP Marketing Cloud SAP Marketing Cloud</p> <p>Ignored in response</p>

### i Note

Some servers expect a CSRF token for modifying requests, such as *POST*. Send a non-modifying request, such as *GET* from the client to retrieve the token. If the server does not support CSRF token mechanism, the client sends *POST* requests without a token header.

### Campaign ID Value Help OData Request

An example of a campaign ID value help request follows, as sent by SAP Marketing Cloud. This example corresponds with the outbound service in the communication arrangement that retrieves information for the value help for campaign assignment ([CampaignValueHelpSet](#)):

- Request URL: `https://<HostName>/../<YourService>/CampaignSet?$filter=(CampaignNameeq'Summer Campaign'and ((Filter/KeyPart2Id eq 'ADVERTISER') and (Filter/KeyPart2Valueeq'GLOBAL_MARKETING'))&$inlinecount=allpages&$top=100&$skip=0&$format=json`
- HTTP Method: GET

### Sample Response Payload:

#### Sample Code

```
{
  "d": {
    "results": [{
      "ServerCampaignId": "54321",
      "CampaignName": "Summer Campaign 2018",
      "StartDate": "\/Date(1455494400000)\/",
      "EndDate": "\/Date(1455494700000)\/",
      "ExternalCampaignStatus": "RELEASED"
    },
    {
      "ServerCampaignId": "54322",
      "CampaignName": "Jumpstart Summer Campaign",
      "StartDate": "\/Date(1455495400000)\/",

```



```

    "EndDate": "\/Date(1455495700000)\/",
    "ExternalCampaignStatus": "FINISHED"
  }
}
}

```

### 4.3.2.1.3 Creating an External Target Group

To create a target group in the external system, SAP Marketing Cloud calls the create method of the ExtTargetGroup entity.

You can create a target group in the external system by transferring member IDs.

Target Group Entity Types

ExtTargetGroup (Entity Type)	Usage in Create Method
Properties	
TargetGroupId (Edm.String, length-10)	ID of the target group in SAP Marketing Cloud Always provided by SAP Marketing Cloud Ignored in response
ExtTargetGroupId (Edm.String, length-32)	ID of the created external target group Never provided by SAP Marketing Cloud Mandatory in response
ExtTargetGroupUrl (Edm.String)	Link to the external target group Never provided by SAP Marketing Cloud Optional in the response – if not provided there is no navigation from the UI in SAP Marketing Cloud to the external target group
ExtTargetGroupName (Edm.String, length-255)	Name of the external target group as entered on the UI in SAP Marketing Cloud Always provided by SAP Marketing Cloud Ignored in response
TransferMaxBatchSize (Edm.Int32, length-10)	The maximum number of target group member facets bundled in one batch Never provided by SAP Marketing Cloud Optional in response – with this parameter the external system can limit the batch size to a suitable value down to 1 in case that no batches are supported at all

<b>ExtTargetGroup (Entity Type)</b>	<b>Usage in Create Method</b>
TransferMethod (Edm.String, length-10)	<p>The way target group member facets are transferred: Must be FULL or DELTA</p> <p>In case of FULL, each update of the target group (re) creates all member facets</p> <p>In case of DELTA, facets of joiners to the target group are created, facets of leavers are deleted</p> <p>Never provided by SAP Marketing Cloud</p> <p>Optional in response – DELTA is assumed as default</p>
MktPermissionCommMedium (Edm.String, length-20)	<p>The communication medium to check marketing permissions for</p> <p>Only contacts with valid marketing permissions for the communication medium are transferred from SAP Marketing Cloud</p> <p>Never provided by SAP Marketing Cloud</p> <p>Optional in response – if no communication medium is provided no marketing permissions are checked.</p>
Navigation Properties	
ExtTargetGroupParameterValues	<p>List of campaign parameters with corresponding values</p> <p>Always provided by SAP Marketing Cloud if requested</p> <p>Ignored in response</p>
Target Group Parameter Entity Types	
<b>ExtTargetGroupParameterValue (Entity Type)</b>	<b>Usage in Create Method</b>
Properties	
TargetGroupId (Edm.String, length-10)	<p>ID of the SAP Marketing Cloud target group</p> <p>Always provided by SAP Marketing Cloud</p> <p>Ignored in response</p>
Id (Edm.String, length-50)	<p>Stable ID of the parameter</p> <p>Always provided by SAP Marketing Cloud</p> <p>If the type is ADVERTISER, the ID must be set to ADVERTISER</p> <p>Ignored in response</p>

ExtTargetGroupParameterValue (Entity Type)	Usage in Create Method
Value (Edm.String, length-255)	<p>Value entered by the user in SAP Marketing Cloud for the parameter; for code lists the code list ID is provided (and not the name)</p> <p>Always provided by SAP Marketing Cloud as all parameters are assumed to be mandatory</p> <p>Ignored in response</p>
ExtTargetGroupDescription (Edm.String)	<p>Description of the external target group as entered in SAP Marketing Cloud</p> <p>Optional field provided by SAP Marketing Cloud</p> <p>Ignored in response</p>

For the creation of an external target group, arbitrary parameters are not supported. The only supported parameter is the [Advertiser](#).

If in the step *Requesting Parameters for Campaign Creation* a parameter of the type ADVERTISER is requested, this advertiser becomes a mandatory field on the campaign UI. The entered value is then provided with the OData request creating the external target group.

Any other parameter from the step *Requesting Parameters for Campaign Creation* will be ignored for the external target group creation and is only available for external campaign creation.

## OData Request

An example of an external target group creation request, as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/ExtTargetGroupSet`
- HTTP Method: POST

### Sample Request Payload:

Sample Code

```
{
  "TargetGroupId": "123",
  "ExtTargetGroupName": "Customers Germany",
  "ExtTargetGroupParameterValues": [{
    "TargetGroupId": "123",
    "Id": "ADVERTISER",
    "Value": "GLOBAL_MARKETING"
  }]
}
```

## Sample Response Payload:

### Sample Code

```
{
  "d": {
    "ExtTargetGroupId": "Ext123",
    "ExtTargetGroupUrl": "https://www.example.com/TG/Ext123",
    "TransferMaxBatchSize": 500,
    "TransferMethod": "DELTA",
    "MktPermissionCommMedium": "DISPLAY_ADS"
  }
}
```

### 4.3.2.1.3.1 Requesting ID Origins

After creating the external target group the needed ID origins are requested by SAP Marketing Cloud with a `GetEntitySet` call for the `ExtTargetGroupSupportedIdOrigin`.

Target Group ID Origin Entity Types

ExtTargetGroupSupportedIdOrigin (Entity Type)	Usage in GetEntitySet Method
Properties	
IdOrigin (Edm.String, length-10)	ID origin to transfer IDs for (e.g. request transfer of email addresses and phone numbers)  Never provided by SAP Marketing Cloud  Mandatory in response
HashingMethod (Edm.String, length-10)	Hashing algorithm to hash the IDs to be transferred.  Supported methods: SHA256  Never provided by SAP Marketing Cloud  Optional in response - if no hashing method is provided the IDs of the target group members are transferred without hashing

## odata Request

An example of a target group ID origins request as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/ExtTargetGroupSupportedIdOriginSet?$format=json`
- HTTP Method: GET

## Sample Response Payload:

### Sample Code

```
{
  "d": {
    "results": [{
      "IdOrigin": "EMAIL",
      "HashingMethod": "SHA256"
    },
    {
      "IdOrigin": "PHONE",
      "HashingMethod": ""
    }
  ]
}
```

## 4.3.2.1.3.2 Transferring Contact IDs

The Target Group Members are transferred either as a single member or in batch. This depends on the parameter 'TransferMaxBatchSize'. With this parameter the external system can limit the batch size to a suitable value down to 0 or 1 in case that no batches are supported at all. The default maximum batch size is 10,000 members. If the external system sends a larger batch number, it will be reduced to 10,000.

Target Group Contact ID Entity Types

ExtTargetGroupMemberFacet (Entity Type)	Usage in Create method / Delete Method
Properties	
ExtTargetGroupId (Edm.String, length-50)	ID of the external target group to add the member to Always provided by SAP Marketing Cloud Ignored in response
ExtTGKeyPart2Id (Edm.String, length-50)	The ADVERTISER parameter Or for compatibility: ID of the first parameter marked to be part of the external target group key Provided by SAP Marketing Cloud if available Ignored in response
ExtTGKeyPart2Value (Edm.String, length-50)	Value of this first parameter Provided by SAP Marketing Cloud if available Ignored in response

ExtTargetGroupMemberFacet (Entity Type)	Usage in Create method / Delete Method
ExtTGKeyPart3Id (Edm.String, length-50)	<p>The EXT_CAMPAIGN_MANAGING_PARTY parameter</p> <p>Or for compatibility: ID of the second parameter marked to be part of the external target group key</p> <p>Provided by SAP Marketing Cloud if available</p> <p>Ignored in response</p>
ExtTGKeyPart3Value (Edm.String, length-50)	<p>Value of this second parameter</p> <p>Provided by SAP Marketing Cloud if available</p> <p>Ignored in response</p>
IdOrigin (Edm.String, length-20)	<p>Origin of the target group member ID (EMAIL, PHONE, ...)</p> <p>Always provided by SAP Marketing Cloud</p> <p>Ignored in response</p>
Id (Edm.String, length-100)	<p>ID of the target group member to be created (the actual email address, the phone number, ...), hashed if hashing was requested</p> <p>Always provided by SAP Marketing Cloud</p> <p>Ignored in response</p>

There are 4 different types of call that are sent from SAP Marketing Cloud for external member transfer.

- Single Creation
- Batch Creation
- Single Deletion
- Batch Deletion

## Example of Single Creation of External Target Group Member

### OData Request

An example of a single creation request as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/ExtTargetGroupMemberFacetSet`
- HTTP Method: POST

### Request Payload

#### Sample Request Payload

```
{
  "ExtTargetGroupId": "Ext123",
  "ExtTGKeyPart2Id": "ADVERTISER",
  "ExtTGKeyPart2Value": "SAP_GLOBAL_MARKETING",
  "ExtTGKeyPart3Id": "",
}
```

```

    "ExtTGKeyPart3Value": "",
    "IdOrigin": "EMAIL",
    "Id": "75304ebddec51e37966325d7950229110177ec502248d106cc29ccd8612bb75f"
  }

```

## Response Payload

It is mandatory that the http header contains the value ( ~status\_code: 201 ~status\_reason : Created ) after successful creation of members in external system.

### Sample Response Payload

```

{
  "d": {
    "results": [{
      "ExtTargetGroupId": "Ext123",
      "ExtTGKeyPart2Id": "ADVERTISER",
      "ExtTGKeyPart2Value": "SAP_GLOBAL_MARKETING",
      "ExtTGKeyPart3Id": "",
      "ExtTGKeyPart3Value": "",
      "IdOrigin": "EMAIL",
      "Id":
"75304ebddec51e37966325d7950229110177ec502248d106cc29ccd8612bb75f"
    }]
  }
}

```

## Example of Batch Creation of External Target Group Member

### OData Request

An example of a batch creation request as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/$batch.`
- HTTP Method: POST
- HEADERS: Content-Type: multipart/mixed;boundary=batch\_01869434-0008

#### i Note

Please be aware of the blank lines in the sample batch payloads, they must be maintained and kept empty.

### Request Payload

#### Sample Batch Request Payload

```

--batch_01869434-0008
Content-Type: multipart/mixed; boundary=changeset_01869434-0005-0002

--changeset_01869434-0005-0002
Content-Type: application/http
Content-Transfer-Encoding: binary

POST ExtTargetGroupMemberFacetSet HTTP/1.1
Accept-Language: en
Accept: application/json
MaxDataServiceVersion: 2.0
DataServiceVersion: 2.0

```

```

Content-Type: application/json

{"ExtTargetGroupId":"Ext123","ExtTGKeyPart2Id":"ADVERTISER","ExtTGKeyPart2Value":
"SAP_GLOBAL_MARKETING","ExtTGKeyPart3Id":"","ExtTGKeyPart3Value":"","IdOrigin":"E
MAIL","Id":"75304ebddec51e37966325d7950229110177ec502248d106cc29ccd8612bb75f"}

--changeset_01869434-0005-0002

--changeset_01869434-0005-0002
Content-Type: application/http
Content-Transfer-Encoding: binary

POST ExtTargetGroupMemberFacetSet HTTP/1.1
Accept-Language: en
Accept: application/json
MaxDataServiceVersion: 2.0
DataServiceVersion: 2.0
Content-Type: application/json

{"ExtTargetGroupId":"Ext123","ExtTGKeyPart2Id":"ADVERTISER","ExtTGKeyPart2Value":
"SAP_GLOBAL_MARKETING","ExtTGKeyPart3Id":"","ExtTGKeyPart3Value":"","IdOrigin":"E
MAIL","Id":"76304ebddec51e37966325d7950229110177ec502248d106cc29ccd8612bb75e"}

--changeset_01869434-0005-0002--

--batch_01869434-0008--

```

## Response Payload

The batch request response headers should contain ~status\_reason: Accepted, ~status\_code: 202 for a valid request.

It's mandatory to maintain the order of response on batch response payload to match with the requests in request payload. For every request there is a response expected in the response payload.

The value '201 Created' in response is mandatory which denotes the status code and status reason of a successful response.

In case of error suitable error message and status code, should be available in response. Please check odata documentation for batch handling. <http://www.odata.org/documentation/odata-version-2-0/batch-processing/>

## Sample Batch Response Payload

```

--8ECFC9976DBAAC45349E7A1DAC19BE200
Content-Type: multipart/mixed; boundary=8ECFC9976DBAAC45349E7A1DAC19BE201
Content-Length: 2995

--8ECFC9976DBAAC45349E7A1DAC19BE201
Content-Type: application/http
Content-Length: 1348
content-transfer-encoding: binary

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 927

--8ECFC9976DBAAC45349E7A1DAC19BE201
Content-Type: application/http
Content-Length: 1348
content-transfer-encoding: binary

```



```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 927

--8ECFC9976DBAAC45349E7A1DAC19BE201--

--8ECFC9976DBAAC45349E7A1DAC19BE200--
```

## Example of Single Deletion of External Target Group Member

### OData Request

An example of a single deletion request as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/ExtTargetGroupMemberFacetSet (ExtTargetGroupId='Ext123', ExtTGKeyPart2Id='ADVERTISER', ExtTGKeyPart2Value='SAP_GLOBAL_MARKETING', ExtTGKeyPart3Id='', ExtTGKeyPart3Value='', IdOrigin='EMAIL', Id='75304ebddec51e37966325d7950229110177ec502248d106cc29ccd8612bb75f')`.
- HTTP Method: DELETE

There is no request payload, and no response payload is returned from the external system. The HTTP response header field (`~status_code :204`, `~status_reason: No Content`) represents successful deletion of member at the external system and it is mandatory.

### Response Payload

## Example of Batch Deletion of External Target Group Member

### OData Request

An example of a batch deletion request as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/$batch`.
- HTTP Method: POST
- HEADERS: Content-Type: multipart/mixed;boundary=batch\_01869434-0008

#### i Note

Please be aware of the blank lines in the sample batch payloads, they must be maintained and kept empty.

### Request Payload

#### Sample Batch Request Payload

```
--batch_01869434-0008
Content-Type: multipart/mixed; boundary=changeset_01869434-0005-0002

--changeset_01869434-0005-0002
Content-Type: application/http
Content-Transfer-Encoding: binary

DELETE
ExtTargetGroupMemberFacetSet (ExtTargetGroupId='Ext123', ExtTGKeyPart2Id='ADVERTISE
```

```

R',ExtTGKeyPart2Value='SAP_GLOBAL_MARKETING',ExtTGKeyPart3Id='',ExtTGKeyPart3Value='',IdOrigin='EMAIL',Id='75304ebddec51e37966325d7950229110177ec502248d106cc29ccd8612bb75f') HTTP/1.1

--changeset_01869434-0005-0002
Content-Type: application/http
Content-Transfer-Encoding: binary

DELETE
ExtTargetGroupMemberFacetSet (ExtTargetGroupId='Ext123',ExtTGKeyPart2Id='ADVERTISER',ExtTGKeyPart2Value='SAP_GLOBAL_MARKETING',ExtTGKeyPart3Id='',ExtTGKeyPart3Value='',IdOrigin='EMAIL',Id='76304ebddec51e37966325d7950229110177ec502248d106cc29ccd8612bb75e') HTTP/1.1

--changeset_01869434-0005-0002--

--batch_01869434-0008--

```

## Response Payload

The batch response headers should contain ~status\_reason: Accepted, ~status\_code: 202 for a valid request.

It's mandatory to maintain the order of response in the batch response payload to match with the requests in the request payload. For every request there is a response expected in the response payload.

The value '204 No Content' in the response is mandatory which denotes the status code and status reason of a successful response.

In case of error, a suitable error message and status code should be available in the response. Please check OData documentation for batch handling. <http://www.odata.org/documentation/odata-version-2-0/batch-processing/>

## Sample Batch Response Payload

```

--5113C8BC9FF8909118DE29520A93D9430
Content-Type: multipart/mixed; boundary=5113C8BC9FF8909118DE29520A93D9431
Content-Length: 437

--5113C8BC9FF8909118DE29520A93D9431
Content-Type: application/http
Content-Length: 71
content-transfer-encoding: binary

HTTP/1.1 204 No Content
Content-Length: 0
dataserviceversion: 2.0

--5113C8BC9FF8909118DE29520A93D9431
Content-Type: application/http
Content-Length: 71
content-transfer-encoding: binary

HTTP/1.1 204 No Content
Content-Length: 0
dataserviceversion: 2.0

--5113C8BC9FF8909118DE29520A93D9431--

```

### 4.3.2.1.3.3 Transferring Contact Attributes (Deprecated)

#### ⚠ Caution

This feature has been deprecated. To transfer contact attributes to external systems, we recommend that you set up an Open Channel Integration.

For more information, see [Open Channel Integration \[page 194\]](#).

### 4.3.2.1.4 Requesting Campaign Success Data

The periodic request for success data consists of two steps:

- The call for campaign-related administrative data
- The call for success data

The call for success data can result in either a synchronous transfer of data or an asynchronous transfer using a `ReportID`.

#### Calling Administrative Data

The administrative data includes the following:

- The date up to which SAP Marketing Cloud is supposed to request success data for the campaign. Typically, the date differs from the campaign termination date since the success data is collected after the campaign has ended, based on an attribution window, or because a terminated campaign is resumed for a certain period of time.
- Indication of the most important KPI for the campaign, which is displayed in SAP Marketing Cloud. Since the KPI may not yet be determined when the campaign is created, it is requested with the success data.

For the administrative data, SAP Marketing Cloud calls the `GetEntitySet` method of the `CampaignSet` entity set to filter for a list of campaigns (see the following table).

Note that each additional part of the campaign key (from `ServerCampaignIDContextCT`) is called separately (no filtering for multiple key parts in one `GetEntitySet` call).

Starting in 1611, there are different calls for `ADVERTISER` and `EXT_CAMPAIN_MANAGING_PARTY`.

GetEntitySet Method

Campaign (Entity Type)	Usage in GetEntitySet Method
ClientCampaignId	Not used, ignored in response
CampaignName	Not used, ignored in response No update of the campaign name in SAP Marketing Cloud from the external campaign
StartDate	Not used, ignored in response No update of the campaign start date in SAP Marketing Cloud from the external campaign
EndDate	Not used, ignored in response No update of the campaign end date in SAP Marketing Cloud from the external campaign
ServerCampaignId	ID of the campaign in the external system; used as filter by SAP Marketing Cloud Mandatory in response The ID is considered to be the first part of the campaign key on server side.
ReportId	ID used for asynchronous data transfer Combination of ReportId, ServerCampaignId, and Advertiser must be unique Optional in response, not needed for synchronous transfer
ServerCampaignUrl	Not used, ignored in response Link to the external campaign No update of the link to the external campaign in the campaign in SAP Marketing Cloud

**Campaign (Entity Type)****Usage in GetEntitySet Method**

MainKPI

Most important KPI, displayed in SAP Marketing Cloud

Determines the first measure tiles on the *Performance* tab of the campaign.

Main KPI Options

<b>Main KPI</b>	<b>First Tiles</b>
NumberOfAppEngagements	NumberOfAppEngagements, AdServingCostAppEngagementInDC
NumberOfAppInstalls	NumberOfAppInstalls, AdServingCostPerAppInstallInDC
NumberOfClicks	NumberOfClicks, AdServingCostPerClickInDC, AdServingCostPer1000ClicksInDC, ClickThroughRateInPercent
NumberOfDownloads	NumberOfDownloads, AdServingCostPerDownloadInDC
NumberOfEventResponses	NumberOfEventResponses, AdServingCostPerEventRsplnDC
NumberOfImpressions	NumberOfImpressions, AdServingCost1000ImprnsInDC
NumberOfLeads	NumberOfLeads, AdServingCostPerLeadInDC
NumberOfMktgOfferClaims	NumberOfMktgOfferClaims, AdServingCostPerOfferClaimInDC
NumberOfOrders	NumberOfOrders, AdServingCostPerOrderInDC, OrderAmountInDC
NumberOfPageLikes	NumberOfPageLikes, AdServingCostPerPageLikeInDC
NumberOfPostEngagements	NumberOfPostEngagements, AdServingCostPerPostEngmntInDC
NumberOfRegistrations	NumberOfRegistrations, AdServingCostRegistrationInDC
NumberOfVideoViews	NumberOfVideoViews, AdServingCostPerVideoViewInDC, AdServingCost1000VidViewsInDC, VideoViewedAverageInPercent
NumberOfWebsiteConversions	NumberOfWebsiteConversions, AdServingCostWebsiteCnvrnsInDC

Optional in response

If main KPI is not provided, *NumberOfImpressions* is the default.

Campaign (Entity Type)	Usage in GetEntitySet Method
SuccessDataEndDate	<p>Last date by which SAP Marketing Cloud requests administrative and success data for the campaign</p> <p>Mandatory in response</p> <p>The date can be changed until it is reached.</p> <p>The date can be provided some time after the creation of the campaign, however, it is required to prevent endless success data requests.</p>
SuccessDataTimeZone	<p>The time zone for which the success data is requested</p> <p>Optional in response</p> <p>If the time zone is missing, the success data is requested for UTC.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>For timezone UTC+05:30, the expected value from the SuccessDataTimeZone field is <b>UTC+53</b></li> <li>For timezone UTC-8:30, the expected the value from the SuccessDataTimeZone field is <b>UTC-83</b></li> </ul>
Filter	Filter by <i>ServerCampaignIDContextCT</i>
CampaignParameterValues	<p>Not used</p> <p>No update of external campaign parameters in SAP Marketing Cloud from the external campaign</p>

An example of a request without additional parameters in the campaign key, as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/../<YourService>/CampaignSet?$filter=(ServerCampaignId eq '54321' or ServerCampaignId eq '54322')&$format=json`
- HTTP Method: GET

An example of a request including the parameter `ADVERTISER` in the campaign key, as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/../<YourService>/CampaignSet?$filter=(ServerCampaignId eq '54321' or ServerCampaignId eq '54322') and ((Filter/KeyPart2Id eq 'ADVERTISER') and (Filter/KeyPart2Value eq 'SAP_GLOBAL_MARKETING'))&$format=json`
- HTTP Method: GET

Note that "Filter": {...} is optional for the SAP Marketing Cloud, and is therefore omitted after the following example.

Sample Response Payload:

```

Sample Code
{
  "d": {

```

```

"results": [{
  "Filter": {
    "KeyPart2Id": "",
    "KeyPart2Value": "",
    "KeyPart3Id": "",
    "KeyPart3Value": ""
  },
  "ServerCampaignId": "54321",
  "MainKPI": "CLICKS",
  "SuccessDataEndDate": "\/Date(1455494400000)\/",
  "SuccessDataTimeZone": "UTC"
},
{
  "ServerCampaignId": "54322",
  "MainKPI": "",
  "SuccessDataEndDate": "\/Date(1455494400000)\/",
  "SuccessDataTimeZone": "UTC"
}]
}

```

## Calling Actual Success Data

For the actual success data, SAP Marketing Cloud calls the *GetEntitySet* method of the *MarketingSuccessSet* to filter for a list of campaigns, and a date range (see the following table).

Note that each additional part of the campaign key is called separately.

The success data is requested for all relevant campaigns, for today, and for yesterday (as today's data still may change). If the success data changes for a longer period, it can be returned in addition (beyond the requested dates). SAP Marketing Cloud can request success data for the past to recover from errors, or to reconcile historic data in case of data inconsistencies.

SAP Marketing Cloud requests success data using paging. The number of pages (data) is specified in `$top` and `$skip` of a request. If the call for one page fails, the already received success data for all campaigns and dates is completely persisted, the partially retrieved data, such as the data for one gender, is discarded. To enable this procedure, SAP Marketing Cloud requests the success data sorted by `ServerCampaignId` and `Date`.

SAP Marketing Cloud stores the success data completely per campaign, and per date, and it overwrites existing success data completely even when more data is returned than actually requested.

## Synchronous and Asynchronous Response

The external system can respond with the success data immediately (synchronous response) or with a `ReportID` (asynchronous response). If a `ReportID` is received, a second *GET* request will be sent with `ReportID`. This process repeats every four hours.

## GetEntitySet Method

<b>MarketingSuccess (Entity Type)</b>	<b>Usage in GetEntitySet Method</b>
ServerCampaignId (Edm.String, length-32)	<p>ID of the campaign in the external system. The ID is expected to be the first part of the campaign key on server side.</p> <p>Semantic key (characteristic)</p> <p>Used as filter by SAP Marketing Cloud</p> <p>Mandatory in response; to be sorted</p>
Date (Edm.DateTime)	<p>Date of the success data</p> <p>Semantic key (characteristic)</p> <p>Used as filter by SAP Marketing Cloud</p> <p>Mandatory in response, to be sorted</p> <p>YearWeek or YearMonth may be used instead. However, only one of the three is allowed in a single record.</p>
YearWeek (Edm.String, length-6)	<p>The year and week associated with the success data.</p> <p>Semantic key (characteristic)</p> <p>Used as filter by SAP Marketing Cloud</p> <p>Mandatory in response, to be sorted</p> <p>Date or YearMonth may be used instead. However only one of the three is allowed in a single record.</p>
YearMonth (Edm.String, length-6)	<p>The year and month associated</p> <p>Semantic key (characteristic)</p> <p>Used as filter by SAP Marketing Cloud</p> <p>Mandatory in response, to be sorted</p> <p>Date or YearWeek may be used instead. However only one of the three is allowed in a single record.</p>



MarketingSuccess (Entity Type)	Usage in GetEntitySet Method
TimeZone (Edm.String, length-6)	<p>Time zone to which the date refers</p> <p>Time zone is not characteristic (not part of semantic success data key)</p> <p>Used as filter by SAP Marketing Cloud</p> <p>Optional in response. If no time zone is provided, UTC is used.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>For timezone UTC+05:30, the expected value from the TimeZone field is <b>UTC+53</b></li> <li>For timezone UTC-8:30, the expected the value from the TimeZone field is <b>UTC-83</b></li> </ul> <p>A filter for TimeZone is set when a <code>SuccessDataTimeZone</code> is provided with <code>Campaign/GetEntitySet</code></p>
CommunicationMedium (Edm.String, length-20)	<p>The communication medium</p> <p>Semantic key (characteristic)</p> <p>Mandatory in response</p> <p>Relevant communication media: <code>DISPLAY_ADS</code> <code>MOBILE_ADS</code></p>
Gender (Edm.String, length-1)	<p>Gender to which the success data is related</p> <p>SAP Marketing Cloud provides a mapping of external to internal gender codes.</p> <p>Standard: <code>FEMALE</code>   <code>MALE</code></p> <p>Semantic key (characteristic)</p> <p>Optional in response; if not provided, the gender is considered as unknown</p>
Country(Edm.String, length 40)	<p>Country that the success data is related to</p> <p>SAP Marketing Cloud provides a mapping of external to internal country codes.</p> <p>Semantic key (characteristic)</p> <p>Optional in response; if not provided, the country is considered as unknown</p>
Region(Edm.String, length 40)	<p>Region that the success data is related to</p> <p>SAP Marketing Cloud provides a mapping of external to internal region codes.</p> <p>Semantic key (characteristic)</p> <p>Optional in response; if not provided, the region is considered as unknown</p>

## MarketingSuccess (Entity Type) Usage in GetEntitySet Method

---

AgeRangeLow (Edm.Byte, length-3)	Lower boundary of age  AgeRangeLow and AgeRangeHigh specify the age range to which the success data is related. The age ranges should not differ for all success data from the same external system. If data is available for each exact age, set AgeRangeLow and AgeRangeHigh to the same value.  Semantic key (characteristic)  Optional in response; if not provided, the age is considered as unknown
AgeRangeHigh (Edm.Byte, length-3)	Higher boundary of age  Semantic key (characteristic)  Optional in response; if not provided the age is considered as unknown
CampaignContentLinkName (Edm.String)	Name of the link in the campaign content that the success data refers to  Typically only provided for the KPIs "Clicks" and "UniqueClicks"  Semantic key (characteristic)  Optional in response
SpendAmount (Edm.Decimal, Precision- 31, scale -2)	Amount spend for the campaign on the external platform  KPI  Optional in response
SpendCurrency (Edm.String, length-5)	Currency of SpendAmount  One currency per campaign is supported  Mandatory in response when SpendAmount is provided
Uniquelmpressions (Edm.Int32, length-10)	Number of unique impressions  KPI  Optional in response
Impressions (Edm.Int32, length-10)	Number of impressions  KPI  Optional in response
UniqueClicks (Edm.Int32, length-10)	Number of unique clicks  KPI  Optional in response

---

<b>MarketingSuccess (Entity Type)</b>	<b>Usage in GetEntitySet Method</b>
Clicks (Edm.Int32, length-10)	Number of clicks KPI Optional in response
Orders (Edm.Int32, length-10)	Number of orders KPI Optional in response
OrderAmount (Edm.Decimal, Precision- 31, scale -2)	Monetary value of the orders KPI Optional in response
OrderAmountCurrency (Edm.String, length-5)	Currency of OrderAmount KPI Optional in response
VideoViews (Edm.Int32, length-10)	Number of video views KPI Optional in response
VideoViewedAverageInPercent (Edm.Decimal, Precision- 5, scale -2)	Average percentage of video viewed KPI Optional in response
Registrations (Edm.Int32, length-10)	Number of registrations KPI Optional in response
Downloads (Edm.Int32, length-10)	Number of downloads KPI Optional in response
SentMessages (Edm.Int32, length-10)	Number of sent messages KPI Optional in response
RejectedMessages (Edm.Int32, length-10)	Number of rejected messages KPI Optional in response

<b>MarketingSuccess (Entity Type)</b>	<b>Usage in GetEntitySet Method</b>
DeliveredMessages (Edm.Int32, length-10)	Number of delivered messages KPI Optional in response
OpenedMessages (Edm.Int32, length-10)	Number of opened messages KPI Optional in response
HardBounces (Edm.Int32, length-10)	Number of hard bounces KPI Optional in response
SoftBounces (Edm.Int32, length-10)	Number of soft bounces KPI Optional in response
PageLikes (Edm.Int32, length-10)	Number of page likes KPI Optional in response
PostEngagements (Edm.Int32, length-10)	Number of post engagements KPI Optional in response
OfferClaims (Edm.Int32, length-10)	Number of offer claims KPI Optional in response
WebsiteConversions (Edm.Int32, length-10)	Number of website conversions KPI Optional in response
AppInstalls (Edm.Int32, length-10)	Number of app installs KPI Optional in response
AppEngagements (Edm.Int32, length-10)	Number of app engagements KPI Optional in response

MarketingSuccess (Entity Type)	Usage in GetEntitySet Method
Filter	Filter by <code>ServerCampaignIDContextCT</code>
GrossRatingPoints (Edm.Decimal, Precision- 15, scale -2)	<p>The number impressions for a defined population in relation to the size of this population.</p> <p>Gross rating points are defined as <math>100 * impressions / size\ of\ defined\ population</math></p> <p>KPI</p> <p>Optional in response</p>
GrossRatingPointBase (Edm.String, length-80)	<p>The population the gross rating points relate to – for example, "US M18-39", which indicates that the gross rating points relate to male adults from 18 to 39 in the US.</p> <p>Semantic key (characteristic)</p> <p>Optional in response; it's recommended to always provide a gross rating point base together with gross rating points.</p>
InteractionReason (Edm.String, length-20)	<p>Reason for the interaction</p> <p>Possible values can be found in configuration. For more information, see <a href="#">Managing Interaction Content</a>.</p> <p>Semantic key (characteristic)</p> <p>Optional in response</p>
InteractionType (Edm.String, length-20)	<p>Type of interaction</p> <p>Possible values can be found in configuration. For more information, see <a href="#">Managing Interaction Content</a>.</p> <p>Semantic key (characteristic)</p> <p>Optional in response</p>
InteractionStatus (Edm.String, length-2)	<p>Status of interaction</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <b>01</b> (In Process)</li> <li>• <b>02</b> (Released)</li> <li>• <b>03</b> (Completed)</li> <li>• <b>04</b> (Canceled)</li> <li>• <b>05</b> (Converted)</li> <li>• <b>06</b> (Successful)</li> <li>• <b>07</b> (Unsuccessful)</li> <li>• <b>00</b> (New)</li> </ul> <p>Semantic key (characteristic)</p> <p>Optional in response</p>

<b>MarketingSuccess (Entity Type)</b>	<b>Usage in GetEntitySet Method</b>
DeviceType (Edm.String, length-60)	The type of device, for example tablet or desktop Semantic key (characteristic) Optional in response
AdNetwork (Edm.String, length-60)	The company that connects advertisers to the websites that host advertisements Semantic key (characteristic) Optional in response
CampaignContentName (Edm.String, length-100)	Name of the campaign content that the success data refers to Semantic key (characteristic) Optional in response
ExecutedInteractions (Edm.Int64, length-10)	Number of executed interactions KPI Optional in response
EventResponses (Edm.Int64, length-10)	Number of event responses KPI Optional in response
Leads (Edm.Int64, length-10)	Number of leads KPI Optional in response
Opportunities (Edm.Int64, length-10)	Number of opportunities KPI Optional in response
PhoneCalls (Edm.Int64, length-10)	Number of phone calls KPI Optional in response
Appointments (Edm.Int64, length-10)	Number of appointments KPI Optional in response
FailedInteractions (Edm.Int64, length-10)	Number of failed interactions KPI Optional in response

MarketingSuccess (Entity Type)	Usage in GetEntitySet Method
OfferViews (Edm.Int64, length-10)	Number of offer views KPI Optional in response
EmailComplaints (Edm.Int64, length-10)	Number of email complaints KPI Optional in response
Tasks (Edm.Int64, length-10)	Number of Tasks KPI Optional in response
UniquelmpressionsInPercent (Edm.Decimal, Precision-5, scale-2)	Reach in percent KPI Optional in response
OpportunityAmount (Edm.Decimal, Precision- 31, scale -2)	Monetary value of the opportunity KPI Optional in response
OpportunityAmountCurrency (Edm.String, length-5)	Currency for opportunity value Semantic key (characteristic) Mandatory in response when OpportunityAmount is provided

An example of a campaign success data request, with no additional parameters in the campaign key, as sent by SAP Marketing Cloud:

### Note

Certain aggregated KPIs don't have values for "Date" or "Timezone". To support ;, the OData request calls have additional filters in the request: `((TimeZone eq ''), (Date eq null))`.

- Request URL: `https://<HostName>/.../<YourService>/MarketingSuccessSet?$filter=((ServerCampaignId eq '54321') or (ServerCampaignId eq '54322')) and ((TimeZone eq 'UTC') or (TimeZone eq '')) and ((Date ge datetime'2016-01-27T00:00:00' and Date le datetime'2016-01-28T00:00:00') or (Date eq null))&$top=50&$skip=50&$format=json&$orderby=ServerCampaignId,Date desc`
- HTTP Method: GET

An example of a campaign success data request, including the parameter `ADVERTISER` in the campaign key, as sent by SAP Marketing Cloud:

- Request URL: `https://<HostName>/.../<YourService>/MarketingSuccessSet? $filter=((ServerCampaignId eq '54321') or (ServerCampaignId eq '54322')) and (Filter/KeyPart2Id eq 'ADVERTISER' and Filter/KeyPart2Value eq 'SAP_GLOBAL_MARKETING')and ((TimeZone eq 'UTC') or (TimeZone eq '')) and ((Date ge datetime'2016-01-27T00:00:00' and Date le datetime'2016-01-28T00:00:00') or (Date eq null))&$top=50&$skip=50&$format=json&$orderby=ServerCampaignId,Date desc`
- HTTP Method: GET

Sample Response Payload (Synchronous):

### Sample Code

```
{
  "d": {
    "results": [{
      "ServerCampaignId": "54321",
      "Date": "\\",
      "CommunicationMedium":
      "CampaignContentLinkName": "",
      "TimeZone": "UTC",
      "Gender": "2",
      "Country": "",
      "Region": "",
      "AgeRangeLow": 55,
      "AgeRangeHigh": 64,
      "SpendAmount": "60.00",
      "SpendCurrency": "USD",
      "UniqueImpressions": 0,
      "Impressions": 511,
      "Clicks": 5660,
      "UniqueClicks": 0,
      "Orders": 0,
      "OrderAmount": "70.00",
      "OrderCurrency": "USD",
      "Registrations": 0,
      "Downloads": 0,
      "HardBounces": 0,
      "SoftBounces": 0,
      "SentMessages": 0,
      "RejectedMessages": 0,
      "OpenedMessages": 0,
      "DeliveredMessages": 0,
      "PageLikes": 0,
      "PostEngagements": 0,
      "OfferClaims": 0,
      "VideoViews": 0,
      "VideoViewedAverageInPercent":
      "WebsiteConversions": 0,
      "AppInstalls": 0,
      "AppEngagements": 0,
      "GrossRatingPoints": "0.00",
      "GrossRatingPointBase": "",
      "YearWeek": "201751",
      "YearMonth": "",
      "InteractionReason": "",
      "InteractionType": "",
      "InteractionStatus": "06",
      "DeviceType": "DESKTOP",
      "AdNetwork": "GOOGLE_SEARCH",
      "CampaignContentName": "",
      "ExecutedInteractions": 0,
      "Date (1487635200000) \\",
      "DISPLAY_ADS",
      "0.00",
    }
  ]
}
```



```

    "EventResponses": 0,
    "Leads": 100,
    "Opportunities": 20,
    "PhoneCalls": 3612,
    "Appointments": 200,
    "FailedInteractions": 0,
    "OfferViews": 0,
    "EmailComplaints": 0,
    "Tasks": 0,
    "UniqueImpressionsInPercent":
"0.00",
    "OpportunityAmount": "70.00",
    "OpportunityAmountCurrency":
"USD",
  }
}

```

Sample Response Payload with Report ID (Asynchronous):

#### Sample Code

```

{
  "d": {
    "results": [{
      "ServerCampaignId": "54321",
      "ReportId": "4711"
    },
    {
      "ServerCampaignId": "54322",
      "ReportId": "4711"
    }
  ]
}

```

An example of a campaign success data request with ReportId:

- Request URL: `https://<HostName>/.../<YourService>/MarketingSuccessSet?$filter=( ReportId eq '4711') and (Filter/KeyPart2Id eq 'ADVERTISER' and Filter/KeyPart2Value eq 'SAP_GLOBAL_MARKETING') &$top=50&$skip=50&$format=json`
- HTTP Method: GET

Sample Response Payload:

#### Sample Code

```

{
  "d": {
    "results": [{
      "ServerCampaignId": "54321",
      "Date": "\/Date(1474588800000)\/",
      "CommunicationMedium": "DISPLAY_ADS",
      "SpendAmount": "60.00",
      "SpendCurrency": "USD",
      "UniqueImpressions": 0,
      "Impressions": 511,
      "Clicks": 32,
      "ReportId": "4711"
    },
    {
      "ServerCampaignId": "54322",
      "Date": "\/Date(1474588800000)\/",
      "CommunicationMedium": "DISPLAY_ADS",
      "AppEngagements": 300,
    }
  ]
}

```

```

    "ReportId": "4711"
  }
}
}

```

If no data is available, an empty response will be received. It may also be the case that there is not data available for all campaigns with the *ReportID*, and the response will only contain campaigns for which there is data available.

## Related Information

[Creating Custom OData Fields for Campaign Success Data \[page 190\]](#)

### 4.3.2.1.4.1 Creating Custom OData Fields for Campaign Success Data

#### Prerequisites

- Define the custom field in the *Custom Fields* application with the business context **Marketing: Campaign Performance Actual Measure**.

For more information about setting up custom fields, see [Creating Custom Fields](#).

For more information about custom fields for campaign performance, including dimensions and target measures in addition to actual measures, see [Custom Fields for Campaign Performance](#).

When you enter a label, an identifier is automatically generated.

- Example custom field label: Tickets Sold
- Generated identifier: YY1\_TicketsSold\_MCS

The prefix YY1 and suffix MCS are automatically generated.

- In the external campaign interface, add the custom field under *MarketingSuccess Entity* as a property. The field name must be the same throughout. Example: YY1\_TicketsSold\_MCS

Once defined, the external interface can send the value for the new field.

Sample OData response with the custom field YY1\_TicketsSold\_MCS:

#### Sample Code

```

{
  "d": {
    "results": [{
      "ServerCampaignId": "54321",
      "Date": "\/
Date (1487635200000) \/",
      "CommunicationMedium":
      "DISPLAY_ADS",
      "SpendAmount": "60.00",
      "SpendCurrency": "USD",
      "Impressions": 511,
    }
  ]
}

```

```

    "Clicks": 5660,
    "YY1_TicketsSold_MCS": "3443"
  }
}
]]
}

```

### 4.3.2.1.5 Handling Errors

If an error occurs when retrieving the required parameters, or when creating the campaign in the external system, SAP Marketing Cloud blocks the process until the error is resolved. The error messages are displayed in your application.

If an error occurs when transferring target group members, the error log can be accessed on the Automation panel, in the *Create External Target Group* section.

If an error occurs when retrieving the success data, a red status for the success data retrieval is displayed in your application. Clicking the red status provides the detailed error messages. If one request for success data fails, the already retrieved success is still available. If requests for success data fail, the requests are automatically repeated with the next success data retrieval.

For the error response, `error-code`, and `error-message-value` are mandatory. Further error messages can be returned as `error-innererror`.

Example of an error response:

#### Sample Code

```

{
  "error":{
    "code":"123",
    "message":{
      "lang":"en",
      "value":"Your error message describing the issue"
    },
    "innererror":{
      "errordetails":[{
        "code":"234",
        "message":"Your description for the error resolution",
        "severity":"error"
      }]
    }
  }
}

```

For more information about error handling, see <http://www.odata.org/documentation/odata-version-2-0/operations/>.

## 4.3.2.2 Communication Arrangement for External Campaign Execution

You set up a communication arrangement to enable the external campaign execution, and the requesting of success data from the external system. To set up a communication arrangement, you require the business catalog role *Marketing – Business Administration* (`SAP_BCR_CEC_MKT_ADM_PC`).

To set up a communication arrangement for external campaign execution you create a communication system, and a communication arrangement.

### Communication System

Create the communication system as follows:

1. In the SAP Fiori launchpad, click [Communication Systems](#). In *Communication Systems*, click [New](#).
2. In the *New Communication System* dialog, define the *ID* for the communication system, for example, `Z_CUAN_ECPG`. Define a *System Name*. You can freely define a name; note that the name is used when you create the communication arrangement. Click [Create](#).
3. Under *Technical Data, Host Name*, specify the external system you want to use for the campaign execution, such as Facebook, or Twitter. Indicate the pure host name, no path, no port. Note that *Log System ID*, *Client Name*, and *Business System* are not relevant for the external campaign execution.
4. Optionally, you can provide your *Contact Information* for the communication system you are defining.
5. Under *User for Outbound Communication*, click [+](#) to add a set of access details for the external server. One option is to use the *Authentication Method* user and password by entering the information in the corresponding fields. Alternatively, you can use authentication via an SSL client certificate. For this option, you need to select *Default Client Certificate* as the certificate type and then download the certificate before creating the outbound user.
6. Click [Save](#) to save the new or edited communication system in an active status.

If you chose the SSL client certificate as your authentication method, you will need to upload the certificated you downloaded to your external server. For example, if you have implemented the interface for externally executed campaigns on SAP Cloud Integration, the certificate has to be uploaded to the HTTP channel of your integration flow.

### Communication Arrangement for Use of External Platforms in Multichannel Campaigns

Create the communication arrangement as follows:

1. In the SAP Fiori launchpad, click [Communication Arrangement](#). In *Maintain Communication Arrangement*, click [New](#).
2. In the *New Communication Arrangement* dialog, under *Scenario*, use the value help to select the predefined scenario *Marketing - External Campaign Execution Integration* (`SAP_COM_0037`). Under *Arrangement Name*, define a name.

3. Under *Common Data, Communication System*, use the value help to select the communication system you have created (see section *Communication System*).  
Note that *My System* is not relevant for external campaign execution.
4. Under *Outbound Communication*, use the value help to select the relevant system access details, which you have specified in the communication system you are using for the communication arrangement.  
For multichannel campaigns, OAuth authentication is not supported.
5. Under *Additional Properties*, define an *External Campaign System ID*. If you are setting up a communication for Google Campaign Manager, use the code **SDM**. For other external platforms, use a three letter code that starts with **z**. This ID cannot be changed later.  
Here you can also define an action name. This name will be used for the multichannel action in the campaign designer.
6. Under *Additional Properties*, you can specify a marketing area for which this action is relevant. If you do, the action will only be available in a multi-channel campaign that has the same marketing area.
7. Under *Outbound Services*, specify the paths for the predefined outbound services using the following pattern: */<your\_service>/<your\_entity>*. The predefined services map to the following actions (specified as entity):
  - Request campaign parameters (**CampaignParameterSet**)
  - Create campaign (**CampaignSet**)
  - Request success data for the executed campaign (**MarketingSuccessSet**)
  - Transferring external target group (**ExtTargetGroupSet**)
  - Transferring external target group members (**ExtTargetGroupMemberFacetSet**)
  - Read supported ID origins for external system (**ExtTargetGroupSupportedIdOriginSet**)
  - Retrieve information for the value help for campaign assignment (**CampaignValueHelpSet**)
 Note that the outbound services are defined in the *Scenario* you have selected when creating the communication arrangement.
8. Click [Save](#) to save the new or edited communication arrangement in an active status.

## Communication Arrangement for External-Only Campaigns

You require a campaign category for campaigns that are externally executed only. Your system comes with a general category that can be set up for use with the external system of your choice. Only one external campaign category is available, but may be edited using the configuration app [Define Campaign Categories and Actions](#). If you need more external campaign categories, you can copy the delivered entry and enter a unique ID and your desired name for the new category.

Typically, one campaign category corresponds to one external campaign execution system.

### **i** Note

To perform HTTP calls to an external system from SAP Marketing Cloud, a trust relationship with the external system is required. If this relationship is not established by default, such as when using SAP Cloud Integration, you have to upload the root certificate of the external system's host in the [Maintain Certificate Trust List](#) app in SAP Marketing Cloud.

Create the communication arrangement as follows:

1. In the SAP Fiori launchpad, click [Communication Arrangement](#). In *Maintain Communication Arrangement*, click [New](#).

2. In the *New Communication Arrangement* dialog, under *Scenario*, use the value help to select the predefined scenario *Marketing - External Campaign Execution Integration (SAP\_COM\_0037)*.

Under *Arrangement Name*, define a name using the following pattern:

**ExtCampaignExec\_<campaign\_category>**, for example, **ExtCampaignExec\_EEC**. For authorization purposes, you can optionally include a marketing area in the arrangement name. To include the marketing area, extend the pattern as follows: **ExtCampaignExec\_<campaign\_category>\_<marketing\_area>**.

If you add a marketing area ID in the communication arrangement name, such arrangement will be use to send data to the external system when executing external campaigns which have the same marketing area. If more marketing areas are used, you need to create one communication arrangement per marketing area.

If a communication arrangement for a given marketing area is not found, the system uses a more generic arrangement which name pattern is **ExtCampaignExec\_<campaign\_category>**.

3. Under *Common Data, Communication System*, use the value help to select the communication system you have created (see section *Communication System*).

Note that *My System* is not relevant for external campaign execution.

4. Under *Outbound Communication*, use the value help to select the relevant system access details, which you have specified in the communication system you are using for the communication arrangement. For external-only campaigns, OAuth authentication is not supported.

5. **Do not** define an *External Campaign System ID* under *Additional Properties*. This will mark the communication arrangement for use in multichannel campaigns and not external-only campaigns.

6. Under *Outbound Services*, specify the paths for the predefined outbound services using the following pattern: */<your\_service>/<your\_entity>*. The predefined services map to the following actions (specified as entity):

- Request campaign parameters (**CampaignParameterSet**)
- Create campaign (**CampaignSet**)
- Request success data for the executed campaign (**MarketingSuccessSet**)
- Transferring external target group (**ExtTargetGroupSet**)
- Transferring external target group members (**ExtTargetGroupMemberFacetSet**)
- Read supported ID origins for external system (**ExtTargetGroupSupportedIdOriginSet**)
- Retrieve information for the value help for campaign assignment (**CampaignValueHelpSet**)

Note that the outbound services are defined in the *Scenario* you have selected when creating the communication arrangement.

7. Click *Save* to save the new or edited communication arrangement in an active status.

### 4.3.3 Open Channel Integration

With this integration you create own actions that send data for further processing to an external system, such as SAP Business Technology Platform, when the campaign has been executed. But you can also just implement the inbound side of this integration to get external data in your campaigns.

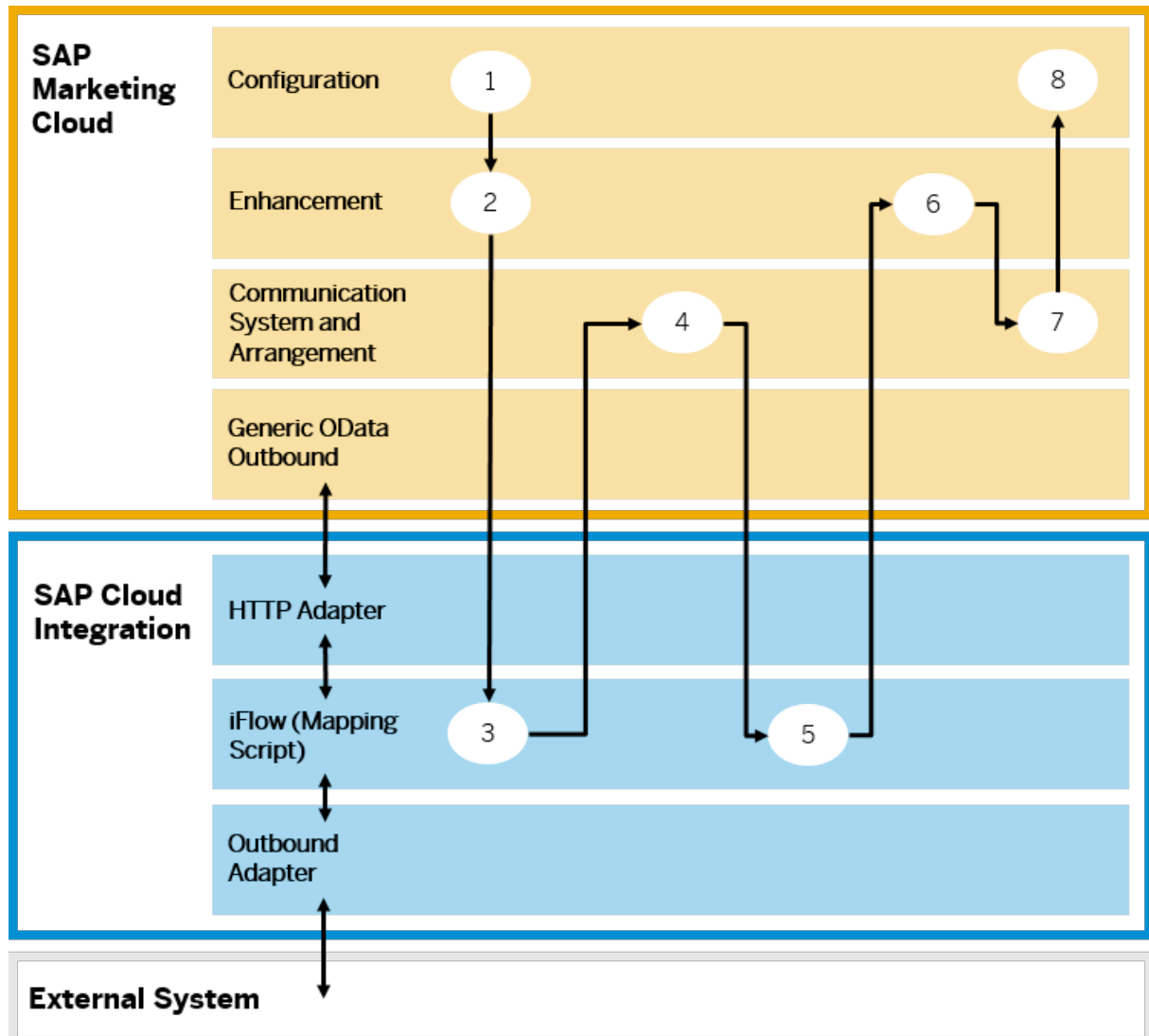
**Technology:** OData Service using SAP BTP

## Benefits

Develop custom logic for the campaign automation in SAP Marketing Cloud:

- Programmatically connect external applications using SAP BTP by developing custom actions in the campaign automation.
- Automatically provide follow-up data for any external system such as lettershop, mobile app, web shop, or service portal.
- Pass over all personalization data automatically for each contact using SAP BTP.
- In-place analytics for outbound and inbound interactions
- Campaign (with all actions) is executed in the SAP system
- Targeted contact is known.
- Guaranteed delivery.
- You can use SAP or partner infrastructure, as Amazon or SAP Cloud for Customer.
- Content needs to be provided by receiving system (not part of SAP Marketing)

## System Setup and Integration Steps



For setting up the Open Channel Integration you walk through the following steps:

1. Check that the business catalog role *Marketing - Segmentation and Campaign Configuration* (SAP\_CEC\_BC\_MKT\_CPC\_PC) has been assigned to your user.
2. [Adapt Enhancements \[page 206\]](#) (mandatory for outbound)  
For a minimal integration you have to implement only the enhancement (1) *Open Channel: Define Implementations* and define an *Implementation ID*.
3. [Setting Up SAP Business Technology Platform \[page 217\]](#) (mandatory for outbound)
4. [Create Communication Systems and Arrangements \[page 218\]](#) (mandatory for outbound)
5. Setup SAP BTP according to your needs. (mandatory for outbound)  
For more information for this step and the following ones, see [Inbound Service Settings API \[page 229\]](#).
6. Then adapt the enhancement (5) *Open Channel: Define Template for Outbound Interaction*. (mandatory for inbound)  
For more information, see [Processing Details \[page 198\]](#) and [Inbound Service Settings API \[page 229\]](#).



7. Create communication systems and arrangements for the inbound side. (mandatory for inbound)
8. Dependent on how you set up your integration and to be able to use the [Open Channel](#) action in your system, you create an export definition:
  1. Choose the [Export Definitions](#) app and under [Details](#) choose [New](#).
  2. Then enter a [Definition](#). You can freely define a name; note that the name is used when you assign the [Open Channel](#) action in the campaign user interface (UI).
  3. Select [Usage Open Channel](#).
  4. Select a [Segmentation Profile](#), for example, [All Contacts](#).
  5. Select an [Export Profile](#), for example, [File Export](#).
  6. Under [Available](#) select these attributes you want to use in your export definition and bring them to the list [Selected for Export Definition](#).
  7. Choose [Save](#) to save the new export definition.

#### i Note

This step is optional.

In the following you find some useful thoughts for the implementation:

- We recommend to plan the integration flow beforehand, because at least you need to implement the Processing integration flow, which is mandatory, and you can do everything in this implementation. But you can also do all three or only two of the offered open channel implementations, just depending of your needs.
- Preprocessing and Processing send their data in the deep create format that means the whole data structure is created directly.
- All processing steps of the integration send messages by HTTP Post method, but only the creation mode is supported which means this integration enforces the creation of new object instances in the external system, but not any updates and deletions.
- We recommend to use the HTTP sender channel at least for the processing step, because scripting might be required anyway for transforming the attribute IDs and values of the `TargetGroupMemberAttributeData` entity set into a new message.
- Read the information about the message choreography and the error handling in chapter [OData Service Settings for Outbound \[page 220\]](#).

## Create a Campaign

Now you can create a campaign and use the action [Open Channel](#).

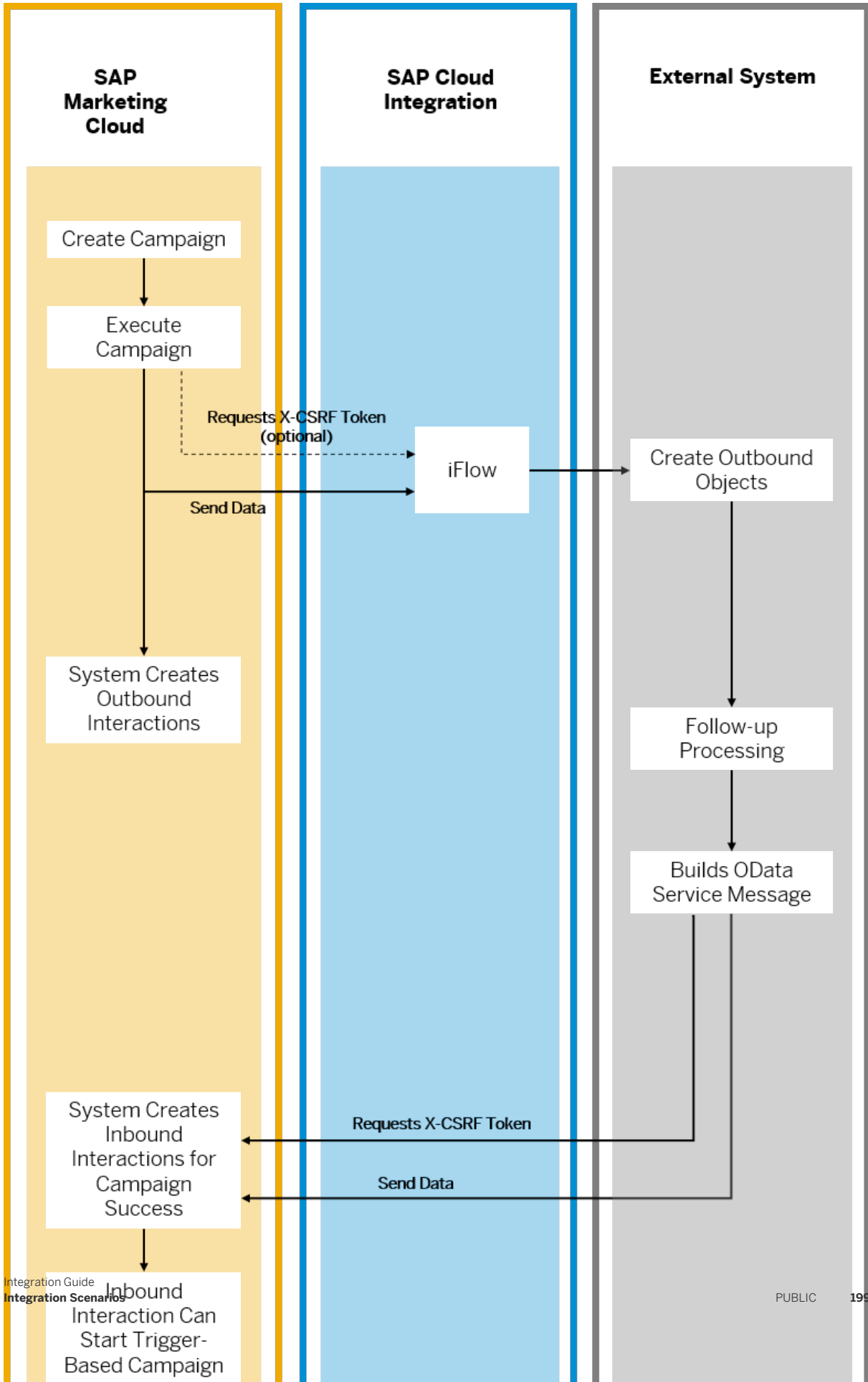
For more information about the general handling, see [Creating a Campaign](#) under [Use Cases](#).

### 4.3.3.1 Processing Details

With the implementation, you can execute a campaign and send outbound objects also to an external system using SAP Business Technology Platform.

When the work has been done in the external system, you can get back data to analyze your success and create follow-up triggers in your system, for example, by using a trigger-based campaign. The selection of transferred data is done by using an export definition.

The graphic shows the process steps and the systems that are involved:



## i Note

The following description is an example and demonstrates the steps and the data required to link outbound and inbound records for open channel processing.

## Campaign Execution

The campaign has a target group with the following members:

- Julie Armstrong
- John Miller
- Michael Adams

The campaign executes the open channel action. The action transfers data using the OData service `CUAN_CAMPAIGN_OPEN_CHANNEL` and the entity sets `CampaignExecutionRunPackages`, `CampaignTargetGroupMembers`, and `TargetGroupMemberAttributeData`:

### Sample Code

```
{
  "d": {
    "Campaign": {
      "CampaignId": "0000381379",
      "Name": "Open Channel Demo 1",
      "MarketingAreaId": "CXXGLOBAL",
      "SegmentationObject": "SAP_CONTACT_ENGAGEMENT_SIN",
      "ImplementationId": "ZOC_EXPORT"
    },
    "PackageId": 1,
    "ExecutionStartDateTime": "2016-07-07T07:44:40Z",
    "ExecutionRunKey": "8CD4A847681EE69182D1BBA1C39EF3",
    "CampaignTargetGroupMembers": [
      {
        "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10",
        "PackageId": 1,
        "ExecutionRunKey": "8CD4A847681EE69182D1BBA1C39EF3",
        "TargetGroupMemberAttributeData": [
          {
            "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_FIRST",
            "Value": "Julie",
            "EdmTypeId": "Edm.String",
            "OutboundId":
"1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
          }, {
            "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_LAST",
            "Value": "Armstrong",
            "EdmTypeId": "Edm.String",
            "OutboundId":
"1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
          }, {
            "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
CONTACT_KEY",
            "Value": "005056AC4A181ED598D20A84AB8AC6E9",
            "EdmTypeId": "Edm.Binary",
            "OutboundId":
"1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
          }, {
```

```

        "AttributeId": "OUTBOUND_INTERACTION",
        "Value": "8CDCD4A847681EE69182D4A1498E1EF5",
        "EdmTypeId": "Edm.String",
        "OutboundId":
"1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
    }, {
        "AttributeId": "ZOC_EXPORT_DESCRIPTION",
        "Value": "Open Channel Demo 1",
        "EdmTypeId": "Edm.String",
        "OutboundId":
"1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
    }, {
        "AttributeId": "ZOC_EXPORT_DATE",
        "Value": "2016-07-07",
        "EdmTypeId": "Edm.Date",
        "OutboundId":
"1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
    }, {
        "AttributeId": "ZOC_EXPORT_PRIORITY",
        "Value": "PRIORITY_1",
        "EdmTypeId": "Edm.String",
        "OutboundId":
"1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
    }
    ],
    {
        "OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23",
        "PackageId": 1,
        "ExecutionRunKey": "8CDCD4A847681EE69182D1BBA1C39EF3",
        "TargetGroupMemberAttributeData": [
            {
                "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_FIRST",
                "Value": "John",
                "EdmTypeId": "Edm.String",
                "OutboundId":
"EA297547B0DBBDF81C308FD14A3757C1420ABB23"
            }, {
                "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_LAST",
                "Value": "Miller",
                "EdmTypeId": "Edm.String",
                "OutboundId":
"EA297547B0DBBDF81C308FD14A3757C1420ABB23"
            }, {
                "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
CONTACT_KEY",
                "Value": "005056AC4A181ED598D20A84AB8B06E9",
                "EdmTypeId": "Edm.Binary",
                "OutboundId":
"EA297547B0DBBDF81C308FD14A3757C1420ABB23"
            }, {
                "AttributeId": "OUTBOUND_INTERACTION",
                "Value": "8CDCD4A847681EE69182D4A1498EDEF5",
                "EdmTypeId": "Edm.String",
                "OutboundId":
"EA297547B0DBBDF81C308FD14A3757C1420ABB23"
            }, {
                "AttributeId": "ZOC_EXPORT_DESCRIPTION",
                "Value": "Open Channel Demo 1",
                "EdmTypeId": "Edm.String",
                "OutboundId":
"EA297547B0DBBDF81C308FD14A3757C1420ABB23"
            }, {
                "AttributeId": "ZOC_EXPORT_DATE",
                "Value": "2016-07-07",
                "EdmTypeId": "Edm.Date",

```

```

        "OutboundId":
"EA297547B0DBBDF81C308FD14A3757C1420ABB23"
    }, {
        "AttributeId": "ZOC_EXPORT_PRIORITY",
        "Value": "PRIORITY_1",
        "EdmTypeId": "Edm.String",
        "OutboundId":
"EA297547B0DBBDF81C308FD14A3757C1420ABB23"
    }
]
}, {
    "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF",
    "PackageId": 1,
    "ExecutionRunKey": "8CDCD4A847681EE69182D1BBA1C39EF3",
    "TargetGroupMemberAttributeData": [
        {
            "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_FIRST",
            "Value": "Michael",
            "EdmTypeId": "Edm.String",
            "OutboundId":
"CA5F1FE120237480E6054B06D61371081AE095DF"
        }, {
            "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_LAST",
            "Value": "Adams",
            "EdmTypeId": "Edm.String",
            "OutboundId":
"CA5F1FE120237480E6054B06D61371081AE095DF"
        }, {
            "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
CONTACT_KEY",
            "Value": "40F2E93065BD1ED598D1DCFDB65F97C0",
            "EdmTypeId": "Edm.Binary",
            "OutboundId":
"CA5F1FE120237480E6054B06D61371081AE095DF"
        }, {
            "AttributeId": "OUTBOUND_INTERACTION",
            "Value": "8CDCD4A847681EE69182D4A1498F3EF5",
            "EdmTypeId": "Edm.String",
            "OutboundId":
"CA5F1FE120237480E6054B06D61371081AE095DF"
        }, {
            "AttributeId": "ZOC_EXPORT_DESCRIPTION",
            "Value": "Open Channel Demo 1",
            "EdmTypeId": "Edm.String",
            "OutboundId":
"CA5F1FE120237480E6054B06D61371081AE095DF"
        }, {
            "AttributeId": "ZOC_EXPORT_DATE",
            "Value": "2016-07-07",
            "EdmTypeId": "Edm.Date",
            "OutboundId":
"CA5F1FE120237480E6054B06D61371081AE095DF"
        }, {
            "AttributeId": "ZOC_EXPORT_PRIORITY",
            "Value": "PRIORITY_1",
            "EdmTypeId": "Edm.String",
            "OutboundId":
"CA5F1FE120237480E6054B06D61371081AE095DF"
        }
    ]
}
]
}
}

```

## i Note

The transferred OutboundId is kept by the external system because this ID serves as external anchor to link outbound and inbound records.

## Outbound Interactions

The campaign must create outbound interactions, otherwise the inbound data cannot be linked to the campaign.

To create outbound interactions you implement the enhancement **(5) Open Channel: Define Template for Outbound Interaction**.

### Sample Code

```
template-id_origin          = 'SAP_HYBRIS_MKT_IC'.
template-interaction_type   = 'ZOC_CALL_CENTER_OUTB'.
template-communication_medium = 'BUSINESS_DOCUMENT'.
```

The following outbound interactions have been created:

DB_KEY	ID_ORIGIN	ID	COMM_MEDI UM	IA_TYPE	SOURCE_OB JECT_TYPE	SOURCE_OB JECT_ID	TIMESTAMP	INITIATIV E_ID
1	SAP_HYBRI S_MKT_IC	005056AC4 A181ED598 D20A84AB8 AC6E9 =	BUSINESS_ DOCUMENT	ZOC_CALL_ CENTER_OU TB	CUAN_MARK ETING_ORC HESTRATIO N	1202654B2 1C72A50A0 D4E5CB35E 6D2FBE916	2016-07-0 7T07:48:4 8Z	381379
		contact key of Julie Armstrong				EA10		
2	SAP_HYBRI S_MKT_IC	005056AC4 A181ED598 D20A84AB8 B06E9 =	BUSINESS_ DOCUMENT	ZOC_CALL_ CENTER_OU TB	CUAN_MARK ETING_ORC HESTRATIO N	EA297547B 0DBBDF81C 308FD14A3 757C1420A	2016-07-0 7T07:48:4 8Z	381379
		contact key of John Miller				BB23		
3	SAP_HYBRI S_MKT_IC	40F2E9306 5BD1ED598 D1DCFDB65 F97C0 =	BUSINESS_ DOCUMENT	ZOC_CALL_ CENTER_OU TB	CUAN_MARK ETING_ORC HESTRATIO N	CA5F1FE12 0237480E6 054B06D61 371081AE0	2016-07-0 7T07:48:4 8Z	381379
		contact key of Michael Adams				95DF		

Meaning of the attributes:

- ID\_ORIGIN: set by enhancement coding, see `template-id_origin`
- ID: key of contact, set by campaign execution
- COMM\_MEDIUM: set by enhancement coding, see `template-communication_medium`
- IA\_TYPE: set by enhancement coding, see `template-interaction_type`
- SOURCE\_OBJECT\_TYPE: always `CUAN_MARKETING_ORCHESTRATION`, set by campaign execution
- SOURCE\_OBJECT\_ID: this ID is the `OutboundId`, set by campaign execution
- TIMESTAMP: time stamp when outbound happened, set by campaign execution
- INITIATIVE\_ID: ID of the campaign, set by campaign execution

## Inbound Interactions

In the external system, the transferred data is processed and the processing of the data for the contact **Michael Adams** results into an inbound interaction.

### i Note

Before calling the import service, the external system must request the X-CSRF-token: `/sap/opu/odata/sap/API_MKT_INTERACTION_SRV/InteractionsDeepInsert`

The external system uses the OData service `API_MKT_INTERACTION` and transfers the following data using the entity set `ImportHeaders`:

### Sample Code

```
{
  "UUID": "32575914-a9db-476c-a51a-2b0d4a899b95",
  "Interactions": [
    {
      "InteractionUUID":
"00000000-0000-0000-0000-000000000000",
      "InteractionType": "ZOC_CALL_CENTER_INB",
      "InteractionSourceObjectType":
"CUAN_CAMPAIGN_OUTBOUND",
      "InteractionSourceObject":
"18D3620CC1DBAEB8E5F97AFB922E84E092F271F0",
      "InteractionTimeStampUTC":
"2019-06-13T08:55:00"
    }, {
      "InteractionUUID":
"00000000-0000-0000-0000-000000000000",
      "InteractionType": "ZOC_CALL_CENTER_INB",
      "InteractionSourceObjectType":
"CUAN_CAMPAIGN_OUTBOUND",
      "InteractionSourceObject":
"0FB6D7D9DCDB8616A4A39D0E931C01DA57B5E48F",
      "InteractionTimeStampUTC":
"2019-06-13T08:55:00"
    }
  ]
}
```

The `OutboundId` is transferred with the `SourceObjectId`. By setting `SourceObjectType` to `CUAN_CAMPAIGN_OUTBOUND` the OData service knows that the given `OutboundId` belongs to an open channel



scenario and copies data from the outbound to the inbound record: InteractionContact, Campaign, CampaignExecutionRun, and so on, are determined by the OutboundId and therefore not provided by the external system.

For more information, see [Inbound Service Settings API \[page 229\]](#).

After a successful processing of the OData service, the interaction table contains the following outbound (DB\_KEY: 1 - 3) and inbound (DB\_KEY: 4) records:

DB_KEY	ID_ORIGIN	ID	COMM_MEDI UM	IA_TYPE	SOURCE_OB JECT_TYPE	SOURCE_OB JECT_ID	TIMESTAMP	INITIATIV E_ID
1	SAP_HYBRI	005056AC4	BUSINESS_	ZOC_CALL_	CUAN_MARK	1202654B2	2016-07-0	381379
	S_MKT_IC	A181ED598	DOCUMENT	CENTER_OU	ETING_ORC	1C72A50A0	7T07:48:4	
		D20A84AB8		TB	HESTRATIO	D4E5CB35E	8Z	
		AC6E9 =			N	6D2FBE916		
		contact			EA10			
		key of						
		Julie						
		Armstrong						
2	SAP_HYBRI	005056AC4	BUSINESS_	ZOC_CALL_	CUAN_MARK	EA297547B	2016-07-0	381379
	S_MKT_IC	A181ED598	DOCUMENT	CENTER_OU	ETING_ORC	0DBBDF81C	7T07:48:4	
		D20A84AB8		TB	HESTRATIO	308FD14A3	8Z	
		B06E9 =			N	757C1420A		
		contact			BB23			
		key of						
		John						
		Miller						
3	SAP_HYBRI	40F2E9306	BUSINESS_	ZOC_CALL_	CUAN_MARK	CA5F1FE12	2016-07-0	381379
	S_MKT_IC	5BD1ED598	DOCUMENT	CENTER_OU	ETING_ORC	0237480E6	7T07:48:4	
		D1DCFDB65		TB	HESTRATIO	054B06D61	8Z	
		F97C0 =			N	371081AE0		
		contact			95DF			
		key of						
		Michael						
		Adams						
4	SAP_HYBRI	40F2E9306	BUSINESS_	ZOC_CALL_	CUAN_MARK	CA5F1FE12	2016-07-2	381379
	S_MKT_IC	5BD1ED598	DOCUMENT	CENTER_IN	ETING_ORC	0237480E6	3T19:56:2	
		D1DCFDB65		<b>B</b>	HESTRATIO	054B06D61	9Z	
		F97C0 =			N	371081AE0		
		contact			95DF			
		key of						
		Michael						
		Adams						

The OData service API\_MKT\_INTERACTION changes the source object type from CUAN\_CAMPAIGN\_OUTBOUND back to CUAN\_MARKETING\_ORCHESTRATION. CUAN\_MARKETING\_ORCHESTRATION is the source object type of the corresponding outbound record.

## 4.3.3.2 Adapt Enhancements

You define an implementation for an enhancement to enable the campaign execution to use the open channel action.

To define an implementation, you require the business catalog role Communication Management (SAP\_CORE\_BC\_COM).

Create the implementation as follows:

1. Open the *Custom Logic* app and choose *Create* (+ icon)..
2. In the *New Enhancement Implementation* dialog select **Marketing: Campaign** as *Business Context* and as *Enhancement Option* the following enhancements depending of what you want to do:
  - (1) *Open Channel: Define Implementations* (**mandatory**)
  - (2) *Open Channel: Define Parameters for Implementation* (optional)
  - (3) *Open Channel: Define Global Settings for Execution* (optional)
  - (4) *Open Channel: Enhance Payload for Data Transfer* (optional)
  - (5) *Open Channel: Define Template for Outbound Interaction* (mandatory)

### i Note

The implementation of this enhancement is **mandatory** for creating any kind of interactions such as contact has been rejected by permission checks, and for processing inbound interactions. But it is **optional** for the outbound interactions.

3. Then enter a name and choose *Create*.
4. The example coding is automatically used in the draft version. You can use this coding or modify it.
5. Choose *Test* to test the coding.
6. Choose *Publish* to release your coding.

### i Note

The *Implementation ID*, such as ZOC\_EXPORT, will be used in the communication arrangement as property value.

For more information, see:

[Data Flow \[page 233\]](#)

[BAdI Details \[page 207\]](#)

Here you find additional information about the offered BAdIs for the open channel integration.

[Activating Marketing Permissions \[page 215\]](#)

With the following adaptations in the example coding of the enhancements you can activate marketing permissions for your open channel integration.

[Get Deviating Communication ID Based on Origin ID \[page 216\]](#)

You can get deviating communication IDs based on origin IDs, doing some coding in enhancements.

You need this switch, in case you have activated the enhancement (5) *Open Channel: Define Template for Outbound Interaction*. If the communication ID does not fit to the origin ID, the system cannot create interactions.

## 4.3.3.2.1 BAdI Details

Here you find additional information about the offered BAdIs for the open channel integration.

### → Recommendation

We recommend to adapt the BAdIs according the given numbering.

We also recommend that you review the provided examples for each BAdI in the *Custom Logic* app.

- [\(1\) Open Channel: Define Implementations \[page 207\]](#)  
This enhancement is **mandatory**. You have to define an *Implementation ID* which represents your open channel action. Once activated, you will see your open channel action in the campaign automation user interface (UI), for example *Open Channel: Letter Export* in the *Add Action* value help.
- [\(2\) Open Channel: Define Parameters for Implementation \[page 208\]](#)  
This enhancement is optional and you can use it to add additional parameters for your open channel action (defined in *(1) Open Channel: Define Implementations*) to the campaign automation UI, such as *Description, Date* or *Priority*.  
Without this implementation your open channel action will offer only the *Export Definition* as action parameter.
- [\(3\) Open Channel: Define Global Settings for Execution \[page 209\]](#)  
This enhancement is optional and you can use it for activating the marketing permission check of open channel actions.  
In addition, you can also check and change, for example, the values of the action parameter or scale the package size that is used during the parallel processing of the campaign execution.
- [\(4\) Open Channel: Enhance Payload for Data Transfer \[page 211\]](#)  
This enhancement is optional and you can use it to check, change and enhance the data of the target group members that shall be transferred.
- [\(5\) Open Channel: Define Template for Outbound Interaction \[page 214\]](#)  
This enhancement is optional and you can use it to write interactions for your target group members. This enables you to contact only target group members with a campaign restart whose have not any interaction.

## Related Information

[Data Flow \[page 233\]](#)

### 4.3.3.2.1.1 (1) Open Channel: Define Implementations

Here you find additional information regarding the Define Implementation enhancement for the open channel integration.

This enhancement is **mandatory** and it should also be the first enhancement that you implement. You must define an *Implementation ID* which represents your open channel action. Once activated, you will see your open channel action in the campaign automation user interface (UI), for example, *Open Channel: Letter Export* in the *Add Action* value help.

In the background, you create with this enhancement the action name and the icon as well as the *Implementation ID*, which is itself required as the filter value for the enhancements (2) to (5) and in the communication arrangements as the property value.

## Changing Parameters

- IMPLEMENTATIONS

Each entry in the table `IMPLEMENTATIONS` represents a campaign action. It will appear in the campaign Designer UI in the menu of the *Add Action* button. An entry contains the following attributes.

- `IMPLEMENTATION` is a character of length 20 containing the technical ID of your implementation. This is also the ID of your action. The ID is used as the filter value in the next enhancements of *Campaign Open Channel*. The ID is mandatory and must start with character Z, such as `ZOC_EXPORT`.
- `IMPLEMENTATION_NAME` is a character of length 40 containing the description of the action.
- `ICON_NAME` is a string containing the URL of the icon that will be shown for the action. If `ICON_NAME` and `ICON_URL` are both given, then the UI will take the value of the `ICON_URL`.
- `COMMUNICATION_MEDIUM` is a character of length 20 containing the technical ID of a communication medium, such as `PAPER` or `PHONE`.

If you activate the check for marketing permission by setting `CHECK_PERMISSION` to `abap_true` in enhancement (3) *Open Channel: Define Global Settings for Execution*, then the value of `COMMUNICATION_MEDIUM` will be taken for the marketing permission checks. To check email-based marketing permissions, the `COMMUNICATION_MEDIUM` should be `EMAIL`.

### 4.3.3.2.1.2 (2) Open Channel: Define Parameters for Implementation

Here you find additional information regarding the Define Parameters for Implementation enhancement for the open channel integration.

This enhancement is **optional** and you can use it to add additional parameters for your open channel action (defined in (1) *Open Channel: Define Implementations* [page 207]) to the campaign automation UI, such as *Description*, *Date*, or *Priority*.

Without this implementation your open channel action will offer only the *Export Definition* as action parameter.

#### i Note

In case you have implemented this enhancement with additional parameters and these parameters shall also appear in the payload, you must also implement (4) *Open Channel: Enhance Payload for Data Transfer* [page 211] and add the parameters in the payload.

The parameters are also **optional**.

## Changing Parameters

- ACTION\_PARAMETERS

The table contains the parameters for the action you have defined within the coding of enhancement (1) [Open Channel: Define Implementations \[page 207\]](#). An entry contains the following attributes:

- ACTION\_PARAMETER is a short string of length 255 containing the technical ID of the action.
- ACTION\_PARAMETER\_NAME is a character of length 40 containing the description of the parameter.
- ACTION\_PARAMETER\_TYPE is a character of length 30 containing the technical type of the parameter.

The following values are allowed:

- if\_cuan\_mkt\_orch\_constants=>action\_param\_type-boolean = 'Edm.Boolean'
- if\_cuan\_mkt\_orch\_constants=>action\_param\_type-date = 'EDM.Date'
- if\_cuan\_mkt\_orch\_constants=>action\_param\_type-time = 'Edm.Time'
- if\_cuan\_mkt\_orch\_constants=>action\_param\_type-string = 'Edm.String'.

When your implementation is called, the table already contains the following data:

ACTION_PARAMETER	ACTION_PARAMETER_NAME	ACTION_PARAMETER_TYPE
EXPORT_DEFINITION	Export Definition	Edm.String

You can add additional action parameters.

If you remove the values for action parameter EXPORT\_DEFINITION, the campaign UI will not offer the export definition as action parameter.

- ACTION\_PARAMETER\_VALUES

The table can be empty or it can be filled with one or more allowed values for an action parameter. An entry contains the following attributes:

- ACTION\_PARAMETER is a short string of length 255 containing the technical ID of the action.
- ACTION\_PARAMETER\_VALUE is a short string of length 255 containing the technical ID of the parameter value.
- ACTION\_PARAMETER\_VALUE\_NAME is a short string of length 255 containing the description of the parameter value.

### 4.3.3.2.1.3 (3) Open Channel: Define Global Settings for Execution

Here you find additional information regarding the Define Global Settings for Execution enhancement for the open channel integration.

This enhancement is **optional** and you can use it for activating the marketing permission check of open channel actions.

#### i Note

However, keep in mind that the checkbox *Ignore Marketing Permission* in the campaign automation user interface under *Marketing Information* skips the marketing permission checks for open channel actions as well.

In addition, you can also check and change, for example, the values of the action parameter or scale the package size that is used during the parallel processing of the campaign execution.

For more information, see [Activating Marketing Permissions \[page 215\]](#).

## Changing Parameters

- `CHECK_PERMISSION` is a boolean controlling if marketing permission shall be checked. Set it to value `abap_true` if you want that the campaign execution checks the marketing permission for every target group member. The check can be executed only if you have set a `COMMUNICATION_MEDIUM` in the implementation of enhancement [\(1\) Open Channel: Define Implementations \[page 207\]](#). If the target group member has the permission, it will be processed in further steps of the execution. If the target group member has no permission, the campaign execution will create a corresponding interaction to log the missing permission.
- `PACKAGE_SIZE` is an integer of length 10 containing the size of a package. The campaign execution transfers the data in parallel processed packages. With this parameter you can define the number of target group members processed in one package. The parameter contains the default value 50 which can be changed in your implementation.

- `HEADER_ATTRIBUTES`

The table contains the name and value of header attributes (for example the campaign ID). An entry contains the following attributes:

- `PARAM_NAME` is a short string of length 255 containing the name of a parameter.
- `PARAM_VALUE` is a short string of length 255 containing the value of a parameter.

The table `HEADER_ATTRIBUTES` initially contains the following data which can be used in your implementation:

PARAM_NAME	PARAM_VALUE
CAMPAIGN	<ID of the current campaign with leading zeros; you see the ID in the campaign UI>
EXECUTION_RUN_KEY	<GUID of the execution run; acts as session ID and is a unique ID of every execution of the action, every execution has its own run ID (if action is executed periodically)>
SEGMENTATION_OBJECT	<ID of the segmentation object>
OC_IMPLEMENTATION_ID	<ID of your implementation, defined in enhancement <b>(1) Open Channel: Define Implementations</b> ; for example, ZOC_EXPORT>
EXPORT_DEFINITION	<ID of the export definition (if you did not remove it as action parameter in the enhancement <b>(2) Open Channel: Define Parameters for Implementation</b> )>

This table also contains the action parameters you have defined in the enhancement [\(2\) Open Channel: Define Parameters for Implementation \[page 208\]](#).

- `ERROR_OCCURED` is a character of length 1 indicating if an error occurred and controlling that campaign execution will stop. Set it to `abap_true` if your implementation wants to indicate an error and wants to stop the complete campaign execution.
- `ERROR_MESSAGES` is a table of messages (each message has 200 characters length) that will appear in the log of the campaign execution.

### 4.3.3.2.1.4 (4) Open Channel: Enhance Payload for Data Transfer

Here you find additional information regarding the Enhance Payload for Data Transfer enhancement for the open channel integration.

This enhancement is **optional** and you can use it to check, change and enhance the data of the target group members that shall be transferred.

#### Importing Parameters

- `HEADER_ATTRIBUTES`  
The table contains the name and value of header attributes (for example the campaign ID). Its content could have been changed by your implementation of enhancement [\(3\) Open Channel: Define Global Settings for Execution \[page 209\]](#). An entry contains the following attributes:
  - `PARAM_NAME` is a short string of length 255 containing the name of a parameter.
  - `PARAM_VALUE` is a short string of length 255 containing the value of a parameter.

#### Changing Parameters

- `TARGET_GROUP_MEMBER_ATTRIBUTES`  
The table contains the meta-data of the transferred data. This is the information of fields which have been defined in the export definition. The export definition will be assigned by the marketing expert, if this user adds your open channel implementation as an action to the campaign. In addition, you can add your own (additional) attributes. In such a case, you should also fill such an attribute with values in table `TARGET_GROUP_MEMBER_DATA`. An entry contains the following attributes:
  - `ATTRIBUTE_ID` is a short string of length 255 containing the ID of an attribute defined in the export definition. It has the following format: `DA-<ID of Data Source>-<ID of Attribute>`. For example, `DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-NAME_FIRST` First Name Edm.String. The data source ID is shown in the UI of Export Definition.
  - `ATTRIBUTE_NAME` is a character of length 40 containing the description of an attribute defined in the export definition. The attribute name is shown in the UI of Export Definition.
  - `ATTRIBUTE_TYPE` is a character of length 30 containing the technical type of an attribute defined in the export definition. Types are EDM types like Edm.Boolean, Edm.Date, Edm.Time, Edm.String etc.

Here is an example of content of table `TARGET_GROUP_MEMBER_ATTRIBUTES`:

ATTRIBUTE_ID	ATTRIBUTE_NAME	ATTRIBUTE_TYPE
DA-SAP_CE_CON-TACT_IA_ERP_CUSTOMER-NAME_FIRST	First Name	Edm.String
DA-SAP_CE_CON-TACT_IA_ERP_CUSTOMER-NAME_LAST	Last Name	Edm.String
DA-SAP_CE_CON-TACT_IA_ERP_CUSTOMER-COUNTRY	Country	Edm.String
DA-SAP_CE_CON-TACT_IA_ERP_CUSTOMER-CITY	City	Edm.String

- `TARGET_GROUP_MEMBER_DATA`

The table contains the data of the target group members that shall be transferred. An entry contains the following attributes:

- `TG_MEMBER_KEY` is a binary GUID containing the technical key of a target group member. Example: 12345678901234567890123456789012.
- `TG_MEMBER_INTERACTION_CONTACT` is a binary GUID containing the technical key of the interaction contact which is member of the target group. Example: 09876543210987654321098765432109; representing the interaction contact: William Smith, Springfield, CA, US.
- `TG_MEMBER_INTERACTION` is a binary GUID containing the technical key of the interaction which is member of the target group. This attribute is only filled if your action is used in a follow-up trigger of campaign automation, such as Trigger Click Through.
- `ATTRIBUTE_ID` is a short string of length 255 containing the technical ID of an attribute defined in the export definition. Example: `NAME_FIRST`.
- `ATTRIBUTE_VALUE` is a short string of length 255 containing the value of the attribute for the target group member. Example: William.

Here is an example of content of table `TARGET_GROUP_MEMBER_DATA`:

TG_MEMBER_KEY	TG_MEMBER_INTERACTION_CONTACT	TG_MEMBER_INTERACTION	ATTRIBUTE_ID	ATTRIBUTE_VALUE
12345678901234567890123456789012	09876543210987654321098765432109	000000000000000000000000	NAME_FIRST	William
8901234567890123456789012	43210987654321098765432109	000000000000000000000000		
12345678901234567890123456789012	09876543210987654321098765432109	000000000000000000000000	NAME_LAST	Smith
8901234567890123456789012	43210987654321098765432109	000000000000000000000000		



TG_MEMBER_KEY	TG_MEMBER_IN- TERACTION_CON- TACT	TG_MEMBER_IN- TERACTION	ATTRIBUTE_ID	ATTRIBUTE_VALUE
2345678901234567 8901234567890123	9876543210987654 3210987654321098	00000000000000 00000000000000 0000	NAME_FIRST	Adam
2345678901234567 8901234567890123	9876543210987654 3210987654321098	00000000000000 00000000000000 0000	NAME_LAST	Miller

- TARGET\_GROUP\_MEMBER\_STATUS**  
 The table allows you to control if one or several target group members shall not be transferred. When a target group member shall not be transferred, set an `INTERACTION_TYPE` and a corresponding `FAILURE_REASON` in order to remove the target group member from the data transfer. Campaign execution will take given interaction type and failure reason to write a corresponding interaction. All entries of this table having an empty interaction type will be treated as successful and transferred. An entry of table `TARGET_GROUP_MEMBER_STATUS` contains the following attributes:
  - `TG_MEMBER_KEY` is a binary GUID containing the technical key of a target group member.
  - `TG_MEMBER_INTERACTION_CONTACT` is a binary GUID containing the technical key of the interaction contact which is member of the target group.
  - `TG_MEMBER_INTERACTION` is a binary GUID containing the technical key of the interaction which is member of the target group. This attribute is only filled if your action is used in a follow-up trigger of campaign automation, such as Trigger Click Through.
  - `FAILURE_REASON` is a character of length 20 containing the ID of an interaction reason (can be defined in configuration/customizing). It corresponds to the field `IA_REASON` of the interaction table. Example: `NAME_MISSING`.
  - `INTERACTION_TYPE` is a character of length 20 containing the technical ID of an interaction type. (can be defined in configuration/customizing). It corresponds to the field `IA_TYPE` of the interaction table. Example: `OUTBOUND_FAILED`.
  - `OUTBOUND_INTERACTION` is a binary GUID containing the key of the interaction that will be written for this target group member. It corresponds to the key field of the interaction table. You can use this unique identifier and transfer it to use it as transactional identifier per target group member. For example: on HCI you can take the `OUTBOUND_INTERACTION` as identifier for a "preceding document" if you create a new document via HCI.
  - `COMMUNICATION_ID` is a short string of length 255 initially containing the GUID of the Interaction Contact. Its value is identical to the value of `TG_MEMBER_INTERACTION_CONTACT`. It corresponds to the field `COMM_ID` of the interaction table. If you activate the check for marketing permission by setting `CHECK_PERMISSION` to `abap_true` in enhancement **(3) Open Channel: Define Global Settings for Execution**, set the value of `COMMUNICATION_ID` to the value to be checked. For example, to check email-based marketing permissions, the `COMMUNICATION_ID` should contain an email address.
  - `CONTACT_PROJ_KEY` is the key of a Contact-to-Account Relationships. If not empty, that means that the target group member is a relationship.
- `ERROR_OCCURED` is a character of length 1 indicating if an error occurred and controlling that campaign execution will stop. If `ERROR_OCCURED` is set to `abap_true`, the campaign execution will not write any interactions for this package of target group members. Because campaign execution processes the

amount of target group members in packages, it can happen that other packages are processed without errors.

- `ERROR_MESSAGES` is a table of messages (each message has 200 characters length) that will appear in the log of the campaign execution.

### 4.3.3.2.1.5 (5) Open Channel: Define Template for Outbound Interaction

Here you find additional information regarding the Define Template for Outbound Interaction enhancement for the open channel integration.

This enhancement is **optional** and you can use it to write interactions for your target group members. This enables you to contact only target group members with a campaign restart who have not had any interaction.

#### i Note

Note that the enhancements [\(1\) Open Channel: Define Implementations \[page 207\]](#) and [\(2\) Open Channel: Define Parameters for Implementation \[page 208\]](#) are required to enhance the campaign user interface (UI) with the *Open Channel* action.

- You can find the technical details and examples in the example coding of the enhancements.
- If you want to use more than one *Implementation ID*, which means you want to have more than one open channel action, then:
  - you have to define all of them in one implementation of enhancement [\(1\) Open Channel: Define Implementations](#).
  - you have to separate implementations for the other enhancements (2-5) and use the *Implementation ID* as filter.

## Importing Parameters

- `HEADER_ATTRIBUTES`  
The table contains the name and value of header attributes (for example the campaign ID). Its content could have been changed by your implementation of enhancement [\(3\) Open Channel: Define Global Settings for Execution \[page 209\]](#). An entry contains the following attributes:
  - `PARAM_NAME` is a short string of length 255 containing the name of a parameter.
  - `PARAM_VALUE` is a short string of length 255 containing the value of a parameter.

## Changing Parameters

- `TEMPLATE`  
The structure is the template for the interactions that will be created. It contains the following attributes:
  - `ID_ORIGIN` is a character of length 20 containing the technical ID of an origin (can be defined in configuration/customizing). When permission check is active, enter the same origin ID (`ID_ORIGIN`)

as used for retrieving the communication identifier in the enhancement [\(4\) Open Channel: Enhance Payload for Data Transfer \[page 211\]](#). Both interaction attributes `INTERACTION_TYPE` and `COMMUNICATION_MEDIUM` are used for interaction creation of all contacts that passed the permission check successfully.

- `INTERACTION_TYPE` is a character of length 20 containing the technical ID of an interaction type (can be defined in configuration/customizing).
- `COMMUNICATION_MEDIUM` is a character of length 20 containing the technical ID of a communication medium (can be defined in configuration/customizing).

### 4.3.3.2.2 Activating Marketing Permissions

With the following adaptations in the example coding of the enhancements you can activate marketing permissions for your open channel integration.

#### i Note

We recommend that you review the provided examples for each enhancement in the [Custom Logic](#) app.

- [\(1\) Open Channel: Define Implementations](#)

You set the communication medium to run the permission checks. The following example checks email-based marketing permissions.

#### Sample Code

```
APPEND VALUE #( implementation = 'ZOC_MKTG_PRSSN'
                  implementation_name = 'Check Marketing
Permission (Email)'
                  icon_name = 'email'
                  icon_url = ''
                  communication_medium = 'EMAIL' ) TO
implementations.
```

- [\(2\) Open Channel: Define Parameters for Implementation](#)

No code adaptations necessary.

- [\(3\) Open Channel: Define Global Settings for Execution](#)

Activate the marketing permission checks.

#### Sample Code

```
check_permission = abap_true.
```

- [\(4\) Open Channel: Enhance Payload for Data Transfer](#)

Determine the communication identifier for running the permission checks. The communication identifier depends on the communication medium. In the example code, the email address is taken from the contact details under [Email](#) as communication identifier.

#### i Note

To see a sample code, please refer to the BAdI documentation in the [Custom Logic](#) app.

- [\(5\) Open Channel: Define Template for Outbound Interaction](#)

Set the interaction attributes and the origin ID (contact). Enter the same origin ID (ID\_ORIGIN) as used for retrieving the communication identifier in the enhancement [Enhance Payload for Data Transfer](#). Both interaction attributes INTERACTION\_TYPE and COMMUNICATION\_MEDIUM are used for interaction creation of all contacts that passed the permission check successfully.

#### Sample Code

```
template-id_origin          = 'EMAIL'.
template-interaction_type   = 'Z_OPEN_CHANNEL'. " Example for a customer
defined interaction type.
template-communication_medium = 'BUSINESS_DOCUMENT'.
```

#### Note

The implementation of this enhancement is **mandatory** for creating any kind of interactions such as contact has been rejected by permission checks, and for processing inbound interactions. But it is **optional** for the outbound interactions.

#### Example

The following table shows the written interactions in your system: the first row shows an entry of a refused permission whereas the entries of the second and third row are granted permissions.

ID_ORIGIN	ID	COMMUNICATION_MEDIUM	INTERACTION_TYPE
EMAIL	j.armstrong@example.com	BUSINESS_DOCUMENT	OUTBOUND_CHCK_FAILED
EMAIL	j.miller@example.com	BUSINESS_DOCUMENT	Z_OPEN_CHANNEL
EMAIL	m.adams@example.com	BUSINESS_DOCUMENT	Z_OPEN_CHANNEL

### 4.3.3.2.3 Get Deviating Communication ID Based on Origin ID

You can get deviating communication IDs based on origin IDs, doing some coding in enhancements. You need this switch, in case you have activated the enhancement [\(5\) Open Channel: Define Template for Outbound Interaction](#). If the communication ID does not fit to the origin ID, the system cannot create interactions.

#### Example

##### Example 1

You have done an example implementation for the enhancement [\(4\) Open Channel: Enhance Payload for Data Transfer](#) as described in [Activating Marketing Permissions \[page 215\]](#).

In the enhancement [\(5\) Open Channel: Define Template for Outbound Interaction](#), the `template-id_origin` is **EMAIL**.

## Example 2

In the enhancement (5) *Open Channel: Define Template for Outbound Interaction*, the `template-id_origin` is **SAP\_C4C\_BUPA**.

For this origin ID you need the following example coding in the enhancement (4) *Open Channel: Enhance Payload for Data Transfer*:

### Sample Code

```
LOOP AT target_group_member_status ASSIGNING FIELD-  
SYMBOL(<ls_target_group_member_stat>) WHERE tg_member_interaction_contact  
IS NOT INITIAL.  
  CL_CUAN_INTERACT_CNTCT_HELPER=>GET_CONTACT_FACETS( EXPORTING  
IT_CONTACT_KEYS = value #( ( conv #( <ls_target_group_member_stat>-  
tg_member_interaction_contact ) ) )  
IMPORTING  
ET_CONTACT_FACET = data(lt_contact_facet) ).  
  READ TABLE lt_contact_facet ASSIGNING FIELD-SYMBOL(<ls_contact_facet>)  
WITH KEY id_origin = 'SAP_C4C_BUPA'.  
  IF SY-SUBRC EQ 0.  
    <ls_target_group_member_stat>-communication_id = <ls_contact_facet>-id.  
  ENDIF.
```

## 4.3.3.3 Setting Up SAP Business Technology Platform

In the following you find some further information about an integration example using SAP Business Technology Platform.

Keep the following hints and recommendations in mind when you are using SAP BTP:

- HTTP status code shall be **200 (OK)** and **201 (Created)** for a successful processing in SAP BTP. Other HTTP status codes of the group *Success* will lead to a warning with a lower priority in the application log. That means it is not visible in the execution log of the campaign user interface (UI).
- Structure of the export definition and implementation of the enhancement (4) *Open Channel: Enhance Payload for Data Transfer* defines the content of the OData entity `TargetGroupMemberAttributeData`.
- The type of property `Value` of the OData entity `TargetGroupMemberAttributeData` is always `Edm.String` and the property `EdmTypeId` defines the type for formatting property `Value`.

Example for the formatting:

### Sample Code

```
Edm.String: "Value": "Jane", "Value": "0002", "Value":  
"047D7B8BFC411EE596DA0E15129A2367",  
Edm.Date: "Value": "2016-09-23",  
Edm.Guid: "Value": "8cdcd4a8-4768-1ed6-87ca-79a6c5fdf291",  
Edm.Int16: "Value": "-255"  
Edm.Boolean: "Value": "false", "Value": "true",  
Edm.Double: "Value": "-2345.66",
```

## Related Information

[OData Service Settings for Outbound \[page 220\]](#)

### 4.3.3.4 Create Communication Systems and Arrangements

After you have set up your enhancements and the SAP Business Technology Platform you have to create a communication system. With the communication system and the communication arrangements you create your *Implementation IDs*.

#### Prerequisite

To set up a communication system and communication arrangement, you require the business catalog role Communication Management (SAP\_CORE\_BC\_COM).

You require HTTPS 1.1 for your HTTP requests.

#### Communication System

1. Choose the *Communication Systems* app and then *New*.
2. In the popup enter a system ID, such as **Z\_HCI\_CPG\_OPEN\_CHANNEL**, and system name. You can freely define a name; but note that the name is used when you create the communication arrangement. Then choose *Create*.
3. Under *Technical Data* enter the SAP BTP instance you want to use for the campaign execution as *Host Name*. Enter only the pure host name without any path and port. Note that *Log System ID*, *Client Name*, and *Business System* are not relevant for the campaign execution.
4. Optionally, you can provide your contact information.
5. Under *User for Outbound Communication*, choose *Add (+)* to add a set of access details for the external server. Select **SSL Client Certificate** as *Authentication Method* and **Default Client Certificate** as *Certificate Type*. To finish choose *Create*.
6. Choose *Save* to save the new or edited communication system in an active status.

#### Communication User

##### → Recommendation

We recommend to use certificates instead of communication users.

But in case you are using your own communication users, please take care that the communication user is not longer than 32 characters.

## Communication Arrangement

You set up a communication arrangement to enable the campaign execution.

1. Choose *Communication Arrangement* app and then *New*.
2. In the *New Communication Arrangement* dialog use the predefined scenario **Marketing - Campaign Open Channel Integration (SAP\_COM\_0049)** from the value help of the *Scenario* define an *Arrangement Name* using the following pattern: **SAP\_COM\_0049\_<Implementation\_ID>**, for example, **SAP\_COM\_0049\_ZOC\_EXPORT**.
3. Under *Common Data*, select a *Communication System* from the value help that you have created in the *Communication System* app. Note that *My System* is filled automatically. You will need this entry later in your mapping script in SAP BTP.
4. Under *Additional Properties*, select the *Implementation ID* of your enhancement from the value help.

### i Note

Note that **-as a mandatory prerequisite-** you have to define the *Implementation ID* in the mandatory enhancement (1) *Open Channel: Define Implementations*. For more information, see [Adapt Enhancements \[page 206\]](#) and [BAdI Details \[page 207\]](#).

5. Optionally you can set *Retry Send Active* to **True (X)** which enables your system to resend the requests in case of fails. For more information, see [Retry Sending Using Idempotency \[page 234\]](#).
6. Under *Outbound Communication* all required fields are filled automatically from the selected *Communication System* above.
7. Under *Outbound Services*, enter the paths for the predefined outbound services using the following pattern: **/<your\_service>/<your\_entity>** and the corresponding *Service URL* for the following steps of the open channel action:
  - o *Preprocessing* (optional)
  - o *Processing* (mandatory)
  - o *Postprocessing* (optional)

You can find the required data in the iFlow of the SAP BTP system that you want to connect.

### i Note

For each outbound service you must set the *Service Status* to *Active* at least for the outbound service *Processing*.

8. Choose *Save* to save the new or edited communication arrangement in an active status.
9. Now you can download the certificate to your local machine and upload it to use it in SAP BTP.
10. After you have entered the required settings and uploaded the certificate in SAP BTP, you can proof your connection by choosing *Check Connection* under *Outbound Services*.

## Related Information

[Data Flow \[page 233\]](#)

### 4.3.3.5 OData Service Settings for Outbound

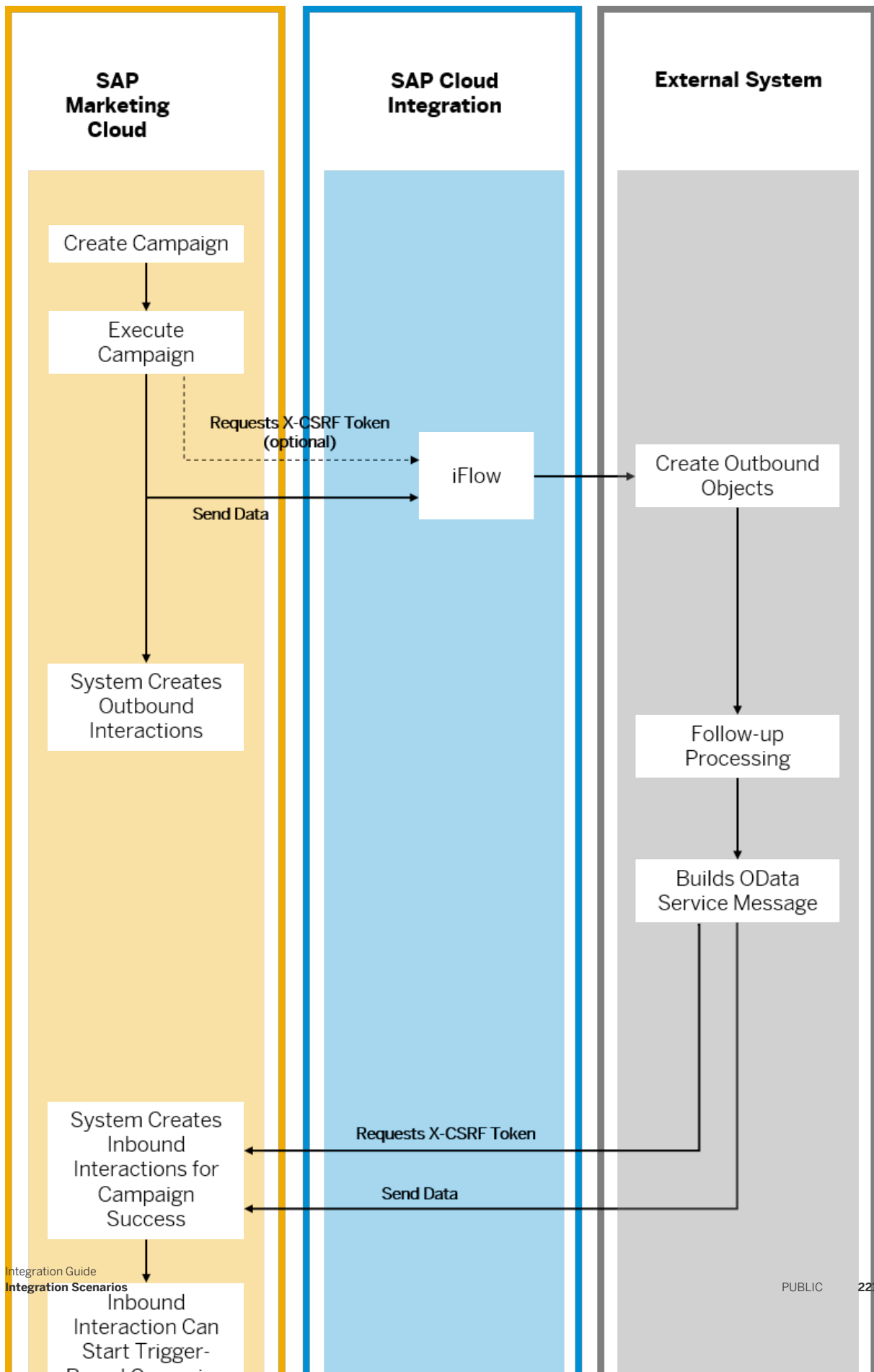
Here you can find more information about the settings in the OData service `CUAN_CAMPAIGN_OPEN_CHANNEL` that are required for the outbound of an open channel integration.

#### Prerequisite

You have implemented the enhancement [\(5\) Open Channel: Define Template for Outbound Interaction](#) in addition to the mandatory enhancement [\(1\) Open Channel: Define Implementations](#).



## About the OData



From the communication point of view the system for SAP Marketing Cloud acts as a client and the server side implementation of the services is done in external systems using SAP Business Technology Platform as a middleware.

Please keep the following things in mind:

- Communication format for requests and responses shall be formatted in JSON only and not in Atom.
- Data protocol of the communication is (almost) OData V2.
- Only code and message object of the OData V2 error response is saved, but not any deeper errors of the OData error document.
- Each processing step requires an active outbound node in the communication arrangement, customers could drop processing of the Pre- and Post-Processing steps by leaving the active flag empty, but data transfer for the processing step is mandatory.
- **Cross-Site Request Forgery (CSRF) Protection**
  - (SAP BTP) HTTP sender channel supports the cross-site request forgery (CSRF) protection.
  - When CSRF protection is activated, it is used in the preprocessing, processing, and postprocessing step. Though the communication with SAP BTP is CSRF protected.
  - We recommend to activate CSRF protection for each integration flow separately.
  - Integration flow must be capable to respond on HTTP HEAD (check communication in communication arrangements, CSRF Protection) and POST (open channel outbound) requests.
- **Restarting a Campaign**
  - In case the an existing campaign has written outbound interactions and you restart your campaign only those members of the used target group are contacted again for whom no outbound interaction exist.
  - In case the an existing campaign has **not** written outbound interactions and you restart your campaign, the campaign sends out emails to all members of the target group again. Therefore we recommend to check your iFlow to prevent duplicates.

OData Entity Types

Entity Type	Description
Campaign	An entity representing the campaign business object.
CampaignExecutionRun	An entity representing a campaign execution run. Each send process belongs to one execution run.
CampaignExecutionRunPackage	An entity representing a package that is processed by an execution run.
CampaignTargetGroupMember	An entity representing a target group member.
TargetGroupMemberAttributeDatum	An entity representing the attributes and attributes values of a target group member.  Note that the entity set is named TargetGroupMemberAttributeData.

The names of the entity sets are the plural form of the entities.

## Entity Properties

The names of the properties are almost the same names as used for OData services located in the package CUAN\_COMMON.

### Campaign Properties

Property	Type	Description
CampaignId	String	An identifier of a campaign key of the entity type.
Name	String	A name of a campaign.
MarketingAreaId	String	An identifier of an marketing area.
SegmentationObject	String	An identifier of a segmentation object.
ImplementationId	String	An identifier of an customer-specific action implementation.

### Campaign Execution Run Properties

Property	Type	Description
ExecutionRunKey	String	An globally unique identifier of an execution run key of the entity type.
ProcessingStepCode	String	A coded representation of the processing step.
CampaignId	String	An identifier of the campaign business object, used as reference to the parent entity type Campaign.

### Campaign Execution Run Package Properties

Property	Type	Description
PackageId	String	An identifier of the package key of the entity type.
Campaign	String	A complex data type representing the entity type Campaign.
ExecutionStartDateTime	String	A timestamp at which the execution run is started.

Property	Type	Description
ExecutionRunKey	String	An globally unique identifier of an execution run, used as reference to the parent entity type CampaignExecutionRun key of the entity type.

#### Campaign Target Group Member Properties

Property	Type	Description
OutboundId	String	A globally unique identifier of the outbound interaction key of the entity type.
PackageId	String	An identifier of the package, used as reference to the parent entity type CampaignExecutionRunPackage.
ExecutionRunKey	String	An globally unique identifier of an execution run, used as reference to the parent entity type CampaignExecutionRunPackage.

For more information about the interaction key, see [Interactions \[page 615\]](#).

#### Campaign Target Group Member Attribute Datum

Property	Type	Description
AttributeId	String	An identifier of an attribute key of the entity type.
Value	String	A value of the attribute.
EdmTypeId	String	An identifier of the OData type.
OutboundId	String	A globally unique identifier of the outbound interaction, used as reference to the parent entity type CampaignTargetGroupMember.

### **i** Note

Structure of the export definition and implementation of the enhancement *(4) Open Channel: Enhance Payload for Data Transfer* defines the content of the OData entity `TargetGroupMemberAttributeData`.

## Preprocessing

One deep create message consisting of the entity sets Campaigns and CampaignExecutionRuns is send.

Example Payload:

### Sample Code

```
{
  "d": {
    "CampaignId": "0000381379",
    "Name": "Open Channel Demo 1",
    "MarketingAreaId": "CXXGLOBAL",
    "SegmentationObject": "SAP_CONTACT_ENGAGEMENT_SIN",
    "ImplementationId": "ZOC_EXPORT",
    "CampaignExecutionRuns": {
      "ExecutionRunKey": "8CD4A847681EE69182D1BBA1C39EF3",
      "ProcessingStepCode": "1",
      "CampaignId": "0000381379"
    }
  }
}
```

## Processing

Multiple deep create messages consisting of the entity sets CampaignExecutionRunPackages, CampaignTargetGroupMembers and TargetGroupMemberAttributeData are send. Each message belongs to one execution package.

### Sample Code

```
{
  "d": {
    "Campaign": {
      "CampaignId": "0000381379",
      "Name": "Open Channel Demo 1",
      "MarketingAreaId": "CXXGLOBAL",
      "SegmentationObject": "SAP_CONTACT_ENGAGEMENT_SIN",
      "ImplementationId": "ZOC_EXPORT"
    },
    "PackageId": 1,
    "ExecutionStartDateTime": "2016-07-07T07:44:40Z",
    "ExecutionRunKey": "8CD4A847681EE69182D1BBA1C39EF3",
    "CampaignTargetGroupMembers": [
      {
        "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10",
        "PackageId": 1,
        "ExecutionRunKey": "8CD4A847681EE69182D1BBA1C39EF3",
        "TargetGroupMemberAttributeData": [
          {
            "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-NAME_FIRST",
            "Value": "Julie",
            "EdmTypeId": "Edm.String",
            "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
          }, {
            "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-NAME_LAST",
            "Value": "Armstrong",
            "EdmTypeId": "Edm.String",
            "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
          }
        ]
      }
    ]
  }
}
```

```

CONTACT_KEY",
    "EdmTypeId": "Edm.String",
    "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
  }, {
    "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
CONTACT_KEY",
    "Value": "005056AC4A181ED598D20A84AB8AC6E9",
    "EdmTypeId": "Edm.Binary",
    "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
  }, {
    "AttributeId": "OUTBOUND_INTERACTION",
    "Value": "8CDCD4A847681EE69182D4A1498E1EF5",
    "EdmTypeId": "Edm.String",
    "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
  }, {
    "AttributeId": "ZOC_EXPORT_DESCRIPTION",
    "Value": "Open Channel Demo 1",
    "EdmTypeId": "Edm.String",
    "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
  }, {
    "AttributeId": "ZOC_EXPORT_DATE",
    "Value": "2016-07-07",
    "EdmTypeId": "Edm.Date",
    "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
  }, {
    "AttributeId": "ZOC_EXPORT_PRIORITY",
    "Value": "PRIORITY_1",
    "EdmTypeId": "Edm.String",
    "OutboundId": "1202654B21C72A50A0D4E5CB35E6D2FBE916EA10"
  }
}
},
{"OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23",
"PackageId": 1,
"ExecutionRunKey": "8CDCD4A847681EE69182D1BBA1C39EF3",
"TargetGroupMemberAttributeData": [
{
"AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_FIRST",
"Value": "John",
"EdmTypeId": "Edm.String",
"OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23"
}, {
"AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_LAST",
"Value": "Miller",
"EdmTypeId": "Edm.String",
"OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23"
}, {
"AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
CONTACT_KEY",
"Value": "005056AC4A181ED598D20A84AB8B06E9",
"EdmTypeId": "Edm.Binary",
"OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23"
}, {
"AttributeId": "OUTBOUND_INTERACTION",
"Value": "8CDCD4A847681EE69182D4A1498EDEF5",
"EdmTypeId": "Edm.String",
"OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23"
}, {
"AttributeId": "ZOC_EXPORT_DESCRIPTION",
"Value": "Open Channel Demo 1",
"EdmTypeId": "Edm.String",
"OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23"
}, {
"AttributeId": "ZOC_EXPORT_DATE",
"Value": "2016-07-07",
"EdmTypeId": "Edm.Date",
"OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23"
}
}
}
}

```

```

    }, {
      "AttributeId": "ZOC_EXPORT_PRIORITY",
      "Value": "PRIORITY_1",
      "EdmTypeId": "Edm.String",
      "OutboundId": "EA297547B0DBBDF81C308FD14A3757C1420ABB23"
    }
  ],
  {
    "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF",
    "PackageId": 1,
    "ExecutionRunKey": "8CDCD4A847681EE69182D1BBA1C39EF3",
    "TargetGroupMemberAttributeData": [
      {
        "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_FIRST",
        "Value": "Michael",
        "EdmTypeId": "Edm.String",
        "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF"
      }, {
        "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
NAME_LAST",
        "Value": "Adams",
        "EdmTypeId": "Edm.String",
        "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF"
      }, {
        "AttributeId": "DA-SAP_CE_CONTACT_IA_ERP_CUSTOMER-
CONTACT_KEY",
        "Value": "40F2E93065BD1ED598D1DCFD65F97C0",
        "EdmTypeId": "Edm.Binary",
        "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF"
      }, {
        "AttributeId": "OUTBOUND_INTERACTION",
        "Value": "8CDCD4A847681EE69182D4A1498F3EF5",
        "EdmTypeId": "Edm.String",
        "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF"
      }, {
        "AttributeId": "ZOC_EXPORT_DESCRIPTION",
        "Value": "Open Channel Demo 1",
        "EdmTypeId": "Edm.String",
        "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF"
      }, {
        "AttributeId": "ZOC_EXPORT_DATE",
        "Value": "2016-07-07",
        "EdmTypeId": "Edm.Date",
        "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF"
      }, {
        "AttributeId": "ZOC_EXPORT_PRIORITY",
        "Value": "PRIORITY_1",
        "EdmTypeId": "Edm.String",
        "OutboundId": "CA5F1FE120237480E6054B06D61371081AE095DF"
      }
    ]
  }
]
}
}
}

```

## Postprocessing

One message consisting of the entity set "Campaign Execution Runs" is send.

Example Payload:

### Sample Code

```
{
  "d": {
    "ExecutionRunKey": "8CD4A847681EE69182D1BBA1C39EF3",
    "ProcessingStepCode": "3",
    "CampaignId": "0000381379"
  }
}
```

## Message Choreography Between the OData and SAP Cloud Platform

Notes regarding the message choreography:

- The system for SAP Marketing Cloud saves all error messages in the application log, and also the information with lower severity.
- The system for SAP Marketing Cloud does not require responses that are sent after creation of entities (status code 201).
- The calls are synchronous calls for data transfer between SAP Marketing Cloud and SAP BTP.
- The receiving system should process the data asynchronously to get a better error handling and performance.

### Error Handling

In the following you will find useful remarks about the error handling of this OData service:

- Errors in the *Preprocessing* phase stops all further processing steps.
- Errors during the *Processing* step marks the entire package as erroneous.
- Business errors in the SAP BTP mapping shouldn't occur. We recommend to implement all checks in the enhancements because SAP BTP rejects the entire package.
- We recommend to set an appropriate HTTP status code and send an OData V2 error document formatted as JSON in case of errors.
- Content of `code` and `value` are saved in the application log as error and will be visible in the campaign UI execution log. But note that the content of the inner-error-node is not parsed.
- Content of all error messages not sent as OData V2 error document is also saved in the application log as it is.
- Content of the HTTP responses of HTTP status codes of the group `Success` is not saved in the application log.
- All HTTP status code greater or equal than 300 mark the actual execution package as erroneous and stop the execution of the following packages. All erroneous packages and not processed packages can be restarted again.

### OData V2 Error Document Example (Inner Error Node Is Not Shown)

### Sample Code

```
{
  "error": {
    "code": "CUAN_MKT_ORCH_ODATA/001",
    "message": {
      "lang": "en",
```



```

      "value": "Determination of the key failed for
campaign ID \"0000032784\" version \"1\"."
    }
  }
}

```

## Related Information

[Open Channel Integration \[page 194\]](#)

### 4.3.3.6 Inbound Service Settings API

Here you can find more information about the settings in the API that are required for the inbound of an open channel integration.

The inbound comprises the creation of interactions (only creation) by using the deep-inserts of the OData service `API_MKT_INTERACTION`.

Endpoint OData Service Deep Insert: `/sap/opu/odata/sap/API_MKT_INTERACTION_SRV/InteractionsDeepInsert`

You need this path to create the service URL to your external system. In addition it tells you how the `InteractionsDeepInsert` data is structured.

Entity Property of OData Service	Mandatory (must be filled by the OData service)	Copy (data is taken over from the outbound)	Remarks
Entity <code>InteractionsDeepInsert</code> (This data is part of the interaction API.)			
UUID			
Child Entity <code>baseInteraction</code> (This data is part of the import data.)			
<code>InteractionUUID</code>	X	X	There will be an error in case the inbound and outbound data of this property is not identical.
<code>InteractionContactOrigin</code>		X	There will be an error in case the inbound and outbound data of this property is not identical.

Entity Property of OData Service	Mandatory (must be filled by the OData service)	Copy (data is taken over from the outbound)	Remarks
InteractionContactId		X	There will be an error in case the inbound and outbound data of this property is not identical.
CommunicationMedium		X	There will be an error in case the inbound and outbound data of this property is not identical.
InteractionType	X		<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>i Note</b></p> <p>Keep in mind that the <i>Direction</i> for the interaction type must be In-bound in the self-service configuration app Manage Interaction Content. For more information, see <a href="#">Managing Interaction Content</a>.</p> </div>
InteractionSourceObjectType	X		Passes always CUAN_CAMPAIGN_OUTBOUND
InteractionSourceObject	X		<p>Passes property OutboundId of the entity CampaignTargetGroupMember.</p> <p>There will be an error in case the SourceObjectId is empty.</p>
InteractionTimeStampUTC	X		
SourceSystemType	X		Both properties
SourceSystemId			SourceSystemType and SourceSystemId are optional, but helpful values and therefore should be passed for describing the origin of the interaction.

Entity Property of OData Service	Mandatory (must be filled by the OData service)	Copy (data is taken over from the outbound)	Remarks
CampaignID		X	There will be an error in case the inbound and outbound data of this property is not identical.
CampaignContent		X	There will be an error in case the inbound and outbound data of this property is not identical.
InteractionAdditionalObject		X	There will be an error in case the inbound and outbound data of this property is not identical.
InteractionIsAnonymous			There will be an error in case IsAnonymous has the value <b>X</b> .
Child Entity InteractionAdditionalObject (optional, child of Interaction)			
Child Entity InteractionProduct (optional, child of Interaction)			
Child Entity InteractionInterest (optional, child of Interaction)			

The inbound processing retrieves all outbound interactions of the campaign by using the `SourceObjectId` and takes over data (**X** in column **Copy**) from the outbound interaction.

The OData entity `InteractionAdditionalObject` is always be copied from the outbound interaction.

The following example coding shows the minimum payload required:

#### Sample Code

```
{
  "UUID": "32575914-a9db-476c-a51a-2b0d4a899b95",
  "Interactions": [
    {
      "InteractionUUID":
"00000000-0000-0000-0000-000000000000",
      "InteractionType": "ZOC_CALL_CENTER_INB",
      "InteractionSourceObjectType":
"CUAN_CAMPAIGN_OUTBOUND",
      "InteractionSourceObject":
"18D3620CC1DBAEB8E5F97AFB922E84E092F271F0",
      "InteractionTimeStampUTC":
"2019-06-13T08:55:00"
    }, {
      "InteractionUUID":
"00000000-0000-0000-0000-000000000000",
      "InteractionType": "ZOC_CALL_CENTER_INB",
      "InteractionSourceObjectType":
"CUAN_CAMPAIGN_OUTBOUND",
```

```

        "InteractionSourceObject":
        "0FB6D7D9DCDB8616A4A39D0E931C01DA57B5E48F",
        "InteractionTimeStampUTC":
        "2019-06-13T08:55:00"
    }
}

```

## Interaction Type Configuration

After you have implemented the OData service you need to check whether the interaction types are created properly in the configuration.

1. To do so, open the *Manage Your Solution* app and choose *Configure Your Solution*.
2. Then search for the configuration user interface *Manage Interaction Content*
3. There you must have two entries for each of your interactions: One with the *Direction Outbound* and one with the *Direction Inbound* and with the corresponding interaction channel and communication media assigned.
4. Then you have to implement the enhancement (5) *Open Channel: Define Template for Outbound Interaction* that is mandatory for inbound interactions.

### Sample Code

```

template-id_origin          = 'SAP_HYBRIS_MKT_IC'.
template-interaction_type  = 'ZOC_CALL_CENTER_OUTB'.
template-communication_medium = 'BUSINESS_DOCUMENT'.

```

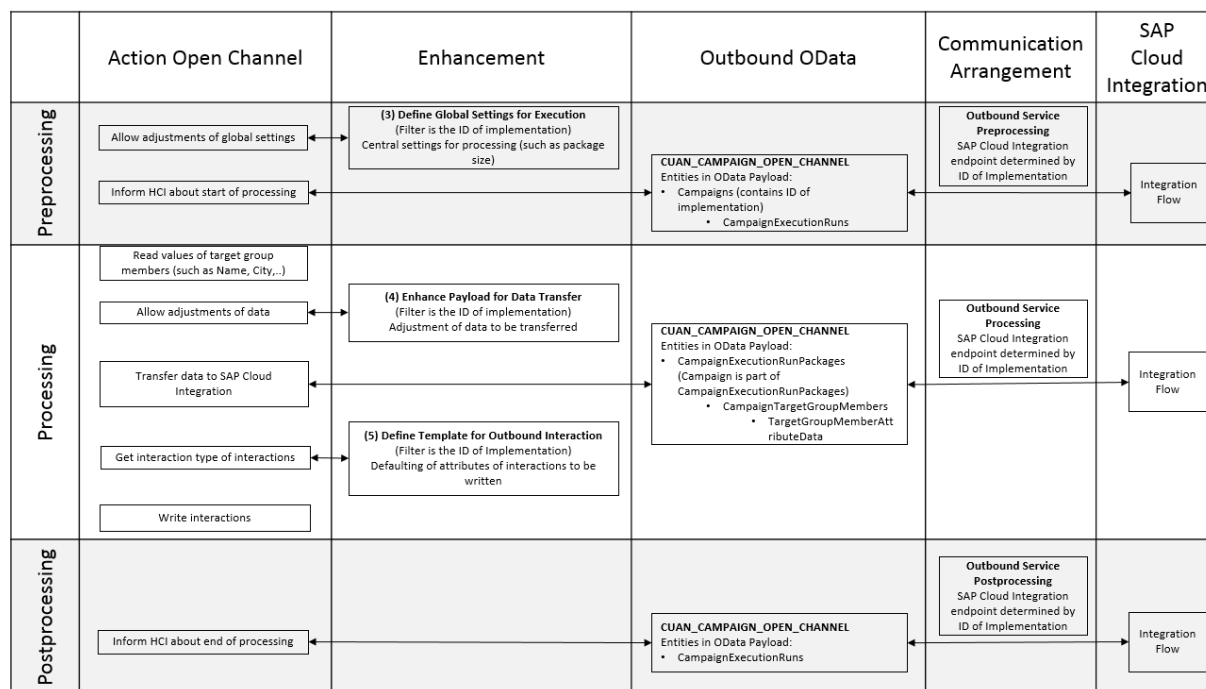
## Communication Arrangement

Now you have to setup inbound communication arrangement for integration scenarios **Business Data Integration** SAP\_COM\_0206

The path will be filled automatically with `/sap/opu/odata/sap/API_MKT_INTERACTION_SRV`. You need only to enter the prefix.

## 4.3.3.7 Data Flow

In the overview below you can see how the different pieces of enhancements, outbound OData and communication arrangement are used during the execution of the campaign action *Open Channel*:



The campaign execution calls the action *Open Channel* in 3 steps.

### 1. Preprocessing

In this step the enhancement *(3) Open Channel: Define global Settings for Execution* is called. The implementation of this enhancement is optional. The filter of the enhancement is the ID of the implementation which was defined in the enhancement *(1) Open Channel: Define Implementations*. In the enhancement some global settings can be made like the package size or whether marketing permissions shall be checked.

Please check the example coding for this enhancement in order to get more details.

If the outbound service `Preprocess` has been activated for the communication arrangement the campaign execution informs the SAP Business Technology Platform by using the OData `CUAN_CAMPAIGN_OPEN_CHANNEL`. In this call the payload consists of the following OData entities:

- Campaigns
- CampaignExecutionRuns

Note that the outbound service `Preprocess` is optional.

### 2. Processing

The campaign execution transfers the data in parallel processed packages.

In every package the action *Open Channel* reads the values of the target group members, for example, `<Name>`, `<City>`, as defined in the assigned export definition.

Now it calls the enhancement *(4) Open Channel: Enhance Payload for Data Transfer*. The implementation of this enhancement is optional. The filter of the enhancement is the ID of the implementation which was defined in the enhancement *(1) Open Channel: Define Implementations*. In the enhancement the data that shall be transferred can be checked, such as sort out an entry if `<Name>` and `<City>` is empty, and changed and enhanced, for example, calculate a value and append it.

Please check the example coding for this enhancement in order to get more details.

If the outbound service `Process` has been activated for the communication arrangement the campaign execution informs the SAP BTP by the usage of the OData `CUAN_CAMPAIGN_OPEN_CHANNEL`. In this call the payload consists of the following OData entities:

- Campaigns (Campaign is part of the `CampaignExecutionRunPackages`)
- `CampaignExecutionRunPackages`
  - `CampaignTargetGroupMembers`
    - `TargetGroupMemberAttributeData`

The OData creates one service call with the following entity sets: `CampaignExecutionRunPackages`, `CampaignTargetGroupMembers`, and `TargetGroupMemberAttributeData`

Note that the outbound service `Process` is mandatory.

After the OData call has been finished the action calls the enhancement [\(5\) Open Channel: Define Template for Outbound Interaction](#). The implementation of this enhancement is optional.

The filter of the enhancement is the ID of the implementation which was defined in the enhancement [\(1\) Open Channel: Define Implementations](#).

In the enhancement an interaction type can be set which is used in order to write outbound interactions for all target group members which have been transferred successfully.

Please check the example coding for this enhancement in order to get more details.

Finally the action [Open Channel](#) writes interactions. If in the enhancement [\(4\) Open Channel: Enhance Payload for Data Transfer](#) some target group members have been sorted out, corresponding interactions with a reason code will be written. For all other target group members successful outbound interactions will be written in case the enhancement [\(5\) Open Channel: Define Template for Outbound Interaction](#) has been implemented.

### 3. Postprocessing

After all packages have been processed the campaign execution calls the post processing of the action [Open Channel](#). If the outbound service `Process` has been activated for the communication arrangement the campaign execution informs the SAP BTP using OData `CUAN_CAMPAIGN_OPEN_CHANNEL`. In this call the payload contains the OData entity `CampaignExecutionRuns`. Note that the outbound service `Post Process` is optional.

## Related Information

[OData Service Settings for Outbound \[page 220\]](#)

### 4.3.3.8 Retry Sending Using Idempotency

With idempotency the system repeats a failed request up to five times to an external system using the connecting iFlow.

When the system sends the request to the external system things can go wrong either on the outbound or the inbound side. The system tries to resend up to five times. Only if all tries fail, the system throws an error message on the user interface and in the application log.

After you have set [Retry Send Active](#) property to **True (X)** in the setup of your communication arrangement, you have the following to do: Check your iFlow and enable it to accept several requests with the same key and

take care that the repetition sends always the same answer. This means that the same answer is sent only if the request has been processed properly by SAP Business Technology Platform.

#### ❁ Example

**First Attempt:** An error happens during send and SAP BTP isn't reached.

**Second Attempt:** An error occurs during receiving in SAP Marketing Cloud. The request has been processed properly and SAP BTP saves the response.

**Third Attempt:** An error occurs during receiving in SAP Marketing Cloud. The request has been already processed properly in the second trial. Though SAP BTP sends again the saved response.

**Fourth Attempt:** An error happens during send and SAP BTP isn't reached.

**Fifth Attempt:** No error, because the request has been already processed properly in the second trial. Though SAP BTP sends again the saved response.

The keys in question are:

- Preprocessing: `ExecutionRunKey (CampaignExecutionRuns)`
- Processing: `ExecutionRunKey` and `PackageID`
- Postprocessing: `ExecutionRunKey`

## Related Information

[Create Communication Systems and Arrangements \[page 218\]](#)

### 4.3.3.9 Questions and Answers

**Q: I implemented the enhancement (5) *Open Channel: Define Template for Outbound Interaction*, but the interactions are not written. What could be gone wrong?**

**A:** It might be that the ID is wrong, because the ID used for writing the interactions has to be the ID as specified by the `ID_ORIGIN`.

For example, when your enhancement contains the following line, the framework expects ERP customer identifier:

#### ≡ Sample Code

```
template-id_origin = 'SAP_ERP_CUSTOMER'
```

You could try to implement the enhancement (4) *Open Channel: Enhance Payload for Data Transfer* to set the right identifier:

#### Sample Code

```

LOOP AT target_group_member_status ASSIGNING FIELD-
SYMBOL(<ls_target_group_member_stat>).
  " Replace the given identifier by ID_ORIGIN specific identifiers
  CL_CUAN_INTERACT_CNTCT_HELPER=>GET_CONTACT_FACETS( EXPORTING
IT_CONTACT_KEYS = value #( ( conv #( <ls_target_group_member_stat>-
TG_MEMBER_INTERACTION_CONTACT ) ) )
                                IMPORTING
ET_CONTACT_FACET = data(lt_contact_facet) ).
  READ TABLE lt_contact_facet ASSIGNING FIELD-SYMBOL(<ls_contact_facet>)
WITH KEY id_origin = 'SAP_ERP_CUSTOMER'.
  IF SY-SUBRC EQ 0.
    <ls_target_group_member_stat>-communication_id = <ls_contact_facet>-id.
  ENDIF.
ENDLOOP.

```

The code snippet demonstrates only how to replace the identifiers!

Another solution would be the following coding:

#### Sample Code

```

LOOP AT target_group_member_status ASSIGNING FIELD-
SYMBOL(<ls_target_group_member_stat>).
  READ TABLE target_group_member_data ASSIGNING FIELD-
SYMBOL(<ls_target_group_member_data>)
    WITH KEY tg_member_key = <ls_target_group_member_stat>-
tg_member_key.
  IF sy-subrc eq 0.
    ...
  ELSE. " // No dynamic content found
    <ls_target_group_member_stat>-interaction_type = <your interaction type>.
    <ls_target_group_member_stat>-failure_reason = <your reason>.
  ENDIF.

```

Explanation:

- Interactions are only written when the enhancement (5) *Open Channel: Define Template for Outbound Interaction* has been implemented.
- If you have **NOT** implemented the ELSE loop, keep the following in mind:
  - The business logic sets the interaction types and optionally also the reasons.
  - The process message only contains the node `TargetGroupMemberAttributeData` It might happen that the figures under, if dynamic content has been found based on an assigned export definition.
- If you have implemented the ELSE loop, keep the following in mind:
  - Interactions are written with the interaction type and reason defined in the coding (ELSE loop) if dynamic content was not found.
  - In case dynamic content has been found, the business logic sets the interaction types and optionally also the reasons.
  - The process message only contains the node `OutboundId` (and subsequent nodes) if dynamic content is found based on an assigned export definition. Only target group members with dynamic content are send.



## Q: How to process interactions in the SAP system using the Open Channel integration combined with trigger-based campaigns?

A: With this code snippet you can, for example, access any interaction columns in the open channel integration and process them further by forwarding the information to SAP Business Technology Platform.

Note that you use any further descriptive information for trigger-based campaigns, such as column INTERACTIONCONTENT, in the enhancement (4) [Open Channel: Enhance Payload for Data Transfer](#).

### Sample Code

```
Sample Code
" Define some new fields to the message
APPEND VALUE #( attribute_id   = 'TRIGGER_INTERACTION'
                 attribute_name = 'Trigger Interaction'
                 attribute_type = 'Edm.String' ) TO
target_group_member_attributes.
APPEND VALUE #( attribute_id   = 'INTERACTION_CONTENT'
                 attribute_name = 'Interaction Content'
                 attribute_type = 'Edm.String' ) TO
target_group_member_attributes.
...
" Add the content for both fields
APPEND VALUE #( tg_member_key           = <ls_target_group_member_data>-
tg_member_key
                 tg_member_interaction_contact = <ls_target_group_member_data>-
tg_member_interaction_contact
                 tg_member_interaction       = <ls_target_group_member_data>-
tg_member_interaction
                 attribute_id               = 'TRIGGER_INTERACTION'
                 attribute_value           = <ls_target_group_member_stat>-
tg_member_interaction ) TO target_group_member_data.
IF <ls_target_group_member_stat>-tg_member_interaction IS NOT INITIAL.
  SELECT SINGLE interactioncontent FROM i_mkt_interaction INTO
  @DATA(lv_content_data) WHERE interaction = @<ls_target_group_member_stat>-
  tg_member_interaction.
  IF sy-subrc EQ 0.
    APPEND VALUE #( tg_member_key           =
<ls_target_group_member_data>-tg_member_key
                   tg_member_interaction_contact =
<ls_target_group_member_data>-tg_member_interaction_contact
                   tg_member_interaction       =
<ls_target_group_member_data>-tg_member_interaction
                   attribute_id               = 'INTERACTION_CONTENT'
                   attribute_value           = lv_interaction_content )
  TO target_group_member_data.
  ENDIF.
ENDIF.
```

## Q: What went wrong, when the figures of the campaign performance don't match?

A: [Performance](#) for your open channel campaign don't match. A reason for this mismatch could be that the interaction types aren't used in a consistent way.

The *Open Channel Interactions* tile represents the sum of all outbound and inbound interactions. For example, you have sent out 156 emails and 134 of them are opened. The figures shown are then as follows:

<i>Delivered Messages</i>	156
<i>Opened Messages</i>	134
<i>Open Channel Interactions</i>	290

To get consistent figures for your open channel campaign performance, such as the delivered messages, we recommend using also the interaction types in a consistent way. When you use, for example, the interaction types `EMAIL_OUTBOUND` and `EMAIL_OPENED` in your integration, the tiles *Delivered Messages* and *Opened Messages* show the correct numbers. This is also valid for tiles with calculated figures, such as *Opened Messages in %*.

For more information about the calculated figures, see [Aggregated Success Data from Interactions \[page 882\]](#).

## Q: Which URL shall I use for SAP BTP integration flow?

A: Each SAP BTP system provides multiple nodes. It is important to select the runtime node and not the tenant management node. The URL is visible in [CPI Operations View > Manage Integration Content](#). Then select the integration flow.

For more information, see also [Runtime in Detail](#).

## Q: Why do not I see any error messages for my wrongly implemented HEAD and GET requests?

A: The open channel functionality sends a http `HEAD` request followed by an http `POST` request. It depends on the implementation of the integration flow, whether the `HEAD` and `GET` requests are handled properly and doesn't lead to any error messages while running the iFlow.

## Q: Can I change the Multiple Value Separator for export definitions with Open Channel usage?

A: No. You may only overwrite the Multiple Value Separator in export definitions using the Export usage type.

Open Channel execution uses the export definition to retrieve the attributes to be sent to the Open Channel interface. The attributes are retrieved from calling the segmentation API as other actions such as Email or SMS. If you use a multiple-value personalization attribute, such as an item of interest, during the execution, the attribute is replaced with its characteristics (such as golf, football, swimming), separated by commas as the Multiple Value Separator.

## Related Information

[FAQ – SAP Marketing Cloud: Campaigns with Open Channel Integration](#)

### 4.3.3.10 DEPRECATED: Inbound Service Settings Using Import Service

Here you can find more information about the settings in the OData service that are required for the inbound of an open channel integration.

#### i Note

The inbound of the open channel integration has been deprecated as of SAP Marketing Cloud 1908. Please use as of 1908 [Inbound Service Settings API \[page 229\]](#).

The inbound comprises the creation of interactions (only creation) by using the deep-inserts of the OData service `CUAN_IMPORT`.

Endpoint OData Service Deep Insert: `/sap/opu/odata/sap/cuan_import_srv/ImportHeaders`

You need this path to create the service URL to your external system. In addition it tells you how the `ImportHeader` data is structured.

Entity Property of OData Service	Mandatory (must be filled by the OData service)	Copy (data is taken over from the outbound)	Remarks
Entity <code>ImportHeader</code> (This data is part of the import data.)			
Id			
<code>SourceSystemType</code> (optional)			
<code>SourceSystemId</code> (optional)			
Child Entity <code>Interaction</code> (This data is part of the import data.)			
<code>ContactIdOrigin</code>		X	There will be an error in case the inbound and outbound data of this property is not identical.
<code>ContactId</code>		X	There will be an error in case the inbound and outbound data of this property is not identical.

Entity Property of OData Service	Mandatory (must be filled by the OData service)	Copy (data is taken over from the outbound)	Remarks
CommunicationMedium		X	There will be an error in case the inbound and outbound data of this property is not identical.
InteractionType	X		<div style="border: 1px solid #0070C0; padding: 5px; background-color: #E6F2FF;"> <p><b>i Note</b></p> <p>Keep in mind that the <i>Direction</i> for the interaction type must be In-bound in the self-service configuration app Manage Interaction Content. For more information, see <a href="#">Managing Interaction Content</a>.</p> </div>
SourceObjectType	X		Passes always CUAN_CAMPAIGN_OUTBOUND
SourceObjectId	X		<p>Passes property OutboundId of the entity CampaignTargetGroupMember.</p> <p>There will be an error in case the SourceObjectId is empty.</p>
SourceSystemType			Both properties SourceSystemType and SourceSystemId are optional, but helpful values and therefore should be passed for describing the origin of the interaction.
SourceSystemId			
CampaignId		X	There will be an error in case the inbound and outbound data of this property is not identical.
InitiativeId		X	There will be an error in case the inbound and outbound data of this property is not identical.

Entity Property of OData Service	Mandatory (must be filled by the OData service)	Copy (data is taken over from the outbound)	Remarks
InitiativeVersion		X	There will be an error in case the inbound and outbound data of this property is not identical.
MarketingOrchestration Id		X	There will be an error in case the inbound and outbound data of this property is not identical.
AdditionalObjectReferences		X	There will be an error in case the inbound and outbound data of this property is not identical.
IsAnonymous			There will be an error in case IsAnonymous has the value <b>X</b> .
Child Entity <code>Product</code> (optional, child of <code>Interaction</code> )			
Child Entity <code>Interest</code> (optional, child of <code>Interaction</code> )			

The inbound processing retrieves all outbound interactions of the campaign by using the `SourceObjectId` and takes over data (**X** in column **Copy**) from the outbound interaction.

The OData entity `InteractionAdditionalObjectReference` is always be copied from the outbound interaction.

The following example coding shows the minimum payload required:

```

Sample Code

{
  "Id" : "",
  "SourceSystemType" : "",
  "SourceSystemId" : "32168",
  "Interactions" : [
    {
      "InteractionType" : "ZOC_CALL_CENTER_INB",
      "SourceObjectType" : "CUAN_CAMPAIGN_OUTBOUND",
      "SourceObjectId" : "CA5F1FE120237480E6054B06D61371081AE095DF",
      "SourceSystemType" : "",
      "SourceSystemId" : "32168"
    }
  ]
}

```

## Interaction Type Configuration

After you have implemented the OData service you need to check whether the interaction types are created properly in the configuration.

1. To do so, open the *Manage Your Solution* app and choose *Configure Your Solution*.
2. Then search for the configuration user interface *Manage Interaction Content*
3. There you must have two entries for each of your interactions: One with the *Direction Outbound* and one with the *Direction Inbound* and with the corresponding interaction channel and communication media assigned.
4. Then you have to implement the enhancement (5) *Open Channel: Define Template for Outbound Interaction* that is mandatory for inbound interactions.

### Sample Code

```
template-id_origin          = 'SAP_HYBRIS_MKT_IC'.
template-interaction_type   = 'ZOC_CALL_CENTER_OUTB'.
template-communication_medium = 'BUSINESS_DOCUMENT'.
```

## Communication Arrangement

Now you have to setup inbound communication arrangement for integration scenarios **Business Data Integration** SAP\_COM\_0004

The path will be filled automatically with `/sap/opu/odata/sap/cuan_import_srv`. You need only to enter the prefix.

### 4.3.4 Mobile, Social, and Digital Channel

With the integrations below you can interact with your customers and communities using social media.

#### [Integration with Google Ads \[page 243\]](#)

Overview of the integration scenario.

#### [Mobile App Integration with Google Firebase \[page 244\]](#)

This section describes how you can integrate SAP Marketing Cloud with Google Firebase for sending push notifications of mobile campaigns to a mobile app.

#### [Social Campaigns Using Facebook and Instagram \[page 254\]](#)

With this integration, you can plan and create campaigns in Facebook, and then use Facebook Ads Manager to push ads to Facebook and Instagram via Facebook. The actual spend and campaign success data from Facebook is pulled into SAP Marketing Cloud for analysis.

#### [Integration with LinkedIn \[page 257\]](#)

Send a target group to LinkedIn to create a Matched Audience for campaign targeting in LinkedIn.

#### [WeChat Integration \[page 258\]](#)

With this integration, you can synchronize the followers of your WeChat official accounts as well as the follower interactions to SAP Marketing Cloud. What's more, you can create and carry out WeChat campaigns through SAP Marketing Cloud. Analytical reports about WeChat followers and interactions are available as well.

#### [LINE Integration \[page 258\]](#)

With this integration, you can synchronize the followers of your LINE accounts as well as the follower interactions to SAP Marketing Cloud. What's more, you can create and carry out LINE campaigns through SAP Marketing Cloud. Analytical reports about LINE followers and interactions are available as well.

#### [Integration with Google Campaign Manager \[page 259\]](#)

Overview of the integration scenario.

#### [Integration with Adform \[page 261\]](#)

With this integration, you can send target groups that you created in SAP Marketing Cloud as custom audiences to Adform and use them in your Adform campaigns.

## 4.3.4.1 Integration with Google Ads

Overview of the integration scenario.

The integration with Google Ads allows you to create and assign Google Ads campaigns, then analyze the performance of these campaigns from SAP Marketing Cloud.

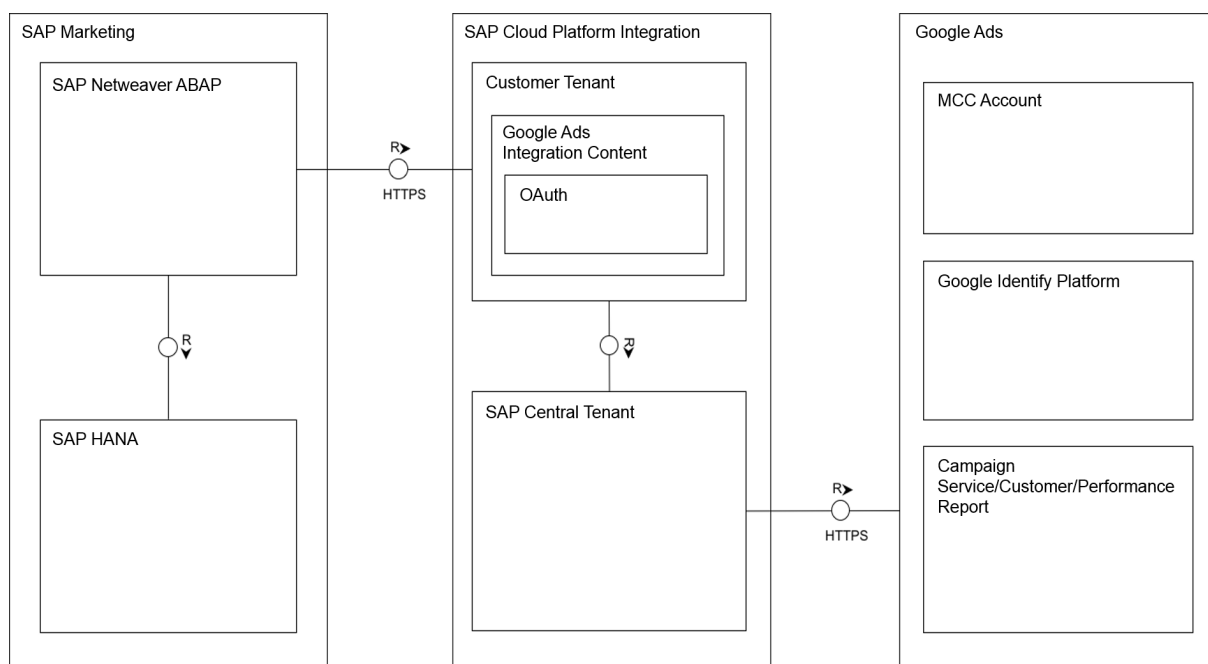
### **i** Note

You can assign a Google Ads campaign to an SAP Marketing Cloud campaign and generate a customer list in Google Ads. However, the ability to trigger the creation of a Google Ads campaign from SAP Marketing Cloud is deprecated and will be obsolete in an upcoming release.

For more information about the *Google Ads Integration with SAP Marketing/SAP Marketing Cloud* integration package, see [the SAP API Business Hub](#).

For more information about Google Ads Campaigns, see [Google Ads Campaigns](#).

The following diagram provides an overview of the main components involved in the integration with Google Ads. SAP Cloud Integration is used as a middleware between SAP Marketing Cloud and Google Ads. It is responsible for the account authentication with OAuth 2.0 and any other API communication routing between the two involved systems.



## Configuration Settings

To run the integration scenario, make settings in the following systems:

- Google Ads
- SAP Cloud Integration
- SAP Marketing Cloud

For a complete description of the configuration settings required for the scenario, see the [Integration Guide](#).

### 4.3.4.2 Mobile App Integration with Google Firebase

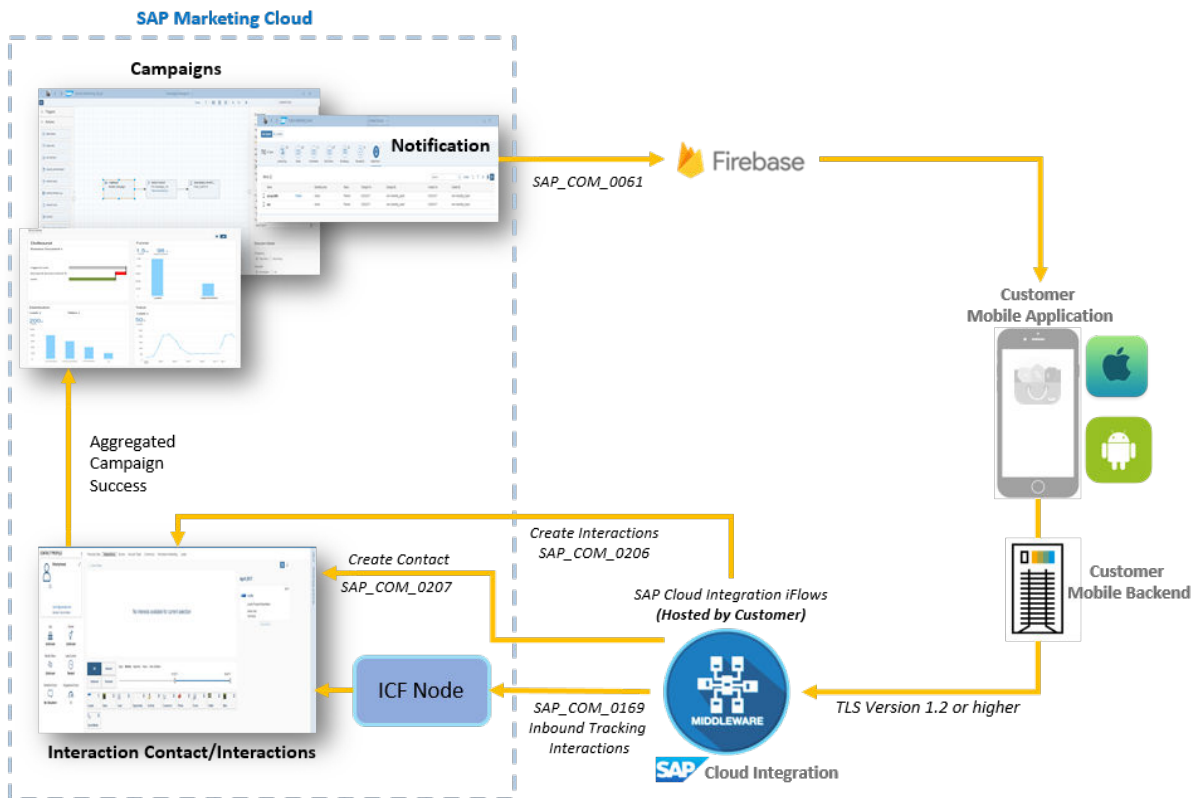
This section describes how you can integrate SAP Marketing Cloud with Google Firebase for sending push notifications of mobile campaigns to a mobile app.

The following graphic illustrates the end-to-end flow for enabling the mobile channel feature. You can create a mobile campaign in the SAP Marketing Cloud system. To this campaign, you can assign an offer or a notification. The offers and notifications are sent as mobile push notifications to either Android or iOS devices. The mobile push notifications are routed via the Google Firebase.

For the mobile device to connect to a SAP Marketing Cloud system, you must use [SAP Cloud Integration](#). [SAP Cloud Integration](#) iFlows can be leveraged to connect your mobile app to SAP Marketing Cloud via your mobile app's backend system. SAP recommends not to connect the mobile app directly to the [SAP Cloud Integration](#) for security reasons. After you've deployed the iFlows, you must set up the communication scenarios for inbound and outbound communication.

For more information, see [Mobile App Integration with SAP Marketing Cloud](#).





### i Note

Transport Layer Security (TLS) version 1.2 or higher is required if you're using a servlet instead of the CPI iFlow for inbound communication from the Customer Mobile Backend.

## 4.3.4.2.1 Configuring Firebase

### Procedure

1. Create a google account.
2. Log in to Firebase Console using the google account.
3. Create a new project in Firebase.
4. Follow steps provided in this [link](#) to add iOS and Android apps.

### i Note

For Android devices, SAP uses the data message format provided by Firebase instead of the standard notification message format. User applications have to handle the notifications appropriately. For more

information, see [Firebase Cloud Messaging \(FCM\)](#) and [Sample Payload of Mobile Push Notification \[page 248\]](#).

## 4.3.4.2.2 Configuring Inbound Communication

Create the communication user, communication system, and communication arrangement required for the inbound communication.

### Prerequisites

- You have configured Firebase.
- The following communication scenarios are available in the system:
  - SAP\_COM\_0206 (Marketing - Interaction UI Integration )
  - SAP\_COM\_0207 (Marketing - Interaction Contact UI Integration )
  - SAP\_COM\_0169 (Marketing - Mobile Push Notification Events Integration)

### Context

For inbound communication, you first create a communication user, then a communication system and communication arrangement.

### Creating the Communication User

To add a communication user, proceed as follows:

1. Log on to SAP Marketing Cloud with a user that has administrator authorizations.
2. From the SAP Fiori launchpad, choose ► [Communication Management](#) ► [Communication User](#) ►.
3. Choose [New](#).
4. Enter the user name and password for your communication user.
5. Choose [Save](#).

#### i Note

You can use the communication user information for setting up communication with communication scenarios [SAP\\_COM\\_0206](#), [SAP\\_COM\\_0207](#), and [SAP\\_COM\\_0169](#).

### Creating the Communication System

The purpose of this communication system is to bind the communication user that you created earlier with the communication arrangement that you will create later.

To create the communication system, proceed as follows:

1. Open the *Communication Systems* app.
2. Choose *New*.  
The *New Communication System* dialog box appears.
3. Enter a system ID and its name.
4. Choose *Create*.
5. A host is irrelevant to the inbound communication. Enter **dummy** in the *Host Name* field to assign a dummy host.
6. Assign the communication user created earlier to this communication system, as follows:
  1. In the *User for Inbound Communication* section, choose *+ (Add)*.  
The *New Inbound Communication User* dialog box appears.
  2. Select the authentication method as **User Name and Password** and enter the user created earlier.
7. Save and activate the communication system.

### Creating the Communication Arrangement

To create a communication arrangement, proceed as follows:

1. Open the *Communication Arrangements* app.
2. Choose *New*.
3. Enter scenario **SAP\_COM\_0206** and an arrangement name. Choose *Create*.
4. In the *Communication System* field, enter the communication system created earlier.
5. Save and activate the communication arrangement.
6. Similarly, create communication arrangement for the following scenarios:
  - SAP\_COM\_0207
  - SAP\_COM\_0169

### Next Steps

[Configuring Outbound Communication \[page 247\]](#)

## 4.3.4.2.3 Configuring Outbound Communication

Create the communication system and communication arrangement required for the outbound communication.

### Context

For outbound communication, you first create a communication system and then a communication arrangement.

### Creating the Communication System

Proceed as follows:

1. Log on to SAP Marketing Cloud with a user that has administrator authorizations.
2. From the SAP Fiori launchpad, choose ► *Communication Management* ► *Communication Systems* ►.
3. Choose *New*.
4. Enter an ID and a system name for your communication system.
5. Choose *Create*.
6. In the *Communication System Draft* screen, enter the host name as **fcm.googleapis.com**.
7. Choose the + button in the *User for Outbound Communication* section.
8. In the *New Outbound Communication* user dialog, choose the *Authentication Method* as **SSL Client Certificate**.
9. Set the Certificate Type to *Default Client Certificate*.
10. Choose *Create* in the popup.
11. Save and activate the communication system.

### Creating the Communication Arrangement

Proceed as follows:

1. Log on to SAP Marketing Cloud with a user that has administrator authorizations.
2. From the SAP Fiori launchpad, choose the *Communication Arrangements* app.
3. Create a new communication arrangement.
4. Choose the scenario *SAP\_COM\_0061* (Marketing - Campaign Mobile Channel Integration).
5. Enter the arrangement name *SAP\_COM\_0061*.
6. Choose *Create*.
7. In the *Communication Arrangement* screen for Mobile Campaign, choose the communication system that you created previously.

#### i Note

All the outbound service URLs are populated in the *Path* field automatically. For Mobile Campaign, please enter the value `/fcm/send`. Also, ensure the Service URL is `https://fcm.googleapis.com/fcm/send`.

8. In the additional properties section, enter an appropriate value for Firebase API Key.
9. Save and activate the communication arrangement.

#### i Note

Ensure that there is only one communication arrangement for scenario *SAP\_COM\_0061* and it is active.

## 4.3.4.2.4 Sample Payload of Mobile Push Notification

After successful setup, the SAP Marketing Cloud system sends push notification to the mobile device.

The following are sample mobile push notification payloads:

### Sample Code

Payload on iOS device

```
{
  "notification":{
    "body":"<NOTIFICATION_BODY_TEXT>",
    "title":"<NOTIFICATION_TEXT_TITLE>",
    "icon":"",
    "click_action":"OPEN_NOTIFICATION"
  },
  "registration_ids":[
    "<REGISTRATION_ID>"
  ],
  "data":{
    "origin":"YMKT",
    "imageURL":"<http>",
    "type":"<COUPON/OFFER/TEXT>",
    "offer":{
      "id":"<OFFER_ID>",
      "name":"<OFFER_NAME>",
      "coupon":"<COUPON_NAME / ID>",
      "image":"<http>",
      "target":"<http>",
      "couponCode":"SAMPLE-CODE-743",
      "EANCodeImageURL":"<https://yourdomain.com/url_to_eancode/743>",
      "QRCodeImageURL":"<https://yourdomain.com/url_to_qrcode/743>",
      "couponCodeSerialNumber":"<SERIAL-NUMBER-743>"
    },
    "trackingURL":"<https>",
    "campaignID":"<CAMPAIGN_ID>"
  },
  "mutable_content":"<true/false>",
  "priority":"<PRIORITY_#>"
}
```

### Sample Code

Payload on Android device

```
{
  "registration_ids":[
    "<REGISTRATION_ID>"
  ],
  "data":{
    "origin":"YMKT",
    "deeplinkTarget":"<DEEPLINK_TARGET_URL>",
    "imageURL":"<http>",
    "type":"<COUPON/OFFER/TEXT>",
    "offer":{
      "id":"<OFFER_ID>",
      "name":"<OFFER_NAME>",
      "coupon":"<COUPON_NAME / ID>",
      "image":"<http>",
      "target":"<http>",
      "couponCode":"SAMPLE-CODE-743",
      "EANCodeImageURL":"<https://yourdomain.com/url_to_eancode/743>",
      "QRCodeImageURL":"<https://yourdomain.com/url_to_qrcode/743>",
      "couponCodeSerialNumber":"<SERIAL-NUMBER-743>"
    },
    "trackingURL":"<TRACKING_URL>",
    "campaignID":"<CAMPAIGN_ID>",
    "body":"<NOTIFICATION TEXT BODY>",
    "title":"<NOTIFICATION_TEXT_TITLE>"
  }
}
```

```

    },
    "mutable_content": "<true/false>",
    "priority": "<PRIORITY_>"
  }

```

The following table contains the list of available parameters:

Parameter	Platform	Description
notification	iOS	This parameter specifies the predefined, user-visible key-value pairs of the notification payload.
body	iOS/ Android	The notification's body text.
title	iOS/ Android	The notification's title.
icon	iOS	The notification's icon.
click_action	iOS	The action associated with a user click on the notification. Corresponds to <code>category</code> in the Apple Push Notification (APN) payload.
data	iOS/ Android	This parameter specifies the custom key-value pairs of the message's payload.
registration_ids	iOS/ Android	This parameter specifies the client apps (registration tokens) receiving the message.
origin	iOS/ Android	The hardcoded YMKT value reflects the system the notification originated from.
deeplinkTarget	Android	The URL that points to the exact link that is relevant for the mobile app user. You can direct your mobile app users to the relevant destination using this link.

#### **i Note**

Multicast messaging (sending to more than 1 registration token) is only allowed using HTTP JSON format.

Parameter		Platform	Description
type		iOS/ An- droid	<p>You can define one of the following types:</p> <ul style="list-style-type: none"> <li>• offer</li> <li>• coupon</li> <li>• text</li> </ul> <p>You can find their respective associated parameters below.</p>
offer	id	iOS/ An- droid	The ID of the offer.
	name	iOS/ An- droid	The name of the offer.
	image	iOS/ An- droid	The image URL for the offer.
	target	iOS/ An- droid	The URL that is navigated to by clicking on the image.
coupon	id	iOS/ An- droid	The ID of the offer.
	name	iOS/ An- droid	The name of the offer.
	coupon	iOS/ An- droid	The SAP Marketing Cloud coupon name and ID.
	image	iOS/ An- droid	The image URL for the offer.
	target	iOS/ An- droid	The URL that is navigated to by clicking on the image.
	couponCode	iOS/ An- droid	The coupon code.
	EANCodeImageURL	iOS/ An- droid	The image URL to European Article Numbers (EAN) code.
	QRCodeImageURL	iOS/ An- droid	The image URL to Quick Response (QR) code.

Parameter		Platform	Description
	couponCodeSerial Number	iOS/ An- droid	The serial number of the coupon code.
text		iOS/ An- droid	This type simply sends a notification that contains the body and title on the notification.
trackingURL		iOS/ An- droid	You can use the URL to track whether the user has viewed the mobile notification. When the link is opened, the system creates an interaction in the SAP Marketing Cloud system.
<div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>To enable this feature, you must configure and deploy the associated integration flow. For more information, see <a href="#">Configure and Deploy the 'Create Interaction Using Tracking URL in SAP Marketing Cloud for Mobile Application Integration' Integration Flow</a>.</p> </div>			
campaignID		iOS/ An- droid	The ID of an SAP Marketing Cloud campaign.
mutable_content		iOS/ An- droid	Currently for iOS 10+ devices only. On iOS, use this parameter to represent mutable-content in the APN's payload. When a notification is sent and <code>mutable_content</code> is set to true, the content of the notification can be modified before it's displayed, using a Notification Service app extension.
<div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>This parameter is ignored for Android and web.</p> </div>			



Parameter	Platform	Description
<code>priority</code>	iOS/ Android	<p>Sets the priority of the message. Valid values are "normal" and "high." On iOS, these correspond to APN's priorities 5 and 10.</p> <p>By default, notification messages are sent with high priority, and data messages are sent with normal priority. Normal priority optimizes the client app's battery consumption and should be used unless immediate delivery is required. For messages with normal priority, the app may receive the message with unspecified delay.</p> <p>When a message is sent with high priority, it's sent immediately, and the app can display a notification.</p>

### i Note

For Android devices, notifications must be explicitly handled irrespective of whether the app is in the foreground, background, or killed. The app must implement the `onMessageReceived()` method of the `FirebaseMessagingService`. For more information, see [Receive Messages in an Android App](#).

## 4.3.4.2.5 Maintaining the Certificate Trust List

To add the SSL certificate of Firebase to the trust list of SAP Marketing Cloud, perform the following steps:

### Procedure

1. Download all of the Google services' certificates issued by any Certificate Authority from [this link](#).

### i Note

As described on the [Google Trust Services FAQ](#), this list is updated regularly. Applications connecting to Google services should trust all of the Certificate Authorities contained in this list.

2. Open the `roots.pem` file you've downloaded and segment each certificate (-----BEGIN CERTIFICATE----- / -----END CERTIFICATE-----) into its own separate `*.pem` file.
3. In SAP Marketing Cloud, choose the [Maintain Certificate Trust List](#) app.
4. Choose **+**.
5. Upload each of the `.pem` files that you segmented from the original download.

### 4.3.4.3 Social Campaigns Using Facebook and Instagram

With this integration, you can plan and create campaigns in Facebook, and then use Facebook Ads Manager to push ads to Facebook and Instagram via Facebook. The actual spend and campaign success data from Facebook is pulled into SAP Marketing Cloud for analysis.

#### i Note

Please note that SAP Marketing Cloud doesn't support direct integration with Instagram.

### Prerequisites on Facebook

Before you begin, a few things need to be done:

- You need your own Facebook app that must be reviewed and released for productive usage by Facebook. When starting the review process, mention that you are using SAP Marketing Cloud. You can only use one Facebook app with this integration.

#### i Note

A prerequisite for the approval is a link to a data privacy policy that is visible to every user of the app. Ensure that your company has such a policy in place.

- Look up the application ID (*App ID*) and client secret (*App Secret*) in Facebook for later use when configuring the communication arrangement.
- To actually do advertising on Facebook you need a Facebook ad account and a user that has been assigned either the *Ad Account Admin* or *Ad Account Advertiser* permissions for at least one ad account in Facebook. If you work together with a marketing agency you have to clarify who owns and manages this account.. It is recommended to use *Facebook Business Manager*. For details refer to the Facebook documentation. In any case your users need marketer permissions on the ad account. You can also work with multiple ad accounts (such as one account per marketing area).

#### i Note

When you create a Facebook campaign in SAP Marketing Cloud, you need to select an ad account from the *Advertiser* dropdown. This dropdown displays all available Facebook ad accounts, regardless of the type of permission. Only ad accounts with *Admin* or *Advertiser* permissions can be used to create campaigns.

When creating a campaign, selecting an ad account with only *Analyst* permissions will result in an error and you will need to choose a different ad account to continue. When transferring a custom audience, using an ad account with only *Analyst* permission will result in an error, and you may need to discard that campaign and start again.

Facebook also requires a check for marketing permissions when using their custom audiences, a check that is done by default in SAP Marketing Cloud. In SAP Marketing Cloud, if the user has not given their permission for their data to be used for advertising purposes it can not be used in campaigns created either for Facebook custom audiences or third-parties. However, some countries have implicit opt-in permission. This means that if the user does not specifically forbid SAP Marketing Cloud from using their user for advertising purposes, the user information can be transferred to a custom audience in Facebook.

Your marketing ad account manager and system administrators can assist you with any questions about these prerequisites.

## Creating a Facebook App

You must request a Facebook app to access the Facebook APIs. This app must be reviewed by Facebook in order to get full API access. Please keep in mind that Facebook can reject your app, and the review process belongs solely to Facebook. SAP is not involved in this process.

### Permissions Needed

The following permissions are necessary to integrate your Facebook campaigns with SAP Marketing Cloud.

- Facebook Login
- Ads\_management (Standard Access)

To gain these permissions, you will need to provide videos to Facebook as part of the app review process. These videos should show how certain features of the Facebook API are used in SAP Marketing Cloud.

You must produce these videos yourself, and they will need to show your Facebook app when demonstrating login steps.

### Facebook Login Permission

You need to create a video to demonstrate how the Facebook Login is used in SAP Marketing Cloud. To do this, create a Facebook campaign and choose *Authenticate* to trigger the user authentication flow to Facebook. It is important to show that the login uses your Facebook app.

### Ad\_management Permission


You need to create a video to demonstrate how the Ads\_management feature is used in SAP Marketing Cloud. You will need to create another video demonstrating the Facebook campaign creation and authentication process, as well as showing the *Performance* tab with analytics available. You can use a CSV upload to add data to this campaign if you do not have a real example of a Facebook campaign yet.

### Additional Notes

For the *App Domains* of your app, you should use your company-specific tenant which can be copied from your browser.

For the review process, and later for productive usage, you will need to turn on the *Live* mode of your app. When you switch to *Live* mode, all of the permissions allowed in *Development* mode are removed. Until those permissions are restored, some features, such as Facebook login in SAP Marketing Cloud, may not work. After the app is reviewed and approved, these permissions will be restored and the features will work again.

### Additional Information

For an example about how to create a Facebook app which can be used for SAP Marketing Cloud, see the following blog: [How to Create a Facebook App for Campaign and Custom Audience Integration](#) 

#### i Note

The SAP blog is not part of the official documentation of SAP Marketing Cloud and some of the information may be outdated.

## Setting Up Communication Arrangement

For more information, see [Setting Up a Communication Arrangement with Facebook \[page 256\]](#).

### Related Information

[Transferring Target Groups to Facebook Custom Audiences](#)  
[Facebook Campaign Performance](#)

### 4.3.4.3.1 Setting Up a Communication Arrangement with Facebook

In SAP Marketing Cloud, you can create and track the success and actual spend of your Facebook campaigns and create Facebook custom audiences for your target groups to be used in either one-time or periodic campaigns.

You set up a communication system and communication arrangement to enable the Facebook execution, and the requesting of success data from Facebook. Only one communication arrangement is needed.

The authentication with Facebook is done with OAuth authorization code grant flow and there's no setup of a technical user for communication. Instead, an SAP Marketing Cloud user grants SAP Marketing Cloud the permission to manage ads on Facebook on behalf of their personal Facebook user from the SAP Marketing Cloud *Campaigns* app.

#### Prerequisites

You need your *App ID* and *App Secret* from Facebook.

You also require the business catalog role `SAP_BR_ADMINISTRATOR` in SAP Marketing Cloud.

You require at least one Facebook user with "Ad Account Admin" or "Ad Account Advertiser" permissions for at least one Facebook ad account. Permissions can be granted, for example, by using the Facebook Business Manager.

#### Redirect URL

To make your OAuth client known to the OAuth authorization server on Facebook, you need to enter a redirect URL in your Facebook app. Your company-specific tenant can be extracted from your browser and put together as follows: `https://<your domain>:443/sap/public/bc/sec/oauth2/client/redirect?sap-client=100`.

#### i Note

To enable the redirect URL for SAP Marketing Cloud, configure the following settings in the Facebook App Dashboard:

- Add the redirection URL in [|| PRODUCTS > Facebook Login > Settings > Valid OAuth Redirect URIs >](#).
- Add the company specific tenant (your domain) in [|| Settings > Advanced > Domain Manager >](#).

## Communication System

Create the communication system as follows:

1. In the SAP Fiori launchpad, click *Communication Systems*. In *Communication Systems*, click *New*.
2. In the *New Communication System* dialog, define the ID for the communication system. Define a *System Name*. You can freely define a name; note that the name is used when you create the communication arrangement. Click *Create*.
3. Under **► Technical Data ► Host Name ▾**, specify the host system for Facebook, which is **facebook.com**.

### i Note

*Log System ID, Client Name, and Business System* aren't relevant for Facebook campaign execution.

4. Optionally, you can provide your *Contact Information* for the communication system you're defining.
5. Under *User for Outbound Communication*, click + to add a set of access details for the Facebook server. You use the Authentication Method OAuth 2.0 and provide the *App ID* and *App Secret* that you've obtained before. Click *Create*.
6. Save the communication system.

## Maintain Communication Arrangement for Facebook

1. From the launchpad, choose **► Communication Management ► Communication Arrangements ▾**
2. From the *Communication Arrangement* screen, choose *New* and enter the following information in the New Communication Arrangement dialog:
  - *Communication Scenario ID*: SAP\_COM\_0031 (*Marketing - Campaign Execution Facebook Integration*)
  - *Communication Arrangement Name*: Enter a name of your choice, for example "Facebook"
3. Select the communication system you created under Communication System.
4. Make sure that the *OAuth 2.0 Client ID* matches the ID you wish to use for your Facebook campaign.
5. Enter a / or other data of your choice in the *Path* field.
6. Save your communication arrangement.

## Related Information

[Social Campaigns Using Facebook and Instagram \[page 254\]](#)

### 4.3.4.4 Integration with LinkedIn

Send a target group to LinkedIn to create a Matched Audience for campaign targeting in LinkedIn.

The integration with LinkedIn allows you to send a target group from SAP Marketing Cloud to LinkedIn to generate a Matched Audience. The frequency of the campaign defines how often the target group is resent to LinkedIn to refresh the Matched Audience. You send the target group to a specific ad account in LinkedIn, which the campaign user must be authorized to access. The campaign user must be authenticated by LinkedIn with a valid OAuth token before they can start the campaign and send the target group.

If the minimum size of 300 matched members is met, then a Matched Audience is created. The audience size is fetched from LinkedIn and displayed in the LinkedIn Audience Transfer action in SAP Marketing Cloud.

For more information about Matched Audiences in LinkedIn, see [LinkedIn Matched Audiences - Overview](#) .

### Prerequisites

Complete the following configuration activities once you have determined your privacy policies and communicated them to your customers:

- Assign the origin of contact `EMAIL` to the communication medium LinkedIn (`LNKD`) in the configuration activity **Assign Contact ID Origin to Communication Media**. For more information, see [Assign Contact ID Origin to Communication Media](#).
- Set your permitting handling per country and for communication medium `LNKD` in the configuration activity **Define Marketing Permission Check**. For more information, see [Define Marketing Permission Check](#).

### Permissions for Contacts

#### → Remember

You're responsible for your privacy policies regarding the transfer of contact data to third parties.

## Related Information

[LinkedIn Audience Transfer](#)

### 4.3.4.5 WeChat Integration

With this integration, you can synchronize the followers of your WeChat official accounts as well as the follower interactions to SAP Marketing Cloud. What's more, you can create and carry out WeChat campaigns through SAP Marketing Cloud. Analytical reports about WeChat followers and interactions are available as well.

For more information, see [WeChat Integration \[page 65\]](#).

### 4.3.4.6 LINE Integration

With this integration, you can synchronize the followers of your LINE accounts as well as the follower interactions to SAP Marketing Cloud. What's more, you can create and carry out LINE campaigns through SAP Marketing Cloud. Analytical reports about LINE followers and interactions are available as well.

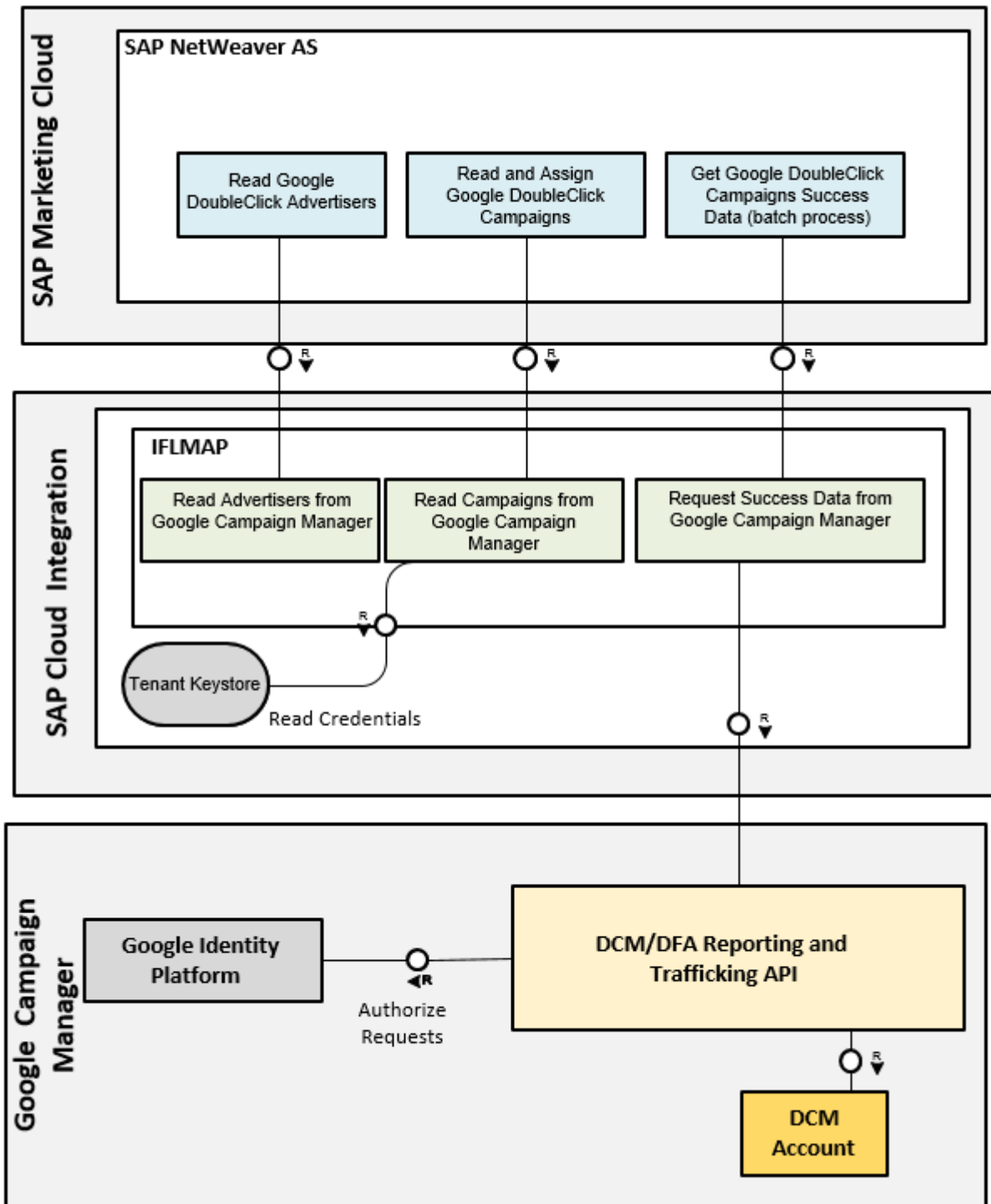
For more information, see [LINE Integration \[page 77\]](#).

## 4.3.4.7 Integration with Google Campaign Manager

Overview of the integration scenario.

This integration allows you to transfer performance data from Google Campaign Manager to SAP Marketing Cloud.

The following diagram provides an overview of the main components involved in the system integration of SAP Marketing Cloud with Google Campaign Manager. SAP Cloud Integration is used as a middleware between SAP Marketing Cloud and Google Campaign Manager. It is responsible for the account authentication with OAuth 2.0 and any other API communication routing between the two involved systems.



## Configuration

To run the integration scenario, you make settings in the following systems:

- Google Campaign Manager



- SAP Cloud Integration
- SAP Marketing Cloud

## Related Links

For a complete description of the configuration settings required for the scenario, see the [Integration Guide](#).

For more information about the *Google Campaign Manager Integration with SAP Marketing Cloud/SAP Marketing* integration package, see [Google Campaign Manager Integration with SAP Marketing Cloud/SAP Marketing](#).

For more information about Google Campaign Manager campaigns, see [Google Campaign Manager Campaigns](#).

### 4.3.4.8 Integration with Adform

With this integration, you can send target groups that you created in SAP Marketing Cloud as custom audiences to Adform and use them in your Adform campaigns.

## Prerequisites

Before you can use this integration, you must do the following:

- Set up your data provider account on the Adform platform.  
Only one data provider account should be assigned to your Adform platform. You choose the data provider account in the *ADFORM AUDIENCES* node in the *Campaigns* app.
- Create categories in your data provider account.
- Download the Adform server certificate (api.adform.com).

## Configuration

To use this integration you must set up the communication system and communication arrangement in your SAP Marketing Cloud system and import the Adform server certificate into SAP Marketing Cloud.

## Related Information

[Configuring SAP Marketing Cloud \[page 262\]](#)

[Import the Certificate \[page 264\]](#)

[Adform Audiences Campaign](#)

## 4.3.4.8.1 Configuring SAP Marketing Cloud

To establish communication with the OData service, you perform procedures in SAP Marketing Cloud.

The overall process is as follows:

1. [Set Up the Communication System \[page 262\]](#)  
Set up a communication system for the Adform integration scenario.
2. [Set Up the Communication Arrangement \[page 263\]](#)  
After setting up the communication system, set up the communication arrangement for the Adform integration scenario.

### 4.3.4.8.1.1 Set Up the Communication System

Set up a communication system for the Adform integration scenario.

#### Prerequisites

To set up a communication system and communication arrangement, you require the Communication Management (SAP\_CORE\_BC\_COM) business catalog role.

#### Procedure

1. Log on to your SAP Marketing Cloud system with a user that has administrator authorizations.
2. From the SAP Fiori launchpad, choose the *Communication Systems* app.
3. Choose *New*.
4. Enter a system ID and name for your communication system.
5. Choose *Create*.
6. On the *Communication System* page, enter the following:
  - a. Under *Technical Data*, enter **dummy** as the *Host Name* to assign a dummy host.  
  
This is a dummy communication system as its only purpose is to bind the communication user that you previously created to the communication arrangement that you will create in the next step.
  - b. Under *User for Outbound Communication*, choose (+).
  - c. For *Authentication Method*, select *OAuth 2.0*.
  - d. Enter your client ID and client secret you received from Adform.
7. Save your changes and exit the app.

**Task overview:** [Configuring SAP Marketing Cloud \[page 262\]](#)

**Next task:** [Set Up the Communication Arrangement \[page 263\]](#)

## 4.3.4.8.1.2 Set Up the Communication Arrangement

After setting up the communication system, set up the communication arrangement for the Adform integration scenario.

### Prerequisites

To set up a communication system and communication arrangement, you require the Communication Management (SAP\_CORE\_BC\_COM) business catalog role.

### Procedure

1. Log on to your SAP Marketing Cloud system with a user that has administrator authorizations.
2. From the SAP Fiori launchpad, choose the *Communication Arrangements* app.
3. Create a new communication arrangement.
4. Select scenario *SAP\_COM\_0496*.
5. Choose *Create*.
6. In the *Communication Arrangements* screen, do the following:
  - a. Under *Common Data*, choose the communication system that you created previously.
  - b. Under *User for Outbound Communication*, choose the user name you assigned to the communication system.
7. Save your changes and exit the app.

**Task overview:** [Configuring SAP Marketing Cloud \[page 262\]](#)

**Previous task:** [Set Up the Communication System \[page 262\]](#)

## 4.3.4.8.2 Import the Certificate

Import the server certificate for Adform into SAP Marketing Cloud.

### Prerequisites

You already downloaded the certificate from Adform.

### Procedure

1. Log into the SAP Fiori launchpad with a business role that contains the security (`SAP_CORE_BC_SEC`) business catalog.
2. Open the *Maintain Certificate Trust List* app.
3. Use (+) to add a certificate .
4. Upload the certificate.

## 4.3.5 Setting Up Captcha Configuration for Forms

Use captcha configuration to enhance the security of your forms and decrease vulnerability to malicious attacks by bots that send fraudulent contact data into your system.

SAP Marketing Cloud integrates with Google reCAPTCHA v3 and provides a *Captcha* element in the Form editor allowing you to set up captchas for your forms, analyze form traffic, and take action to ensure your forms are secure.

To set up the communication between your SAP Marketing Cloud system and Google reCAPTCHA v3, you do the following:

- Download the **GlobalSign** certificate from Google Admin Console.
- Upload the **GlobalSign** certificate to the Maintain Certificate Trust List app.

#### i Note

If the *GlobalSign* certificate is already listed in the *Maintain Certificate Trust List*, you can skip the download and upload steps.

- Register your domain on the Google Admin Console for Google reCAPTCHA v3.
- Create a communication system.
- Create a communication arrangement using the *Marketing - Captcha Integration (SAP\_COM\_0584)* communication scenario. The scenario is used to store the site and secret keys that are generated when you register your domain.

## Download the GlobalSign Certificate

You must download the **GlobalSign Root CA-R2** certificate to ensure that the Captcha communication works as required. You perform the following steps:

1. Open the Google Admin Console in a browser window.

### i Note

The steps for downloading the certificate will differ depending on the device and browser you use. For example, the certificate can be copied either directly to your desktop or you may encounter a step-by-step wizard to support you with the download.

2. Click the small lock icon in the address bar. This is located either to the left or right of the URL.
3. Select the **GlobalSign Root CA-R2** certificate and drag and drop it to your local computer or follow the steps to copy the file to your local computer. Again, how you perform this step depends on the device and browser you are using.

## Upload the GlobalSign Certificate

To upload the **GlobalSign Root CA-R2** certificate, you perform the following steps:

1. Open the Maintain Certificate Trust List app.
2. Choose **+**.
3. Browse your computer for the certificate file you saved by following the steps in the previous section.
4. Select the file and choose *Upload*.

## Register Your Domain

### Prerequisites

To register your domain for Google reCAPTCHA v3, you must have your own Google account.

You register your domain with Google reCAPTCHA v3 as follows:

1. Log in to your Google account.
2. Open the Google Admin Console: <https://www.google.com/recaptcha/admin/create>
3. Enter a label for your form or landing page where the form is embedded. Use a label that's easy to remember.
4. Select reCAPTCHA v3 for the type.
5. Add the domain of your form or landing page where the form is embedded.

### i Note

The domain must be that of the published form or landing page.

6. Add the domain of your SAP Marketing Cloud system, for example *my1234567-api.s4hana.ondemand.com*. This ensures that your form captchas will work correctly when opening your form in test mode.

7. Select the checkbox to accept the reCAPTCHA terms of service and choose [Submit](#). A Site Key and a Secret Key are created and saved in the Admin console settings.  
These keys are necessary when creating the communication arrangement. The site key is used to call the reCAPTCHA service on your form. The secret key authorizes communication between your SAP Marketing Cloud system and the reCAPTCHA server to verify the user's response.

## Create a Communication System

You create the communication system as follows:

1. In the SAP Fiori launchpad, open the Communication Management group.
2. Open the Communication Systems app and choose [New](#).
3. In the [New Communication System](#) dialog, enter names for the system ID and system name. For example, GOOGLE\_CAPTCHA\_v3. You can freely define a system name. It is used when you create the communication arrangement.
4. Choose [Create](#).
5. In the [Technical Data](#) section, enter www.google.com for the host name.
6. In the [Users for Outbound Communication](#) section, choose the plus icon to define the outbound authentication method.
7. Select [None](#) for the authentication method and choose [Create](#).
8. Choose [Save](#).

## Create a Communication Arrangement

You create the communication arrangement as follows:

1. Open the Communication Arrangements app and choose [New](#).
2. In the [New Communication Arrangement](#) dialog, search for the [Marketing - Captcha Integration \(SAP\\_COM\\_0584\)](#) communication scenario.
3. Enter a name for the communication arrangement.
4. Choose [Create](#).
5. In the [Communication System](#) field, open the value help to search for and select the communication system you created.
6. In the [Additional Properties](#) section, verify that the correct property value is set for the type. The default is set at **10**, which is Google reCAPTCHA v3.
7. Open the settings of the Google Admin Console and copy and paste the site and secret keys into the communication arrangement.
8. Choose [Save](#).

You have set up the captcha configuration for your forms. When you create a new form and add a [Captcha](#) element, you can select the configured captchas in the [Captcha Configuration](#) list. For more information, see [Enhancing Form Security](#).

## 4.4 Application-Enabling Integrations

The section provides information about integration options that enable specific applications of SAP Marketing Cloud, such as geospatial segmentation, or analyzing marketing data based on the analytic capabilities of SAP BusinessObjects Cloud.

### [Integrating Custom Themes \[page 267\]](#)

Set a custom company theme in the SAP Fiori launchpad.

### [Integration with SAP Analytics Cloud \(ISO\) \[page 268\]](#)

Set up the Integration of SAP Analytics Cloud (ISO) with SAP Marketing Cloud

### [Content Studio Integrations \[page 304\]](#)

Lists the Content Studio integrations with SAP Marketing Cloud

### [Enabling Geospatial Segmentation with here.com \[page 317\]](#)

Use the integration option to translate addresses to geo-coordinates and reverse, and to enable geospatial analysis for segmentation based on the connected maps.

### [Integration with Baidu Maps for Geospatial Segmentation \(Deprecated\) \[page 318\]](#)

The integration of Baidu Maps into Segmentation enables you to segment contacts in China by geographic location in a visualized way.

### [SAP Jam Integration for Collaboration \[page 320\]](#)

The integration enables using SAP Jam in SAP Marketing Cloud to facilitate the collaboration when planning and executing marketing campaigns.

### [Verifying Email Addresses Using a Partner Solution \[page 321\]](#)

Accurate email addresses are vital for email marketing campaigns. To verify email addresses you can use partner services, such as Neverbounce or others.

### [Integration with an External Coupon Service System \[page 321\]](#)

Integrate SAP Marketing Cloud with an external coupon service system. To use this integration, you must use the communication arrangement: `SAP_COM_0286`.

### [Partner Extension: Integrate with Digital Market Intelligence \[page 330\]](#)

With the partner integration of SimilarWeb, you can see the web no app traffic of your competitors for each channel, such as direct, email, social. These insights help you to make better strategic decisions with regard to your own campaigns. Note that the extension is an offering from a partner of SAP.

### [Marketing Events \[page 330\]](#)

### 4.4.1 Integrating Custom Themes

Set a custom company theme in the SAP Fiori launchpad.

The *Scope Item: UI Theme Designer (2TV)* enables you to set a custom company theme for the SAP Fiori launchpad in the SAP Marketing Cloud. For example, a company color scheme and a company logo.

To integrate a custom theme in SAP Marketing Cloud, do the following:

1. Create a custom theme on your SAP Business Technology Platform (Neo) account.
2. Set up a communication arrangement in SAP Marketing Cloud that uses the `SAP_COM_0086` communication scenario.

3. Set as an SAP Marketing Cloud default theme.

For more information, see [Setting a Custom Theme for the SAP Fiori Launchpad](#) and [Scope Item: UI Theme Designer \(2TV\)](#).

#### i Note

The *Scope Item: UI Theme Designer (2TV)* is excluded from default activation. To activate the scope item, submit a request to BCP – Ticket Component: `XX-S4C-OPR-SRV`. The activation of this scope item requires an additional SAP Business Technology Platform (Neo) account.

## 4.4.2 Integration with SAP Analytics Cloud (1SO)

Set up the Integration of SAP Analytics Cloud (1SO) with SAP Marketing Cloud

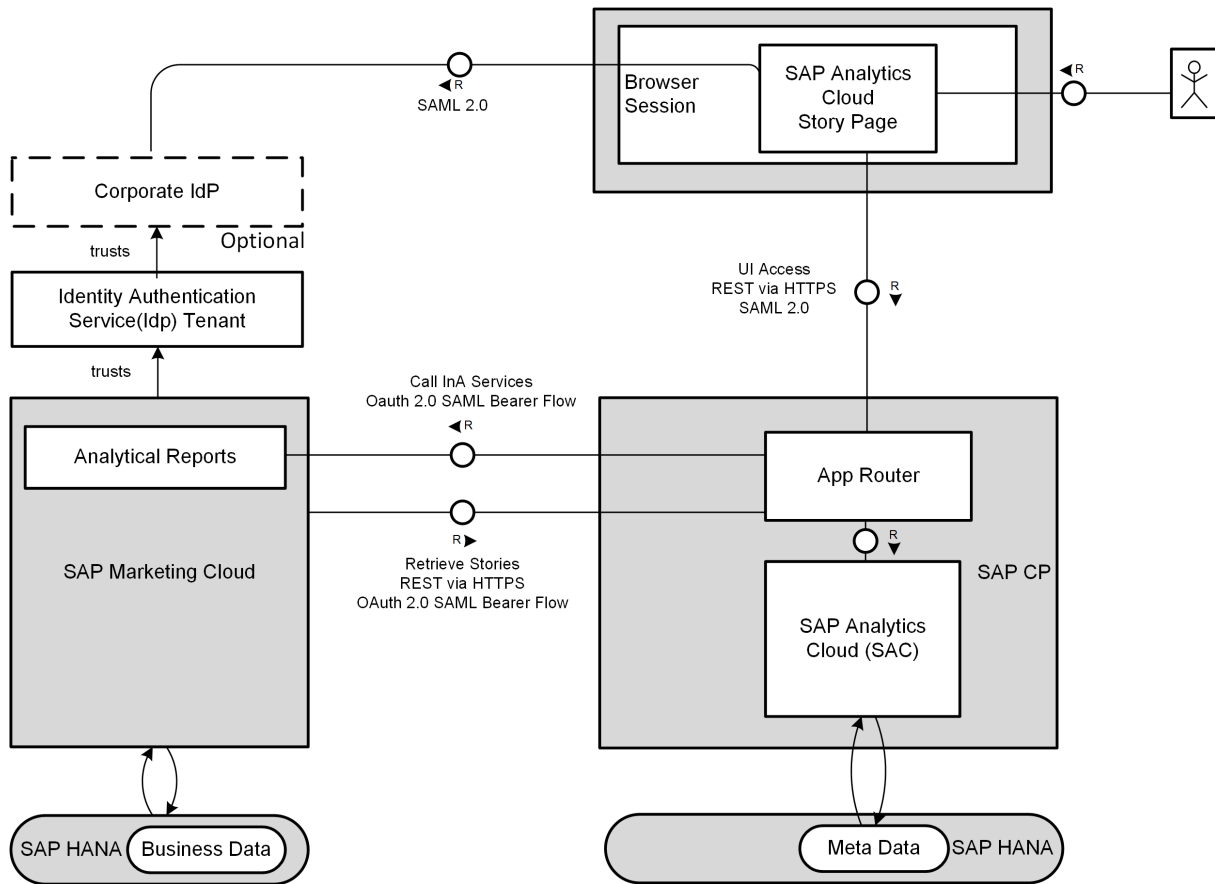
#### i Note

This section is not relevant for SAP Analytics Cloud, embedded edition. You check in the [Set Up Your Marketing Solution](#) application to see if you have the embedded edition. For more information, see [Setup of SAP Analytics Cloud, Embedded Edition/SAP Analytics Cloud](#).

## Use

Integrating SAP Marketing Cloud with SAP Analytics Cloud allows you to make full use of the analytic capabilities of SAP Analytics Cloud to explore marketing data. You can, for example, build analytics stories based on CDS views and use them in the [Analytics and Reporting Gallery](#). The Live Data Connection allows users to run SAP Marketing Cloud CDS query views without data replication. The analytical query is delivered as CDS content and is exposed via the analytical engine and the `INA` protocol via the `REST` services under `/sap/bw/ina/` to SAP Analytics Cloud.





## Prerequisites

To integrate the SAP Marketing Cloud data into SAP Analytics Cloud, ensure the following:

- You have the necessary access data (URL, user, and password, authorizations) for the SAP Cloud Identity Provider (IdP) system and the SAP Analytics Cloud system.
- Both applications must use the same SAP Cloud Identity Provider (IdP).
- The IdP user is authorized for both applications.

## More Information

### SAP Analytics Cloud

Complete the following steps to integrate SAP Analytics Cloud with SAP Marketing Cloud.

1. [Connecting SAP Analytics Cloud to SAP Marketing Cloud Identity Provider \[page 270\]](#)  
Connect your SAP Analytics Cloud to SAP Marketing Cloud Identity Provider.
2. [Create the Live Data Connection to SAP Marketing Cloud \[page 272\]](#)

Creating the live data connection allows you to run SAP Marketing Cloud CDS query views without data replication.

3. [Renewal of Signing Certificates \[page 300\]](#)

Renew signing certificates.

4. [Selecting SAP Marketing Cloud Content Packages to Add to Your Tenant \[page 301\]](#)

Choose and then import SAP Marketing Cloud content to your SAP Analytics Cloud system to improve your analytics scenarios.

## 4.4.2.1 Connecting SAP Analytics Cloud to SAP Marketing Cloud Identity Provider

Connect your SAP Analytics Cloud to SAP Marketing Cloud Identity Provider.

### Prerequisites

1. You need to have the role **System Owner** in SAP Analytics Cloud to set up the SSO integration.
2. If you have previously set up an SAML SSO live connection with the check box *Identity Provider will also be used for Live Data connections with SAML Single Sign On to S/4HANA Cloud Edition* enabled, then you need do the following steps first:
  1. In the SAP Analytics Cloud menu, select **System > Administration** and switch to the *Security* tab. Edit the settings.
  2. In the section *Authentication Method*, change the setting from *SAML Single Sign-On (SSO)* to *SAP Cloud Identity (Default)*, and save the settings. Continue with the following steps.

#### Caution

It is important to base the SAML assertion on *Login name*, in order to ensure a smooth integration. Support for other configurations is not guaranteed.

### Procedure

1. In your SAP Cloud Identity Services - Identity Authentication system, open the administration console. Go to **Application & Resources > Tenant Settings > SAML 2.0 Configuration** and download the identity provider's metadata file.
2. Start the SAP Analytics Cloud application in a separate window as *System Owner*.
3. From the menu, select **System > Administration** and switch to the *Security* tab. Edit the settings.
4. In the section *Authentication Method*, change the setting to *SAML Single Sign-On (SSO)*.
5. Go to the section *SAML Single Sign-On (SSO) Configuration*. In *Step 1: Download Service Provider metadata*, download the `metadata.xml` file.

The downloaded file initially has the same name as the metadata file of the identity provider. Make sure that you rename the metadata file.

6. In *Step 2: Upload Identity Provider metadata*, upload the `metadata.xml` file that you've previously downloaded from SAP Marketing Cloud identity provider (not the file from SAP Analytics Cloud).
7. In *Step 3: Choose a user attribute to map to your identity provider*, select *Custom SAML User Mapping*.

#### ⚠ Caution

*Custom SAML User Mapping* with the SAML assertion based on *Login name* is the only out of the box supported configuration for this step. Support for other configurations is not guaranteed.

8. Return to the SAP Cloud Identity Services - Identity Authentication system and open ► *Application & Resources* ► *Applications* ►. Add a new application for SAP Analytics Cloud.
9. On the *Trust* tab, select *SAML 2.0 Configuration*. In the section *Define from Metadata*, upload the `metadata.xml` file that you downloaded from earlier.
10. Return to the *Trust* tab. Choose the entry *Name ID Attribute* and change the setting to *Login Name*. Save your settings.
11. In the SAP Cloud Identity Services - Identity Authentication system, under ► *Users & Authorizations* ► *User Management* ►, search for the user that you want to map to your existing account. Note the login name of the user for the SAP Marketing Cloud system.
12. In the SAP Analytics Cloud system, you can now verify that all settings are correct. In the section *SAML Single Sign-On (SSO) Configuration*, in *Step 4: Verify your account with the identity provider*, provide the login name of your SAP Marketing Cloud user in the *Login Credential (Custom SAML User Mapping)* and click *Verify Account*.
13. From the upcoming popup, copy the URL.

#### i Note

Use a private session to open the URL, such as incognito mode in the Google Chrome browser. Doing so ensures that when you log on to SAP Analytics Cloud, you're prompted to log on and don't reuse an existing browser session. You will log in with the SAML\_VERIFY user.

14. In the SAP Analytics Cloud system, return to the *Security* settings page, where you now should get a message that your account has been verified. Save your settings.
15. In the popup, click *Convert* and confirm the message.  
  
After some minutes, your tenant is connected to the SAP Marketing Cloud identity provider. The SAML user mapping for your user that carried out the conversion was already changed.
16. Ensure that you adjust the SAML user mappings of all existing users manually to the new identity provider. First, ensure that you created all users in the identity provider.
17. Navigate to ► *Security* ► *Users* ►. In the column *SAML User Mapping*, enter the corresponding login name of the SAP Marketing Cloud system for all other users. Save the changes.

**Task overview:** [Integration with SAP Analytics Cloud \(1SO\) \[page 268\]](#)

**Next:** [Create the Live Data Connection to SAP Marketing Cloud \[page 272\]](#)

## 4.4.2.2 Create the Live Data Connection to SAP Marketing Cloud

Creating the live data connection allows you to run SAP Marketing Cloud CDS query views without data replication.

Before you create the live data connection to SAP Marketing Cloud, you must identify the environment where SAP Analytics Cloud is hosted. The following environments are available:

- [SAC Neo Tenant \[page 272\]](#)
- [SAP Cloud Foundry Tenant \[page 285\]](#)

If the tenant id is visible as a part of the SAP Analytics Cloud URL, then SAP Analytics Cloud is hosted on the **SAC Neo** environment. For example, <https://xxx.sapanalytics.cloud/sap/fpa/ui/tenants/<tenant ID>/app.html>

If the tenant id is **not** visible as a part of the SAP Analytics Cloud URL, then SAP Analytics Cloud is hosted on the **SAP Cloud Foundry** environment. For example, <https://xxx.sapanalytics.cloud/sap/fpa/ui/tenants/app.html>

Once you identify the environment, ensure that you complete the tasks in the sequence they're listed to create the live data connection to SAP Marketing Cloud.

**Parent topic:** [Integration with SAP Analytics Cloud \(1SO\) \[page 268\]](#)

**Previous task:** [Connecting SAP Analytics Cloud to SAP Marketing Cloud Identity Provider \[page 270\]](#)

**Next:** [Renewal of Signing Certificates \[page 300\]](#)

### 4.4.2.2.1 SAC Neo Tenant

Identify if the SAC Neo tenant is the environment where SAP Analytics Cloud is hosted.

If the tenant id is visible as a part of the SAP Analytics Cloud URL, then SAP Analytics Cloud is hosted on the SAC Neo environment. For example, <https://xxx.sapanalytics.cloud/sap/fpa/ui/tenants/<tenant ID>/app.html>

Now that you've identified the environment where SAP Analytics Cloud is hosted, to create the live data connection with the SAC Neo tenant, complete the tasks in the following sequence:

1. [Adding a New OAuth Client \[page 273\]](#)
2. [Adding a New Live Data Connection \[page 274\]](#)
3. [Setting Up a Communication System \[page 275\]](#)
4. [Setting Up a Communication Arrangement \[page 277\]](#)
5. [Completing the Setup of the Live Data Connection \[page 280\]](#)
6. [Setting Up the Outbound Connections from SAP Marketing Cloud to SAP Analytics Cloud \[page 282\]](#)
7. [Adding SAP Marketing Cloud as a Trusted Origin with SAP Analytics Cloud \[page 283\]](#)

Allow the use of and then access the analytics stories in SAP Marketing Cloud.

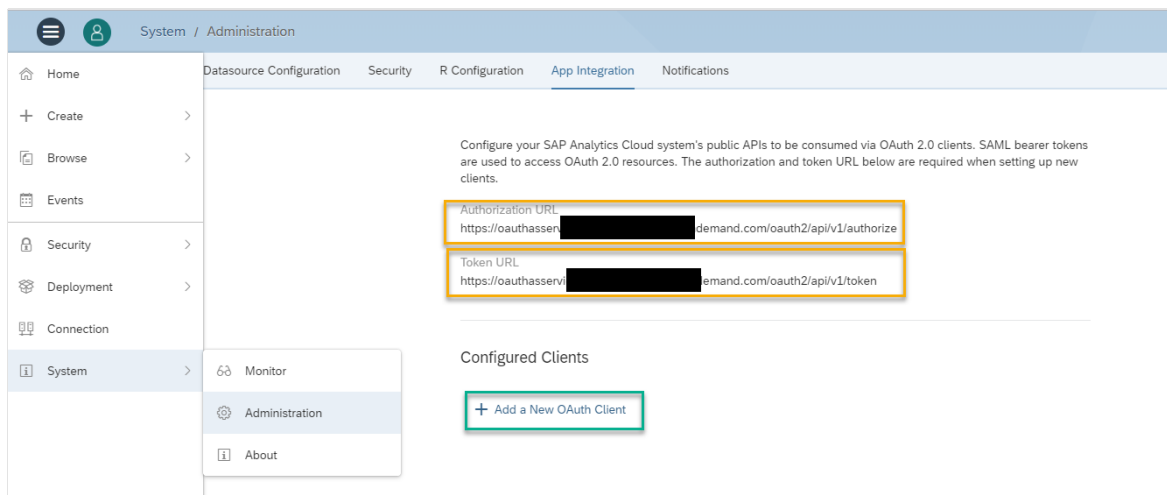
## 4.4.2.2.1.1 Adding a New OAuth Client

### Prerequisites

For the steps in the SAP Analytics Cloud system, the *Admin* or *System Owner* role is required.

### Procedure

1. In the SAP Analytics Cloud system, navigate to **System > Administration > App Integration > OAuth Clients** and note down the authorization URL and the token URL.



2. Under *Configured Clients*, add a new OAuth Client with the following properties:
  1. *Name*, for example "SAP HMC System"
  2. *OAuth Client ID*, for example "my300xxx".
  3. *Purpose*: "Interactive Usage".
  4. **Security > Authorization Grant** Client Credentials
  5. **Security > Secret** chosen by you (like a password). Don't specify a lifetime for the secret.
  6. **Token Details > Token Lifetime** 60 minutes (suggested value).
3. Press *Add*.

The screenshot shows the 'New OAuth Client' configuration form. It contains the following fields and options:

- Name:** SAP SMC System
- OAuth Client ID:** my [redacted]
- Purpose:** Interactive Usage
- Security:**
  - Authorization Grant: Client Credentials
  - Confidential
  - Secret:** [masked]
  - Lifetime:** [ ] days
- Token Details:**
  - Token Lifetime: 1 hours

Buttons: Add, Cancel

**Task overview:** [SAC Neo Tenant \[page 272\]](#)

**Next task:** [Adding a New Live Data Connection \[page 274\]](#)

## 4.4.2.2.1.2 Adding a New Live Data Connection

### Procedure

1. In the SAP Analytics Cloud system, navigate to *Connection* and press the Plus (+) icon to add a new connection.
2. Select **Live Data Connection** **SAP S/4HANA**.
3. In the dialog *New S/4HANA Live Connection*, enter a *Name* and *Description*.  
In order to use SAP Marketing Cloud sample content, the name has to be **SAPMKTNW**.
4. Select *Connection Type* **SAP S/4HANA Cloud**.
5. Under *Host*, enter the host name of the SAP Marketing Cloud tenant. For example, `my300xxx-api.s4hana.ondemand.com`.
6. Note down the provider name that is displayed under **Authentication Method** **OAuth 2 SAML Bearer Assertion** and download the signing certificate.

## i Note

No need to save at this point. We make the required configurations in SAP Marketing Cloud and return to this setup for saving.

**Task overview:** [SAC Neo Tenant \[page 272\]](#)

**Previous task:** [Adding a New OAuth Client \[page 273\]](#)

**Next task:** [Setting Up a Communication System \[page 275\]](#)

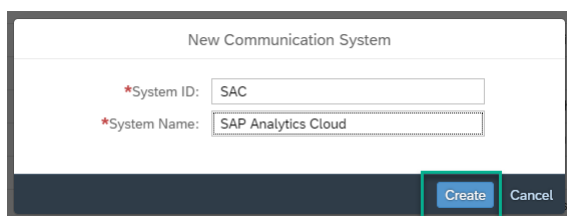
## 4.4.2.2.1.3 Setting Up a Communication System

### Prerequisites

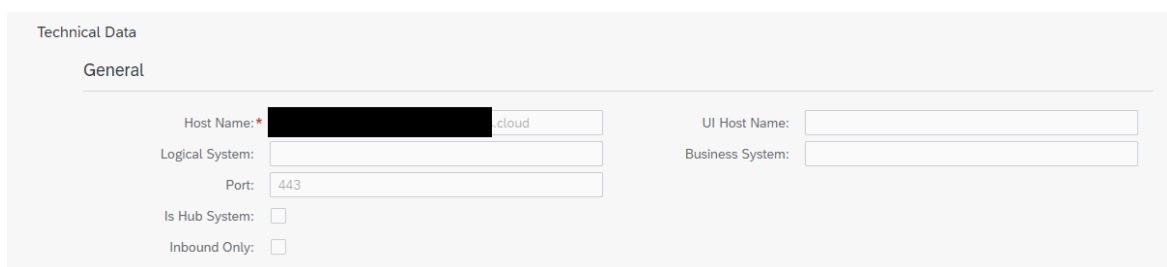
For the steps in the SAP Marketing Cloud system, the *Administrator* role is required.

### Procedure

1. Open SAP Marketing Cloud in a new browser window. In the app *Communication Systems*, click *New* to create a new communication system.



2. Under **► Technical Data ► General ► Host Name**, enter the host name of the SAP Analytics Cloud tenant.



3. Under **► Technical Data ► OAuth 2.0 Settings**, enter the authorization endpoint `oauthservices-  
<SAP CP account ID>.int.sap.hana.ondemand.com/oauth2/api/v1/authorize` and token

endpoint `oauthservices-<SAP CP account ID>.int.sap.hana.ondemand.com/oauth2/api/vi/token` when you added the new OAuth client. For more information, see [Adding a New OAuth Client \[page 273\]](#).

### i Note

Enter these endpoints without the `https://` prefix.

OAuth 2.0 Settings

Auth. Endpoint:  Token Endpoint:

- Under *OAuth 2.0 Identity Provider*, select *Enabled*. Enter the provider name and upload the signing certificate that you obtained in [Adding a New Live Data Connection \[page 274\]](#).

OAuth 2.0 Identity Provider

Enabled:

\*Provider Name:

Signing Certificate Subject: CN=https://account.hana.ondemand.com/...

Signing Certificate Issuer: CN=https://account.hana.ondemand.com/...

- Create a user for Inbound Communication with the *Authentication Method* *User Name and Password* and note down the user name and password.

New Inbound Communication User

\*User Name:

\*Authentication Method:

Maintain User  OK Cancel

SAP Analytics Cloud

General Data

\*System ID:  Notes:

\*System Name:

Technical Data

General

\*Host Name:  UI Host Name:

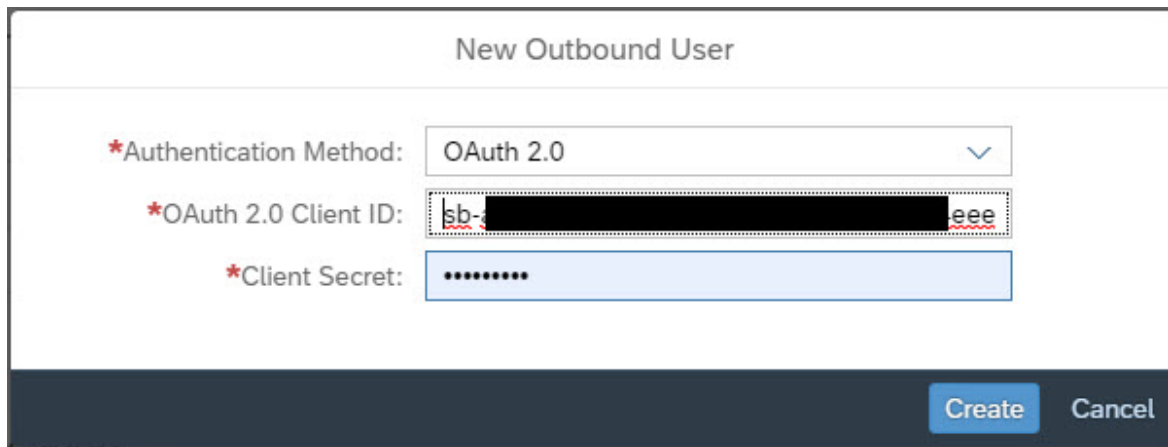
Logical System:  Business System:

HTTPS Port:  Use Cloud Connector:

- To create a user for Outbound Communication, enter the following details:
  - In *Authentication Method*, choose *OAuth 2.0*.



2. Add the OAuth 2.0 client ID and the client secret that you defined in [Adding a New OAuth Client \[page 273\]](#).



7. Save the communication system.

**Task overview:** [SAC Neo Tenant \[page 272\]](#)

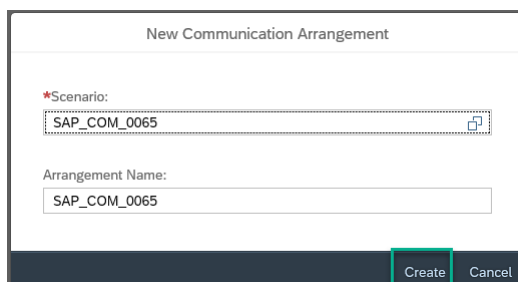
**Previous task:** [Adding a New Live Data Connection \[page 274\]](#)

**Next task:** [Setting Up a Communication Arrangement \[page 277\]](#)

## 4.4.2.2.1.4 Setting Up a Communication Arrangement

### Procedure

1. In the SAP Marketing Cloud system, choose *Communication Arrangements* app, click *New* to create a new arrangement.
2. In the *New Communication Arrangement* pop-up screen, enter the scenario **SAP\_COM\_0065** and click *Create*.



3. Enter the *Communication System* that you defined in [Setting Up a Communication System \[page 275\]](#).

SAP\_COM\_0065

Scenario ID: SAP\_COM\_0065  
Scenario Description: SAP Analytics Cloud Integration

---

Common Data

Arrangement Name:  Own System:

\*Communication System:

- Under **Additional Properties** > **Tenant ID (SAP Analytics Cloud tenant)**, maintain the tenant ID that is visible in the URL of SAP Analytics Cloud. When calling up `https://xxx.sapanalytics.cloud`, you'll be redirected to the full URL where you can find the tenant ID as shown in this example: `https://xxx.sapanalytics.cloud/sap/fpa/ui/tenants/«tenant ID»/app.html`.

SAP\_COM\_0065

Scenario ID: SAP\_COM\_0065  
Scenario Description: SAP Analytics Cloud Integration

---

Common Data

Arrangement Name:  Own System:

\*Communication System:

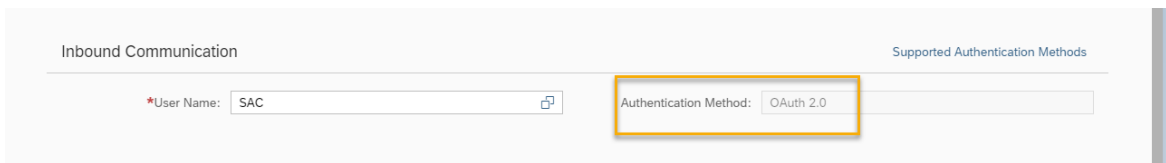
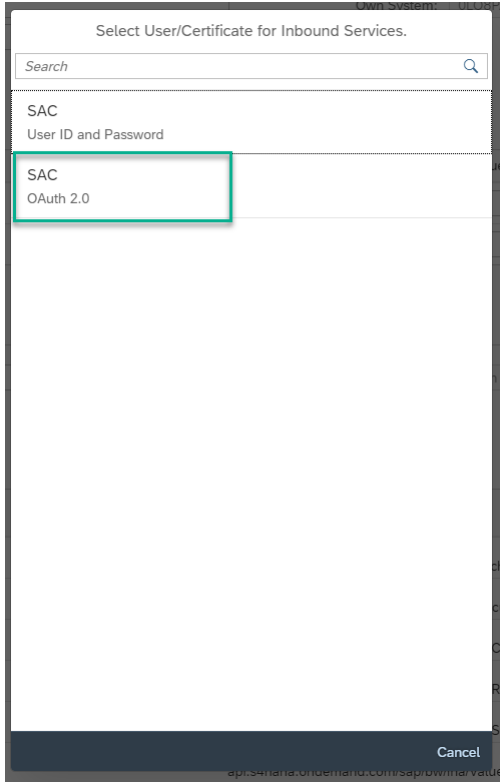
Additional Properties

Property Name	Property Value
Tenant ID	<input type="text" value=""/>
System Alias	<input type="text" value=""/>

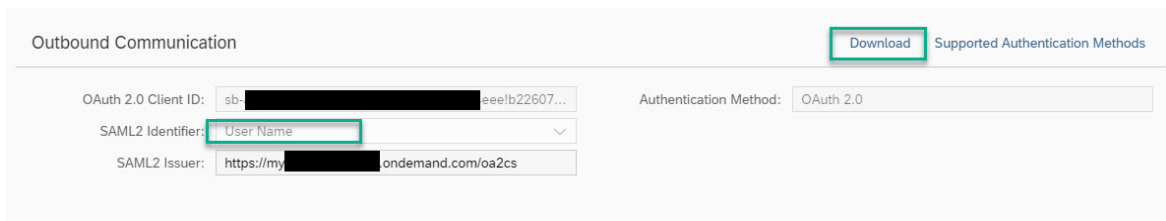
- Under **Inbound Communication** > **User Name** > **Authentication Method**, select **Authentication** with **OAuth 2.0** using the input help of the **User Name** field.

Inbound Communication

\*User Name:



- Under *Outbound Communication*, set *SAML2 Identifier* to *User Name*. Note down the *SAML2 Issuer*. For example, <https://my300xxx-api.s4hana.ondemand.com/oa2cs>. Download the signing certificate (the text file `signing_pse.crt`).



- Ensure that under *Outbound Services*, both *UI Link Navigation* and *Retrieve Stories* have *Service Status* checked (= *Active*).

Outbound Services

▼ UI Link Navigation

Service Status:  Active

Application Protocol: UI Link

Port: 443

Path: /

Service URL: http://[redacted]

---

▼ Retrieve Stories

Service Status:  Active

Application Protocol: REST Service

Port: 443

Path: /

Service URL: http://[redacted]

Draft saved Save Cancel

8. Save the communication arrangement.

**Task overview:** [SAC Neo Tenant \[page 272\]](#)

**Previous task:** [Setting Up a Communication System \[page 275\]](#)

**Next task:** [Completing the Setup of the Live Data Connection \[page 280\]](#)

## 4.4.2.2.1.5 Completing the Setup of the Live Data Connection

### Procedure

1. Switch back to the browser window with the SAP Analytics Cloud Live Data connection definition.
  1. Enter the token service user and token service password that you defined in [Setting Up a Communication System \[page 275\]](#).
  2. Enter the following space-separated list as OAuth scope:
    - SAP\_BW\_INA\_BATCHPROCESSING\_HTTP
    - SAP\_BW\_INA\_GETCATALOG\_HTTP
    - SAP\_BW\_INA\_GETRESPONSE\_HTTP
    - SAP\_BW\_INA\_GETSERVERINFO\_HTTP
    - SAP\_BW\_INA\_LOGOFF\_HTTP
    - SAP\_BW\_INA\_VALUEHELP\_HTTP
2. Click OK.

New S/4HANA Live Connection

\*Name:

Description:

**Datasource Configuration**

Additional components or configuration may be required for this connection type. See our [Help Center](#) to find out what's required.

**Connection Details**

Connection Type:

\*Host:

Default Language:

**Credentials**

Authentication Method:

**OAuth 2.0 Identity Provider Configuration**

Copy the Provider Name below and download the signing certificate. You will need this information to configure your S/4HANA Cloud system.

Provider Name:

\*Token Service User:

\*Token Service Password:

OAuth Scope:

**Task overview:** [SAC Neo Tenant \[page 272\]](#)

**Previous task:** [Setting Up a Communication Arrangement \[page 277\]](#)

**Next task:** [Setting Up the Outbound Connections from SAP Marketing Cloud to SAP Analytics Cloud \[page 282\]](#)

## 4.4.2.2.1.6 Setting Up the Outbound Connections from SAP Marketing Cloud to SAP Analytics Cloud

### Procedure

1. In the SAP Analytics Cloud system, select **System > Administration > App Integration > Configured Clients**.
2. To add a trusted identity provider, enter the following details:
  1. A name chosen by you.
  2. A provider name that is equal to the SAML 2 Issuer and to the signing certificate obtained in [Setting Up a Communication Arrangement \[page 277\]](#).
  3. The contents of the text file signing\_pse.crt into *Signing Certificate*.

Trusted Identity Providers

OAuth clients that utilize a server to server communication require a trusted identity provider to be added to establish a connection.

+ Add a Trusted Identity Provider

The screenshot shows a dialog box titled "Add a Trusted Identity Provider". It contains three required fields, each marked with a red asterisk:

- \*Name:** The input field contains "SAP SMC".
- \*Provider Name:** The input field contains "https://my[redacted]ondemand.com/oa2cs".
- \*Signing Certificate \*:** The input field is empty.

At the bottom right of the dialog, there are two buttons: "Add" and "Cancel".

**Task overview:** [SAC Neo Tenant \[page 272\]](#)

**Previous task:** [Completing the Setup of the Live Data Connection \[page 280\]](#)

**Next task:** [Adding SAP Marketing Cloud as a Trusted Origin with SAP Analytics Cloud \[page 283\]](#)

## 4.4.2.2.1.7 Adding SAP Marketing Cloud as a Trusted Origin with SAP Analytics Cloud

Allow the use of and then access the analytics stories in SAP Marketing Cloud.

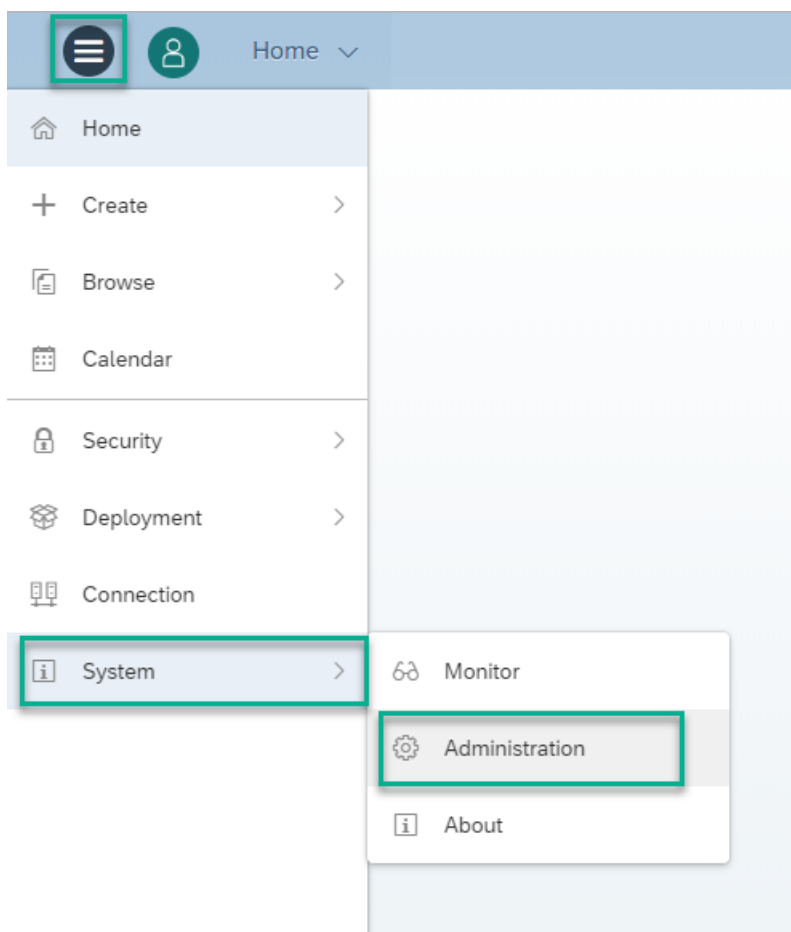
### Context

Analytics stories use charts, visualizations, texts, and pictograms to describe data.

Before you can view your SAP Analytics Cloud stories, you must first add your SAP Marketing Cloud system as a trusted origin by adding the host name of the connected SAP Marketing Cloud system to SAP Analytics Cloud.

### Procedure

1. Log on to SAP Analytics Cloud and select **System** > **Administration**.



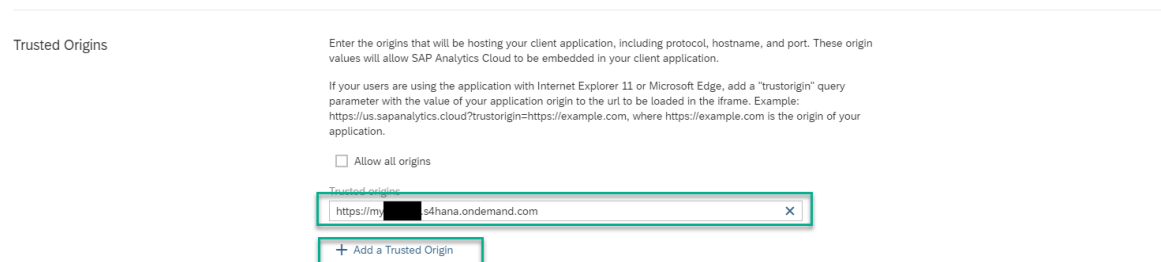
- Navigate to the tab *App Integration*.



#### OAuth Clients

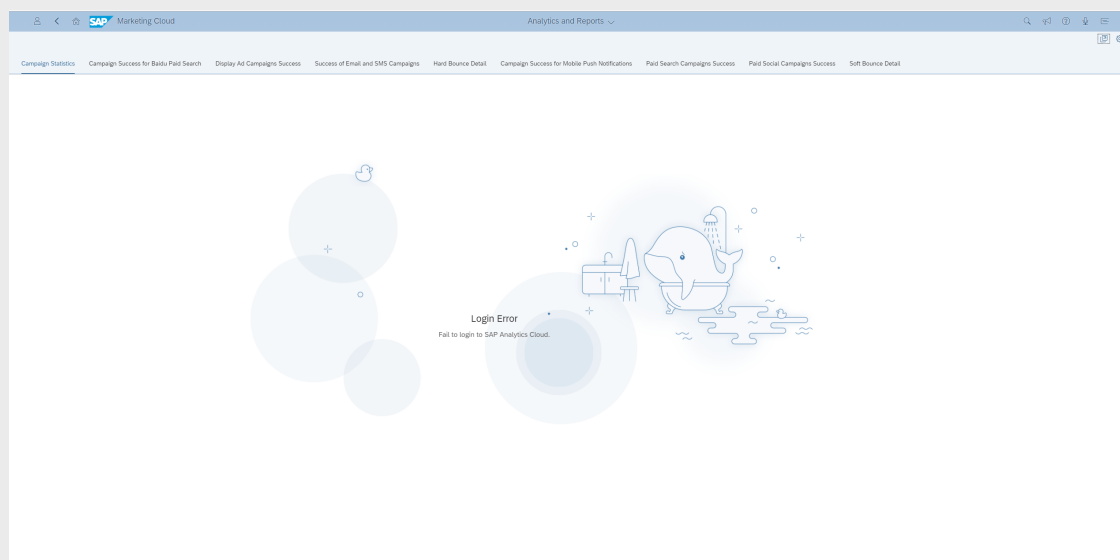
Configure your SAP Analytics Cloud system's public APIs to be consumed via OAuth 2.0 clients. SAML bearer tokens are used to access OAuth 2.0 resources. The authorization and token URL below are required when setting up new clients.

- In the *Trusted Origins* section, click *Add a Trusted Origin*.
- Enter the host name of the connected SAP Marketing Cloud system. For example, **https://myXXXX-api.s4hana.ondemand.com**.
- Click *Save*.



### Note

If the third-party cookie isn't enabled in your browser, you could get the following logon error after clicking *Analyze* to display the analytics stories in SAP Marketing Cloud.



To resolve this error, go to your Chrome browser, open **Advanced Settings** **Privacy and Security** **Content settings** **Cookies**. Add **sapbusinessobjects.cloud** and **sapanalytics.cloud** to the allowed list of third-party cookies.



Clear your browser cache and log in again. You can now see the analytics stories when you click [Analyze](#).

You can click [Analyze](#), for example in [Campaigns](#) and [Spotlighting Accounts](#), to find the analytics stories.

**Task overview:** [SAC Neo Tenant \[page 272\]](#)

**Previous task:** [Setting Up the Outbound Connections from SAP Marketing Cloud to SAP Analytics Cloud \[page 282\]](#)

## Related Information

[Analytics Stories](#)

### 4.4.2.2.2 SAP Cloud Foundry Tenant

Identify if the SAP Cloud Foundry tenant is the environment where SAP Analytics Cloud is hosted.

If the tenant id is **not** visible as a part of the SAP Analytics Cloud URL, then SAP Analytics Cloud is hosted on the SAP Cloud Foundry environment. For example, <https://xxx.sapanalytics.cloud/sap/fpa/ui/tenants/app.html>

Now that you've identified the environment where SAP Analytics Cloud is hosted, create the live data connection with SAP Cloud Foundry. Complete the tasks in the following sequence:

1. [Adding a New OAuth Client \[page 286\]](#)
2. [Adding a New Live Data Connection \[page 288\]](#)
3. [Setting Up a Communication System \[page 289\]](#)
4. [Setting Up a Communication Arrangement \[page 292\]](#)
5. [Completing the Setup of the Live Data Connection \[page 295\]](#)
6. [Setting Up the Outbound Connections from SAP Marketing Cloud to SAP Analytics Cloud \[page 297\]](#)
7. [Adding SAP Marketing Cloud as a Trusted Origin with SAP Analytics Cloud \[page 298\]](#)

Allow the use of and then access the analytics stories in SAP Marketing Cloud.

## 4.4.2.2.1 Adding a New OAuth Client

### Prerequisites

For the steps in the SAP Analytics Cloud system, the *Admin* or *System Owner* role is required.

### Procedure

1. In the SAP Analytics Cloud system, navigate to ► *System* ► *Administration* ► *App Integration* ► *OAuth Clients* ► and note down the authorization URL, oAuth2SAML Token URL, and oAuth2SAML Audience.

The screenshot shows the SAP Analytics Cloud Administration console. The breadcrumb navigation is System / Administration. The main menu includes System Configuration, Datasource Configuration, Security, R Configuration, App Integration (selected), Notifications, and Appearance. The page title is OAuth Clients. A text block explains that OAuth authorization and token services are available from the below URLs. The following URLs are listed and highlighted with yellow boxes:

- Authorization URL: `https://[redacted].authentication.eu10.hana.ondemand.com/oauth/authorize`
- Token URL: `https://[redacted].authentication.eu10.hana.ondemand.com/oauth/token`
- OAuth2SAML Token URL: `https://[redacted].authentication.eu10.hana.ondemand.com/oauth/token/alias/[redacted].aws-live-eu10`
- OAuth2SAML Audience: `[redacted].aws-live-eu10`

2. Under *Configured Clients*, add a new OAuth Client with the following properties:
  1. *Name*, for example "SAP SMC System"
  2. The *OAuth Client ID* will generate upon saving.
  3. *Purpose*: "Interactive Usage".
  4. The *Security Secret* will auto-generate.
3. Press *Add*.

## New OAuth Client

\*Name

SAP Marketing Cloud Q System

OAuth Client ID

*Client ID will be generated once the OAuth client is created.*

Purpose

Interactive Usage

### Security

Secret

*Secret will be generated once the OAuth client is created.*

Redirect URI

### Token Details

Token Lifetime

1 hour

Refresh Token Lifetime

720 hours

Add Cancel

- Once the oAuth Client is created, note down the *OAuth Client ID* and *Secret* (click on the *Show Secret* button to note down the generated secret).



## Edit OAuth Client

**\*Name**

my [REDACTED]

**OAuth Client ID**

sb-[REDACTED]eeeb22607|client!b3650

**Purpose**

Interactive Usage

**Security**

**Secret**

[REDACTED]

Show secret

**Redirect URI**

-

**Token Details**

**Token Lifetime**

1 hour

**Refresh Token Lifetime**

720 hours

Save Cancel

**Task overview:** [SAP Cloud Foundry Tenant \[page 285\]](#)

**Next task:** [Adding a New Live Data Connection \[page 288\]](#)

### 4.4.2.2.2 Adding a New Live Data Connection

#### Procedure

1. In the SAP Analytics Cloud system, navigate to *Connection* and press the Plus (+) icon to add a new connection.

2. Select **Live Data Connection** > **SAP S/4HANA**.
3. In the dialog *New S/4HANA Live Connection*, enter a *Name* and *Description*.  
In order to use SAP Marketing Cloud sample content, the name has to be **SAPMKTNW**.
4. Select *Connection Type* **SAP S/4HANA Cloud**.
5. Under *Host*, enter the host name of the SAP Marketing Cloud tenant. For example, `my300xxx-api.s4hana.ondemand.com`.
6. Note down the provider name that is displayed under **Authentication Method** > *OAuth 2 SAML. Bearer Assertion* and download the signing certificate.

#### i Note

No need to save at this point. We make the required configurations in SAP Marketing Cloud and return to this setup for saving.

**Task overview:** [SAP Cloud Foundry Tenant \[page 285\]](#)

**Previous task:** [Adding a New OAuth Client \[page 286\]](#)

**Next task:** [Setting Up a Communication System \[page 289\]](#)

## 4.4.2.2.3 Setting Up a Communication System

### Prerequisites

For the steps in the SAP Marketing Cloud system, the *Administrator* role is required.

### Procedure

1. Open SAP Marketing Cloud in a new browser window. In the app *Communication Systems*, click *New* to create a new communication system.

The screenshot shows a dialog box titled "New Communication System". It contains two input fields:
 

- \*System ID: SAC
- \*System Name: SAP Analytics Cloud

 At the bottom right, there are two buttons: "Create" (highlighted with a red box) and "Cancel".

- Under **Technical Data** > **General** > **Host Name**, enter the host name of the SAP Analytics Cloud tenant.

The screenshot shows the configuration page for SAP Analytics Cloud. It is divided into two main sections: General Data and Technical Data. In the General Data section, the System ID is set to 'SAC' and the System Name is 'SAP Analytics Cloud'. In the Technical Data section, under the General sub-section, the Host Name is partially filled with '...s.cloud', the Logical System and Business System fields are empty, and the HTTPS Port is set to '443'. There is also a checkbox for 'Use Cloud Connector' which is currently unchecked.

- Under **Technical Data** > **OAuth 2.0 Settings**, enter the authorization endpoint, oAuth2SAML Token endpoint, and oAuth2SAML Audience from when you added the new OAuth client. For more information, see [Adding a New OAuth Client \[page 286\]](#).

### Note

Enter the authorization endpoint and the oAuth2SAML Token endpoint without the `https://` prefix.

The oAuth2SAML Audience must be entered exactly as it appears in the SAP Analytics Cloud systems.

The screenshot shows the OAuth 2.0 Settings configuration page. It includes fields for 'Auth. Endpoint', 'Token Endpoint', and 'Audience'. The 'Auth. Endpoint' is filled with a partially obscured URL ending in 'authentication.eu10.hana.ondema...'. The 'Token Endpoint' is filled with a URL starting with '.com/oauth/token/alias' followed by a partially obscured domain and 'aws-live-eu10'. The 'Audience' field is filled with 'aws-live-eu10'.

- Under **OAuth 2.0 Identity Provider** select **Enabled**. Enter the provider name and upload the signing certificate that you obtained in [Adding a New Live Data Connection \[page 288\]](#).

The screenshot shows the OAuth 2.0 Identity Provider configuration page. The 'Enabled' checkbox is checked. The 'Provider Name' field is filled with a partially obscured name. To the right of the Provider Name field is a button labeled 'Upload Signing Certificate'. Below this, the 'Signing Certificate Subject' and 'Signing Certificate Issuer' fields are both filled with partially obscured text starting with 'CN=https://account.hana.ondemand.com'.

- Create a Token Service User for Inbound Communication with the **Authentication Method** User Name and Password and note down the user name and password.

New Inbound Communication User

\*User Name:

\*Authentication Method: **User Name and Password**

Maintain User **New User** OK Cancel

SAP Analytics Cloud

General Data

\*System ID:  Notes:

\*System Name:

Technical Data

General

\*Host Name:  UI Host Name:

Logical System:  Business System:

HTTPS Port:  Use Cloud Connector:

6. To create a user for Outbound Communication, enter the following details:
  1. In *Authentication Method*, add **OAuth 2.0**.
  2. Add the OAuth 2.0 client ID and the client secret that you defined in [Adding a New OAuth Client \[page 286\]](#).

New Outbound User

\*Authentication Method: **OAuth 2.0**

\*OAuth 2.0 Client ID:

\*Client Secret:

Create Cancel

7. Save the communication system.

**Task overview:** [SAP Cloud Foundry Tenant \[page 285\]](#)

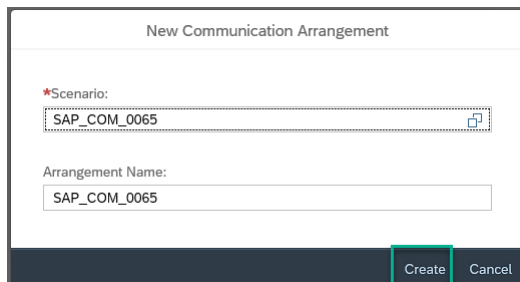
**Previous task:** [Adding a New Live Data Connection \[page 288\]](#)

**Next task:** [Setting Up a Communication Arrangement \[page 292\]](#)

## 4.4.2.2.2.4 Setting Up a Communication Arrangement

### Procedure

1. In the SAP Marketing Cloud system app *Communication Arrangements*, click *New* to create a new arrangement.
2. Enter the *Scenario SAP\_COM\_0065* and click *Create*.



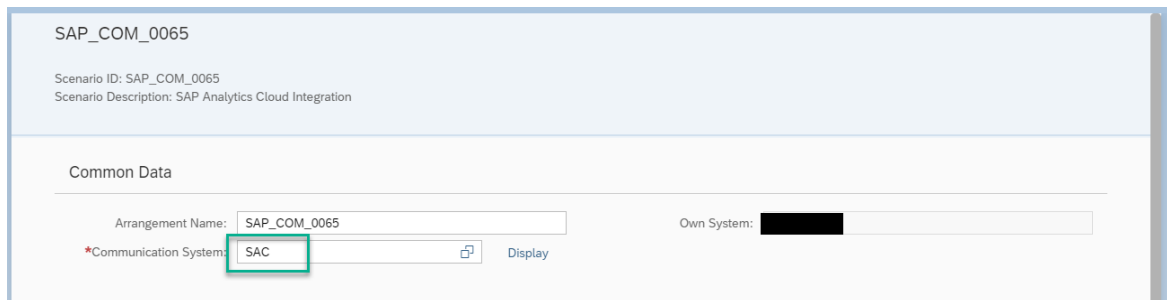
New Communication Arrangement

\*Scenario:  
SAP\_COM\_0065

Arrangement Name:  
SAP\_COM\_0065

Create Cancel

3. Enter the *Communication System* that you defined in [Setting Up a Communication System \[page 289\]](#).



SAP\_COM\_0065

Scenario ID: SAP\_COM\_0065  
Scenario Description: SAP Analytics Cloud Integration

Common Data

Arrangement Name: SAP\_COM\_0065 Own System: [REDACTED]

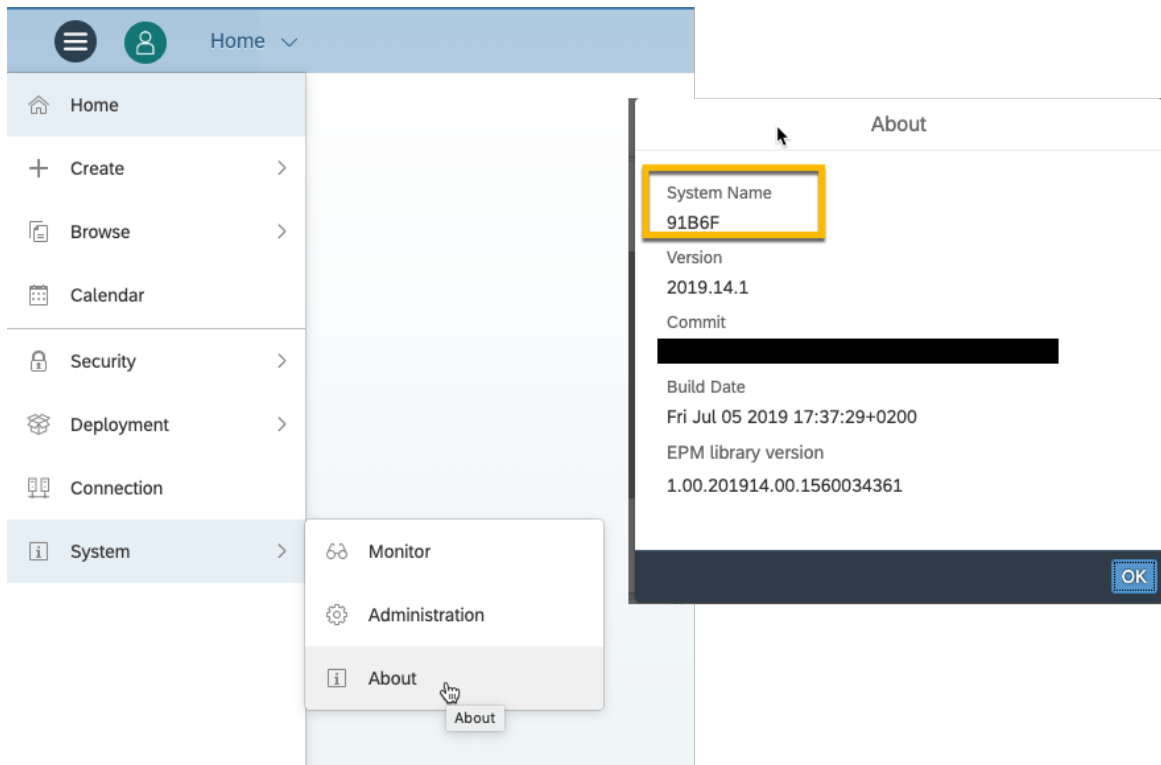
\*Communication System: SAC Display

4. Under **Additional Properties** > *Tenant ID (SAP Analytics Cloud tenant)*, maintain the SAP Analytics Cloud tenant ID. In the SAP Analytics Cloud system, you can find the tenant ID under **Menu** > *System* > *About*. The value displayed under *System Name* is the tenant ID.

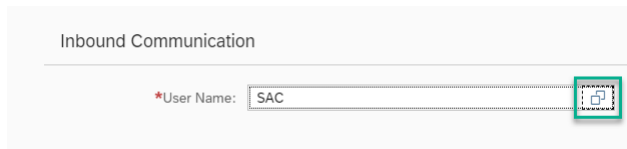
#### i Note

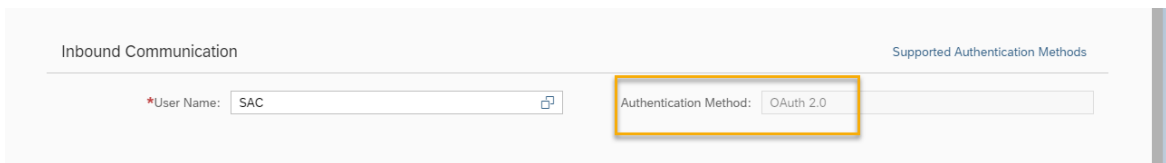
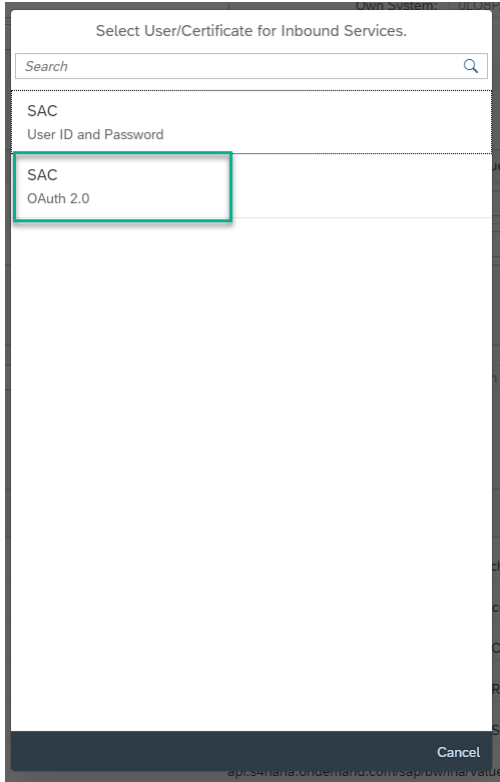
Please maintain the alpha part of the value in lower case. Even if you see the system name as "91B6F", the right value to enter is "91b6f".



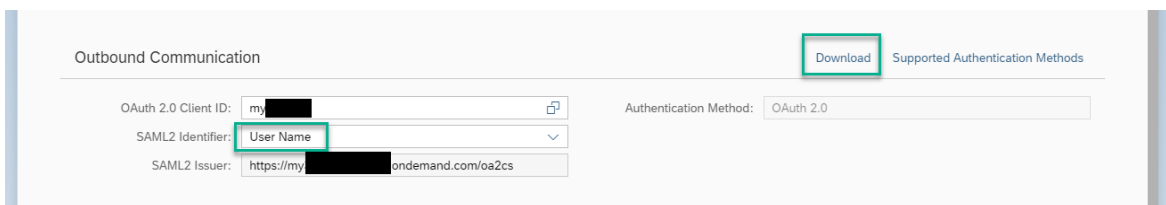


- Under **Inbound Communication** > **User Name** > **Authentication Method**, select Authentication with OAuth 2.0 using the input help of the **User Name** field.





- Under *Outbound Communication*, set *SAML2 Identifier* to *User Name*. Note down the *SAML2 Issuer*. For example, `https://my300xxx-api.s4hana.ondemand.com/oa2cs`. Download the signing certificate (the text file `signing_pse.crt`).



- Ensure that under *Outbound Services*, both *UI Link Navigation* and *Retrieve Stories* have *Service Status* checked (= *Active*).

Outbound Services

▼ UI Link Navigation

Service Status:  Active

Application Protocol: UI Link

Port: 443

Path: /

Service URL: http://[redacted]

---

▼ Retrieve Stories

Service Status:  Active

Application Protocol: REST Service

Port: 443

Path: /

Service URL: http://[redacted]

Draft saved

8. Save the communication arrangement.

**Task overview:** [SAP Cloud Foundry Tenant \[page 285\]](#)

**Previous task:** [Setting Up a Communication System \[page 289\]](#)

**Next task:** [Completing the Setup of the Live Data Connection \[page 295\]](#)

## 4.4.2.2.5 Completing the Setup of the Live Data Connection

### Procedure

1. Switch back to the browser window with the SAP Analytics Cloud Live Data connection definition.
  1. Enter the token service user and token service password that you defined in [Setting Up a Communication System \[page 275\]](#).
  2. Enter the following space-separated list as OAuth scope:
    - SAP\_BW\_INA\_BATCHPROCESSING\_HTTP
    - SAP\_BW\_INA\_GETCATALOG\_HTTP
    - SAP\_BW\_INA\_GETRESPONSE\_HTTP
    - SAP\_BW\_INA\_GETSERVERINFO\_HTTP
    - SAP\_BW\_INA\_LOGOFF\_HTTP
    - SAP\_BW\_INA\_VALUEHELP\_HTTP
2. Click OK.

New S/4HANA Live Connection

\*Name:

Description:

**Datasource Configuration**

*Additional components or configuration may be required for this connection type. See our [Help Center](#) to find out what's required.*

**Connection Details**

Connection Type:

\*Host:

Default Language:

**Credentials**

Authentication Method:

**OAuth 2.0 Identity Provider Configuration**

*Copy the Provider Name below and download the signing certificate. You will need this information to configure your S/4HANA Cloud system.*

Provider Name:

[Download Signing Certificate](#)

\*Token Service User:

\*Token Service Password:

OAuth Scope:

**Task overview:** [SAP Cloud Foundry Tenant \[page 285\]](#)

**Previous task:** [Setting Up a Communication Arrangement \[page 292\]](#)

**Next task:** [Setting Up the Outbound Connections from SAP Marketing Cloud to SAP Analytics Cloud \[page 297\]](#)

## 4.4.2.2.6 Setting Up the Outbound Connections from SAP Marketing Cloud to SAP Analytics Cloud

### Procedure

1. In the SAP Analytics Cloud system, select **System > Administration > App Integration > Configured Clients**.
2. To add a trusted identity provider, enter the following details:
  1. A name chosen by you.
  2. A provider name that is equal to the SAML 2 Issuer and to the signing certificate obtained in [Setting Up a Communication Arrangement \[page 277\]](#).
  3. The contents of the text file signing\_pse.crt into *Signing Certificate*.

Trusted Identity Providers

OAuth clients that utilize a server to server communication require a trusted identity provider to be added to establish a connection.

+ Add a Trusted Identity Provider

Add a Trusted Identity Provider

\*Name  
SAP SMC

\*Provider Name  
https://my[redacted]ondemand.com/oa2cs

\*Signing Certificate \*

Add Cancel

**Task overview:** [SAP Cloud Foundry Tenant \[page 285\]](#)

**Previous task:** [Completing the Setup of the Live Data Connection \[page 295\]](#)

**Next task:** [Adding SAP Marketing Cloud as a Trusted Origin with SAP Analytics Cloud \[page 298\]](#)

## 4.4.2.2.7 Adding SAP Marketing Cloud as a Trusted Origin with SAP Analytics Cloud

Allow the use of and then access the analytics stories in SAP Marketing Cloud.

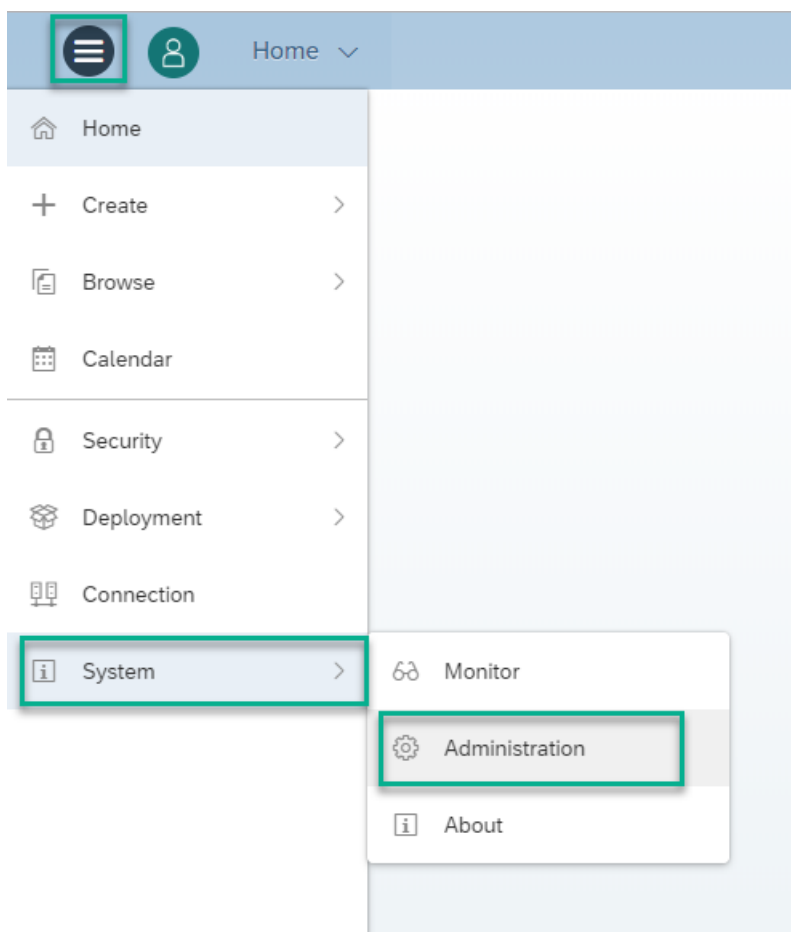
### Context

Analytics stories use charts, visualizations, texts, and pictograms to describe data.

Before you can view your SAP Analytics Cloud stories, you must first add your SAP Marketing Cloud system as a trusted origin by adding the host name of the connected SAP Marketing Cloud system to SAP Analytics Cloud.

### Procedure

1. Log on to SAP Analytics Cloud and select **System** > **Administration**.



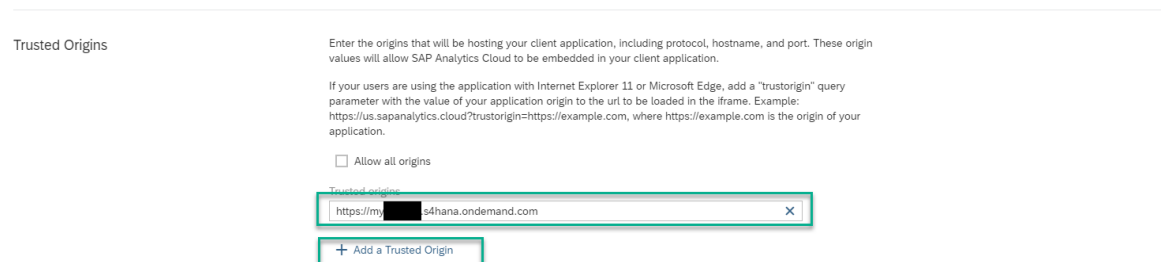
- Navigate to the tab *App Integration*.



#### OAuth Clients

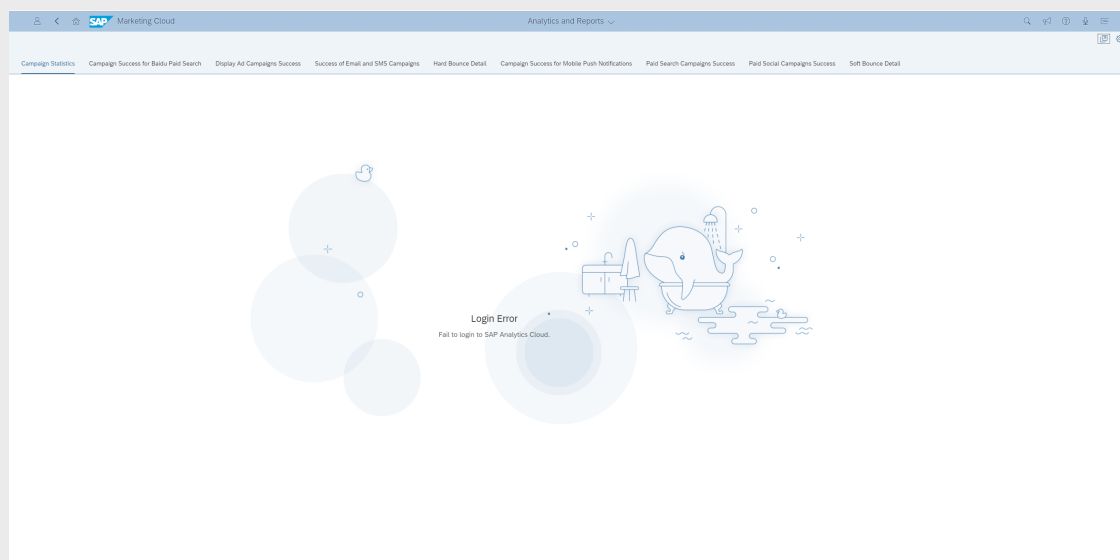
Configure your SAP Analytics Cloud system's public APIs to be consumed via OAuth 2.0 clients. SAML bearer tokens are used to access OAuth 2.0 resources. The authorization and token URL below are required when setting up new clients.

- In the *Trusted Origins* section, click *Add a Trusted Origin*.
- Enter the host name of the connected SAP Marketing Cloud system. For example, **https://myXXXX-api.s4hana.ondemand.com**.
- Click *Save*.



### Note

If the third-party cookie isn't enabled in your browser, you could get the following logon error after clicking *Analyze* to display the analytics stories in SAP Marketing Cloud.



To resolve this error, go to your Chrome browser, open **Advanced Settings** **Privacy and Security** **Content settings** **Cookies**. Add **sapbusinessobjects.cloud** and **sapanalytics.cloud** to the allowed list of third-party cookies.

Clear your browser cache and log in again. You can now see the analytics stories when you click *Analyze*.

You can click *Analyze*, for example in *Campaigns* and *Spotlighting Accounts*, to find the analytics stories.

**Task overview:** [SAP Cloud Foundry Tenant \[page 285\]](#)

**Previous task:** [Setting Up the Outbound Connections from SAP Marketing Cloud to SAP Analytics Cloud \[page 297\]](#)

## Related Information

[Analytics Stories](#)

### 4.4.2.3 Renewal of Signing Certificates

Renew signing certificates.

When you log on to SAP Analytics Cloud, you are notified if the service provider certificate is about to expire. If you then decide to renew the signing certificate, you must download the signing certificate again. For more information about how to download the signing certificate, see:

- [Adding a New Live Data Connection \[page 274\]](#) if SAP Analytics Cloud is hosted on the SAC Neo tenant.
- [Adding a New Live Data Connection \[page 274\]](#) if SAP Analytics Cloud is hosted on the SAP Cloud Foundry tenant.

Afterwards, upload the signing certificate to your communication system. For more information about how to upload the signing certificate, see:

- [Setting Up a Communication System \[page 275\]](#) if SAP Analytics Cloud is hosted on the SAC Neo tenant.
- [Setting Up a Communication System \[page 289\]](#) if SAP Analytics Cloud is hosted on the SAP Cloud Foundry tenant.

**Parent topic:** [Integration with SAP Analytics Cloud \(1SO\) \[page 268\]](#)

**Previous:** [Create the Live Data Connection to SAP Marketing Cloud \[page 272\]](#)

**Next task:** [Selecting SAP Marketing Cloud Content Packages to Add to Your Tenant \[page 301\]](#)

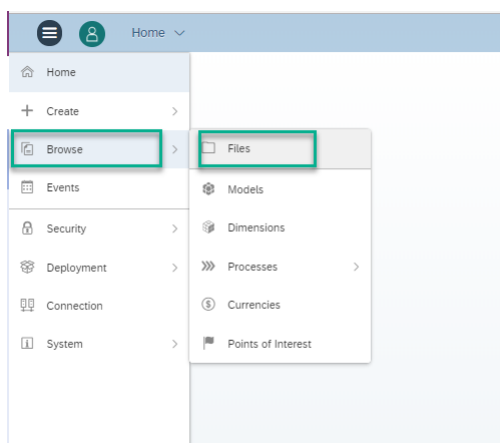


## 4.4.2.4 Selecting SAP Marketing Cloud Content Packages to Add to Your Tenant

Choose and then import SAP Marketing Cloud content to your SAP Analytics Cloud system to improve your analytics scenarios.

### Procedure

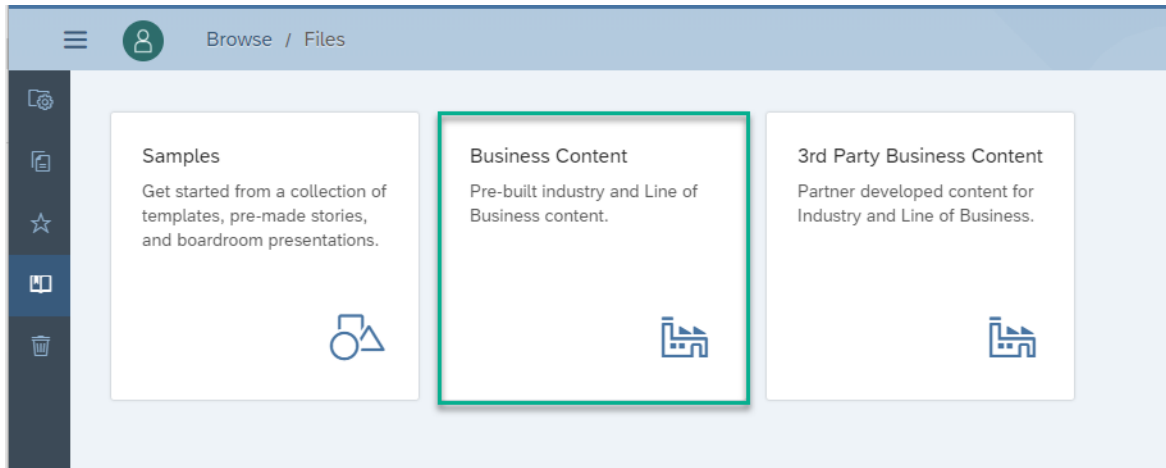
1. As an administrator, log on to your SAP Analytics Cloud system.
2. From the top-left menu, select **Browse** > **Files**.



3. On the left pane, choose **Content Library**.



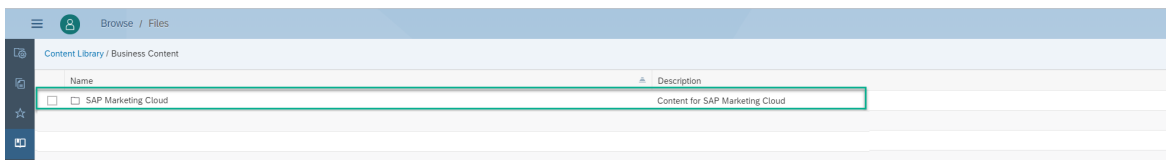
4. Choose the type of content that you want to add. You can choose either Samples or Business Content. Select **Business Content**.



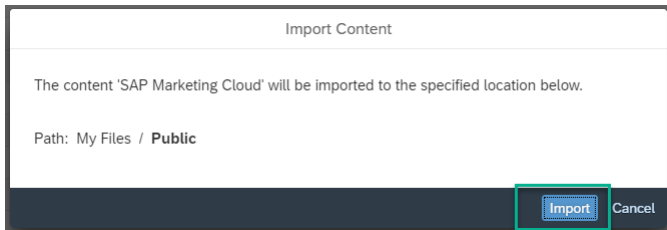
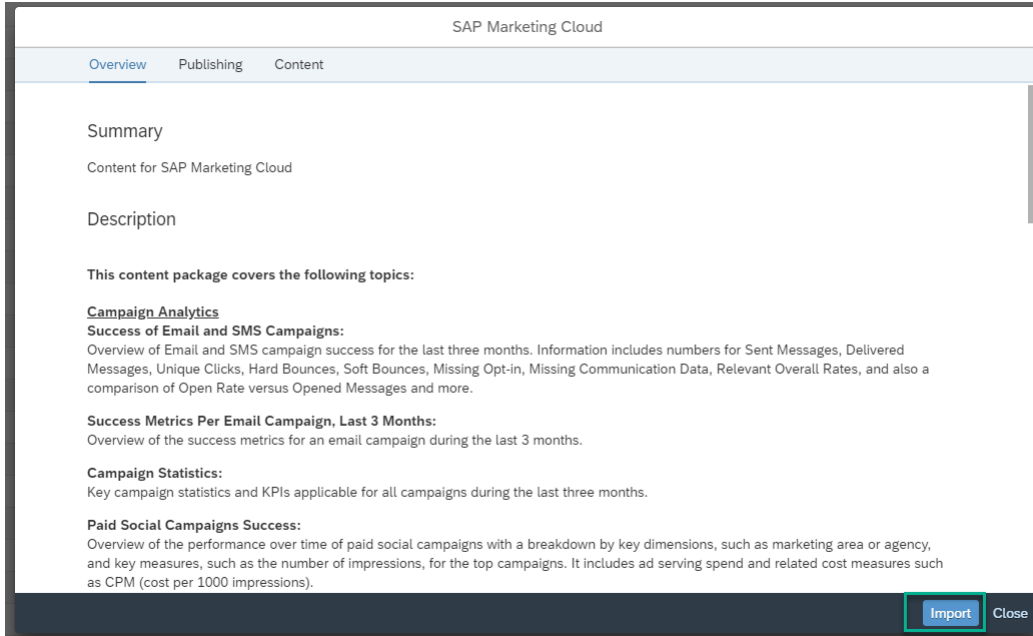
5. Select the SAP Marketing Cloud package.

### i Note

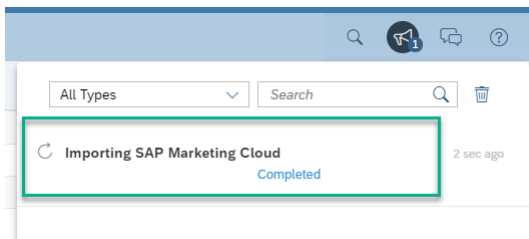
If you're an upgrade customer, please note that in your system has the old package *SAP Hybris Marketing Cloud*. When you do the import, the new package SAP Marketing Cloud is added to your system. Check the What's New document for a description of the updates. Now, you have to move all the existing stories (SAP-delivered plus any stories you created yourself) from the old package into the new package. To select the target folder when moving the existing stories, choose the path *Public → SAP\_Content → SAP\_Marketing* and then the appropriate folder. Please follow the existing folder structure. For every folder, you have to do this step separately. After you've moved all the stories, please delete the old package.

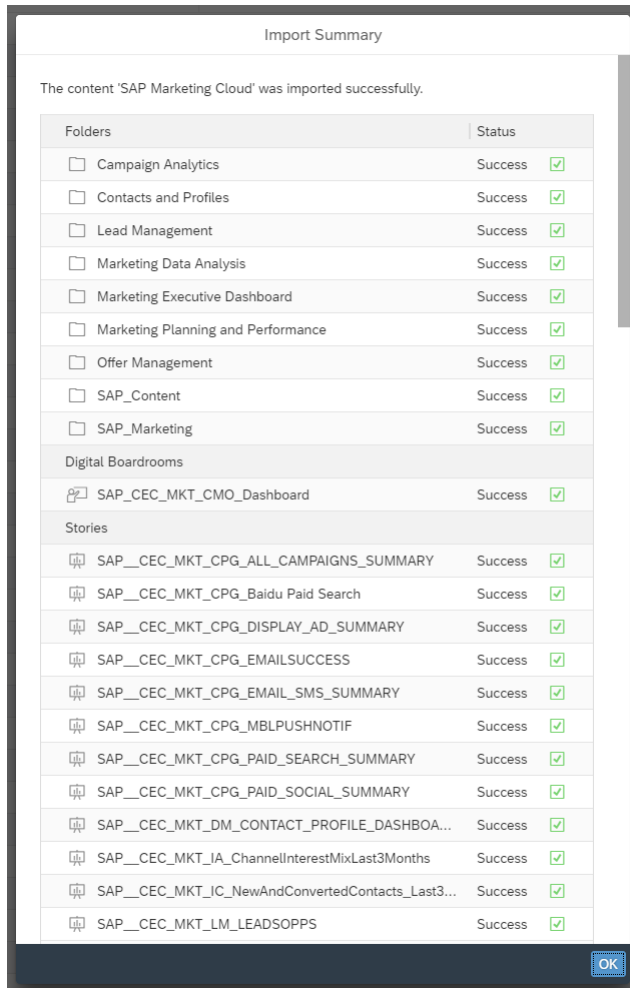


6. A popup comes up with the details of the content in the package. Choose *Import* to get the content.



The import process can take a while. The system notifies you when it's done, or if an error occurs.





**Task overview:** [Integration with SAP Analytics Cloud \(1SO\) \[page 268\]](#)

**Previous:** [Renewal of Signing Certificates \[page 300\]](#)

### 4.4.3 Content Studio Integrations

Lists the Content Studio integrations with SAP Marketing Cloud

#### [Simple Content Repository \[page 305\]](#)

Simple Content Repository, based on SAP Document Center, is available as an out-of-box feature for all users.

#### [Landing Page Publication \[page 305\]](#)

Landing Page Publication allows you to use your own company or brand domain name for landing pages published in SAP Marketing Cloud.

#### [Integrate with Content Management Systems or Digital Asset Management Systems \[page 306\]](#)

Integrate a Content Management System (CMS) or Digital Asset Management (DAM) system with the *Content Studio* app.

#### [Integrate with SAP Document Center \[page 312\]](#)

Integrate an SAP Document Center system with the *Content Studio* app.

#### [Integration with SAP Product Content Management \[page 314\]](#)

Use the integration with SAP Product Content Management to easily incorporate product pictures from SAP Product Content Management in your marketing messages. To use this integration, you must also use the communication arrangement: `SAP_COM_0051`.

#### [Integration with Return Path for Marketing Emails \[page 316\]](#)

You can use this integration to find out whether email providers would categorize emails that you want to send with a campaign as spam. You can also see if your email is displayed correctly on various devices, email programs and browsers.

#### [Integration with Litmus for Marketing Emails \[page 317\]](#)

You can use this integration to see if your email is displayed correctly on various devices, email clients and browsers.

### **4.4.3.1 Simple Content Repository**

Simple Content Repository, based on SAP Document Center, is available as an out-of-box feature for all users.

It allows you to select images from and upload images to the Content Studio.

You can activate this feature in your tenant using the Content Repository Configuration app. For more information, see [Configure Content Repositories](#).

### **4.4.3.2 Landing Page Publication**

Landing Page Publication allows you to use your own company or brand domain name for landing pages published in SAP Marketing Cloud.

You can activate this feature and configure the required CDN domain name using the Content Repository Configuration app. For more information, see [Configure Content Repositories](#).

### 4.4.3.3 Integrate with Content Management Systems or Digital Asset Management Systems

Integrate a Content Management System (CMS) or Digital Asset Management (DAM) system with the *Content Studio* app.

#### Use

CMS or DAM systems provide catalogs of digital images, videos, documents, music, and so on. You can search for digital assets by keywords. You can integrate CMS or DAM systems with SAP Marketing Cloud to access images for use in the *Content Studio* app.

The *SAP Marketing Cloud Integration with Content Management System* integration package is available on the SAP API Business Hub. For more information, see: <https://api.sap.com>

#### Prerequisites

- To set up a communication system and communication arrangement, ensure that the business catalog role Communication Management (SAP\_CORE\_BC\_COM) is assigned to your SAP Marketing Cloud user.
- To configure and deploy the integration package in SAP Marketing Cloud, assign the roles mentioned in the following guide: [Persona](#)

#### Configure and Deploy the Integration Flows

The following artifacts are available in the *SAP Marketing Cloud Integration with Content Management System* package:

- *Template for CMS or DAM Integration*  
Use this integration flow template to develop an integration flow. You can use the developed integration flow to integrate digital assets of any CMS or DAM system with SAP Marketing Cloud.
- *OpenText DAM System Integration*  
Use this integration flow to integrate digital assets of the OpenText DAM system with SAP Marketing Cloud.

To configure and deploy the integration flows:

1. From your development tenant of your SAP Cloud Integration account, choose *Discover*, and then select and copy the *SAP Marketing Cloud Integration with Content Management System* package. The copied package appears in the *Design* view in your tenant.

#### i Note

You can use SAP Marketing Cloud to configure and deploy the integration package. For more information, see [SAP Cloud Integration](#).

2. Select the required artifact.
3. Modify the integration flow of the selected artifact.
4. Configure the following blocks in the Upload integration flow:  
 For more information on modifying externalized parameters in integration flows, see [Externalize Parameters of an Integration Flow](#).

Externalized parameters for integration flows

Field Name	Entry Value
DAM_CREDENTIAL	<p>Enter the name of the deployed credential artifact.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>This credential is required to connect to the CMS or DAM system.</p> </div> <p>For information on deploying the User Credential Artifact, see <a href="#">Deploying or Editing a User Credentials Artifact</a>.</p>
IFLOW_ENDPOINT	<p>Enter the relative path of the integration flow endpoint. The SAP Marketing Cloud system invokes the IFLOW_ENDPOINT endpoint by using the keyword_QUERY parameter along with toGET access the search endpoint.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>If keyword_QUERY parameter is not passed, connect is executed.</p> </div> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>Use this path in the <i>Communication Arrangements</i> app with the prefix <b>/http</b> as follows:</p> <p><b><code>/http&lt;relative_path&gt;</code></b></p> <p>For example, if you specify the value in this field as <b>/OpenText</b>, then use the following path in the <i>Communication Arrangements</i> app:</p> <p><b><code>/http/OpenText</code></b></p> </div>
DAM_ACCESS_URL	<p>Enter the address of the CMS or DAM endpoint to check or establish the connectivity with the CMS or DAM system.</p> <p>For example, use the following URL in the <i>OpenText DAM System Integration</i> integration flow:</p> <p><b><code>https://&lt;OpenText Host&gt;/otmmapi/v3/sessions</code></b></p>
DAM_SEARCH_URL	<p>Enter the address of the CMS or DAM endpoint to search the images in the CMS or DAM system.</p> <p>For example, use the following URL in the <i>OpenText DAM System Integration</i> integration flow:</p> <p><b><code>https://&lt;OpenText Host&gt;/otmmapi/v3/search/text</code></b></p>

Field Name	Entry Value
DAM_ACCESS_QUERY	<p>Enter the value of the query parameter in DAM_ACCESS_QUERY of the URL to check or establish the connectivity with CMS or DAM system.</p> <p><b>&lt;DAM_ACCESS_URL&gt;?&lt;DAM_ACCESS_QUERY&gt;</b></p> <p>For example, there is no value for the query parameter and URL in the <i>OpenText DAM System Integration</i> integration flow.</p>
DAM_SEARCH_QUERY	<p>Enter the value of the query parameter in DAM_SEARCH_QUERY of the URL to search images in the CMS or DAM system.</p> <p><b>&lt;DAM_SEARCH_URL&gt;?&lt;DAM_SEARCH_QUERY&gt;</b></p> <p>For example, use the following URL in the <i>OpenText DAM System Integration</i> integration flow:</p> <p><b>https://&lt;OpenText Host&gt;/otmmapi/v3/search/text?keyword_query\='(*\${header.keyword_query}*)' &amp;load_type\=metadata&amp;metadata_to_return\=ARTESIA.FIELD.ASSET DESCRIPTION</b></p>
DAM_RESULTLIST_XPATH	<p>XPath to split the response structure and get a series of digital assets.</p> <p>For example, use the following XPath in the <i>OpenText DAM System Integration</i> integration flow:</p> <p><b>/search_result_resource/asset_list</b></p>
DAM_SUGGESTION_URL	<p>Enter the address of the CMS or DAM endpoint to read the suggested keywords in the CMS or DAM system.</p> <p>For example, use the following URL in the <i>OpenText DAM System Integration</i> integration flow:</p> <p><b>https://&lt;OpenText Host&gt;/otmmapi/v4/search/text/suggestions</b></p>
DAM_SUGGESTION_QUERY	<p>Enter the value of the query parameter in DAM_SUGGESTION_QUERY of the URL to read the suggested keywords in the CMS or DAM system.</p> <p><b>&lt;DAM_SUGGESTION_URL&gt;?&lt;DAM_SUGGESTION_QUERY&gt;</b></p> <p>For example, use the following URL in the <i>OpenText DAM System Integration</i> integration flow:</p> <p><b>https://&lt;OpenText Host&gt;/otmmapi/v4/search/text/suggestions? input=\${header.keyword_suggestions}&amp;search_plugin_id=ARTESIA.PLUGIN.SEARCH.SOLR.V1&amp;max_suggestions=\${header.top}</b></p>



Field Name	Entry Value
DAM_SUGGESTIONRESULT_XPAT H	<p>XPath to split the response structure and get a series of suggested keyword.</p> <p>For example, use the following XPath in the <i>OpenText DAM System Integration</i> integration flow:</p> <pre><b>/search_suggestion_result_resource/ search_suggestion_result/suggestion_list</b></pre>

5. Configure the following blocks in *Integration Process*:

**i Note**

For more information on configuring integration flow blocks, see [Configure Integration Flow Components](#).

1. (Optional) In the *Content Modifier* block, enter the header parameters, which are required to call the CMS or DAM system. The *Content Modifier* block is placed after the *Router* search route and the *Start Event* block.
  2. Add the `${header.keyword_query} != null` and `${header.CamelHttpMethod} = 'GET'` condition to switch between connect and search logic. Ensure that connect logic is the default logic. The search logic is executed only when both the conditions are true.
  3. You can define the required parameters to modify the request, which connects and searches the CMS or DAM system.
6. Configure the following information in *Local Integration (Handle Search Response from DAM)*:
- In the *Content Modifier* block, enter the following *Properties* to map the image metadata between SAP Marketing Cloud and the required CMS or DAM system. The *Content Modifier* block is placed after the *Generic Splitter* block.

**i Note**

By default, the *Content Modifier* supports the following rendition types:

- Preview
- Thumbnail

You can add or remove the required rendition types as follows:

Add or remove the `DIGITALASSETFILE` node inside the node `ASSET` in *Content Modifier* body. The structure of `DIGITALASSETFILE` node is as follows:

**≡ Sample Code**

```
<DIGITALASSETFILE>
<ASSET_ID></ASSET_ID>
<FILE_ID></FILE_ID>
<MIME_TYPE></MIME_TYPE>
<RENDITION></RENDITION>
<URL></URL>
<WIDTH></WIDTH>
<HEIGHT></HEIGHT>
```

</DIGITALASSETFILE>

<FILE\_ID> accepts only numeric value. This value which must be unique for both Preview and Thumbnail renditions.

Field Name	Entry Value
ASSET_ID	ID of image
CONTENT_TYPE	Type of content
PREVIEW_FILE_ID	ID of image preview
THUMBNAIL_FILE_ID	ID of image thumbnail
PREVIEW_MIME_TYPE	MIME type of preview
THUMBNAIL_MIME_TYPE	MIME type of thumbnail
THUMBNAIL_RENDITION	Name of the rendition. For example, <i>Preview</i> , <i>Thumbnail</i> , <i>Original</i> , and so on.
PREVIEW_RENDITION	Name of the rendition. For example, <i>Preview</i> , <i>Thumbnail</i> , <i>Original</i> , and so on.
PREVIEW_URL	URL of image preview
THUMBNAIL_URL	URL of image thumbnail
TITLE	Title of image
DESCRIPTION	Description of image
ACCESS_DATA	This field does not require an entry value
PREVIEW_WIDTH	The width of the preview rendition image
PREVIEW_HEIGHT	The height of the preview rendition image
THUMBNAIL_WIDTH	The width of the thumbnail rendition image
THUMBNAIL_HEIGHT	The height of the thumbnail rendition image

### **i** Note

The following properties are sent by SAP Marketing Cloud to the integration flow as headers. These parameters are used while searching the CMS or DAM system.

- `keyword_query`: returns the search keywords entered in Content Studio.
- `top`: number of search results to be fetched as a part of the request.

- skip: number of records to be skipped to display the records in the current page.
- keyword\_suggestions: takes the characters entered in Content Studio to fetch the suggested keywords.

7. Deploy the integration package.
8. Set up the communication system.
 

Create a communication system, which you can later use to establish communication arrangements.

  1. Log on to your SAP Marketing Cloud system with a user that has administrator authorizations.
  2. Launch the *Communication Systems* app, and choose *New*.
  3. Enter a *System ID* and the *System Name* in the *New Communication System* window, and choose *Create*.
  4. In the *Communication System* page, enter the following:
    1. Under *Technical Data*, enter the details of either SCI or middleware that you want to connect to.
    2. Under *User for Outbound Communication* section, choose **+** to create a *New Outbound User*, which can connect to the configured SCI or middleware:
 

If you choose *Authentication Method: User Name and Password* from the dropdown, enter the following:

      - *User Name*: <your communication user name>
      - *Password*: <your communication user password>

If you choose *Authentication Method: SSL Client Certificate* from the dropdown, choose either *Default Client Certificate* or *Trusted Third-Party Key Pair*.

If you choose *Trusted Third-Party Key Pair* as authentication method, you have to browse to the Third-Party Key Pair and provide a password.

For more information, see: [Enabling Client Certificate Authentication](#)
5. Choose *OK*.

### Note

You can now establish a communication arrangement with the created system. Use the *Maintain Communication Arrangements* app for this purpose.

9. Set up the communication arrangement.
  1. Log on to your SAP Marketing Cloud system with a user that has administrator authorizations.
  2. In the SAP Fiori launchpad, choose the *Communication Arrangements* app.
  3. Create a communication arrangement.
  4. Choose the scenario *SAP\_COM\_0050 (Marketing - Generic DAM Integration)*.
  5. Enter the required arrangement name.
  6. Choose *Create*.
  7. In the *Communication Arrangements* screen, do the following:
    1. Under *Common Data*, choose the communication system that you defined in *Communication Systems* app.
    2. Under *Additional Properties*, choose the following:
      1. *CMS/DAM Name*: Enter the name of the CMS or DAM system. For example, **OpenText**.
      2. *Implementation Mode*: Choose *O1* for *SCI/Any Middleware*.
      3. *Allow content upload*: Choose *Empty*. If you select *SCI/Any Middleware* implementation mode, you cannot upload images. You can use SAP Document Center to upload images.

4. *Folder for Upload*: If you select SCI/Any Middleware implementation mode, you cannot upload images. You can use SAP Document Center to upload images.
5. *Origin Domain Name and Path for CDN*: Enter the domain name and the path of the Origin system that you have configured in the Content Delivery Network (CDN). For example, **<Origin Domain Name>/<Path>**.
6. *CDN Domain Name*: Enter the domain name generated during CDN configuration.
8. Under *Outbound Communication*, select the communication user name, which you previously defined.
9. Under *Outbound Services*:
  - Enter the path to access the deployed integration flow. The path contains the value of the externalized parameter `IFLOW__ENDPOINT` with the `/http` prefix.  
For example, if you specify the value of the externalized parameter as `IFLOW__ENDPOINT / OpenText`, then use the following path in the *Communication Arrangements* app:  
**`/http/OpenText`**
10. Choose *Save*.

### 4.4.3.4 Integrate with SAP Document Center

Integrate an SAP Document Center system with the *Content Studio* app.

#### Use

SAP Document Center provides anytime, anywhere access to view, edit, and collaborate on personal and corporate content in an easy-to-use mobile app. You can integrate SAP Document Center with SAP Marketing Cloud to access images for use in *Content Studio* app. This integration enables a user to search for content using keywords and also to upload content.

#### Prerequisites

- To set up a communication system and the communication arrangement, ensure that the business catalog role Communication Management (`SAP_CORE_BC_COM`) is assigned to your SAP Marketing Cloud user.
- To configure SAP Document Center in SAP BTP cockpit, you must have an SAP BTP account, enabled with SAP Document Center.

#### Configure SAP Document Center in SAP BTP cockpit

1. Launch SAP BTP.
2. Choose **► account ► subaccount ►**.
3. Choose *Services*.

4. Enable the administrator role for your user in SAP Document Center. For more information, see [https://help.sap.com/viewer/p/SAP\\_Document\\_Center](https://help.sap.com/viewer/p/SAP_Document_Center).
5. Choose ► *Service Configuration* ► *Configure SAP Document Center* ►.

#### i Note

Note down the domain name from the SAP Document Center popup. This is the host name of the system. You need this information to update the *Technical Data* section of the *Communication System* page in the *Communication Systems* app.

6. Choose ► *Settings* ► *Shared documents* ►.
  1. Select the *Allow Sharing* checkbox. The minimum password length should be 0, which is required for anonymous access.
  2. Select the *Allow Upload* checkbox.

## Configuration in the SAP Marketing Cloud

1. Define the communication user for Outbound Integration.
 

Create a new communication system, which you can later use to establish communication arrangements.

  1. Log on to your SAP Marketing Cloud system with a user that has administrator authorizations.
  2. Launch the *Communication Systems* app, and choose *New*.
  3. Enter a *System ID* and the *System Name* in the *New Communication System* window, and choose *Create*.
  4. In the *Communication System* page, enter the following:
    1. Under *Technical Data* section, enter the details of the system that you want to connect to.

#### i Note

Enter the domain name from the SAP Document Center popup here.

2. Under *User for Outbound Communication* section, choose + to create a *New Outbound User*, which can connect to the SAP Document Center:
  - *Authentication Method*: Choose *User Name and Password* from the dropdown.
  - *User Name*: <your communication user name>
  - *Password*: <your communication user password>

#### i Note

The digital assets uploaded from the *Content Studio* app is available in the shared repository of the user.

3. Choose *Create*.
5. Choose *Save*.

#### i Note

You can now establish a communication arrangement with the created system.

2. Set up the communication arrangement for `SAP_COM_0050` scenario.
 

Use the *Communication Arrangements* app for this purpose.

1. In the SAP Fiori launchpad, choose the *Communication Arrangements* app.
2. Choose *New* to create a new communication arrangement.
3. Choose the scenario *SAP\_COM\_0050 (Marketing - Generic DAM Integration)*, enter an appropriate arrangement name, and choose *Create*.
4. In the *Communication Arrangements* page, do the following:
  1. Under the *Common Data* section, choose the *Communication System* that you defined in the *Communication Systems* app.
  2. Under *Additional Properties* section, choose the following:
    - *CMS/DAM Name*: Enter the name of the CMS/DAM system.  
This name is displayed in the *Image Storage* dropdown under the *Upload New Image* tab in the *Content Studio* app.
    - *Implementation Type*: Choose *02* for *SAP Document Center*.
    - *Allow content upload*: Choose *X* to allow content upload.
    - *Folder for Upload*: Enter the folder name where you want to save the uploaded digital assets in the SAP Document Center.

#### **i Note**

If the folder name entered by the user is unavailable, a new folder with the entered name is created.

- *Origin Domain Name and Path for CDN*: Enter the domain name and the path of the Origin system that you have configured in the Content Delivery Network (CDN). For example, `<Origin Domain Name>/<Path>`
  - *CDN Domain Name*: Enter the domain name generated during CDN configuration.
3. Under *Outbound Communication* section, select the communication *User Name*, which you have previously defined.
  4. Under *Outbound Services* section, enter the following path: `/mcm/b/json`
5. Choose *Save*.

### 4.4.3.5 Integration with SAP Product Content Management

Use the integration with SAP Product Content Management to easily incorporate product pictures from SAP Product Content Management in your marketing messages. To use this integration, you must also use the communication arrangement: `SAP_COM_0051`.

#### Use

In SAP Marketing Cloud, you can create marketing emails using the Content Studio. A marketing email can contain digital assets, such as images. Digital assets are usually stored in an external Digital Asset Management (DAM) system. The DAM system also provides search capabilities for the media files. You can integrate SAP Product Content Management with the *Communication Systems* app as a DAM system.

## New Communication System

To add a new communication system, proceed as follows:

1. Open the *Communication Systems* app and choose *New* to create a new communication system. Make the following entries on the dialog box:
  - *System ID*: HYBRIS\_COMMERCE\_PCM
  - *System Name*: HYBRIS\_COMMERCE\_PCM
2. Choose *Create*.
3. On the following window, enter the host name of your SAP Product Content Management server in the *Host Name* field.
4. Add a *User for Outbound Communication* and choose *None* as *Authentication Method* in the *New Outbound User* dialog box.
5. Choose *Create*.

## Communication Arrangement

To add a new communication arrangement, proceed as follows:

1. Open the *Communication Arrangements* app and choose *New* to create a new communication arrangement.
2. On the dialog box, choose the scenario `SAP_COM_0051` and then choose *Create*.
3. On the following window, choose the communication system created above and *None* as the *User Name* in the area *Outbound Communication*. Make the following entries for *Retrieve product images from SAP Commerce PCM*:
  - *Port*: Enter the port number that is setup on the server for HTTPS (SSL)
  - *Path*: Enter the path to the V1 REST API of the Omni Commerce Channel (OCC)
  - *Service URL*: Enter the service URL
4. Choose *Save*.

## More Information

- [Using Image Links in Emails and Email Templates](#)
- [Creating a Personalized Email or Email Template](#)

## 4.4.3.6 Integration with Return Path for Marketing Emails

You can use this integration to find out whether email providers would categorize emails that you want to send with a campaign as spam. You can also see if your email is displayed correctly on various devices, email programs and browsers.

### Use

To check an email, its content is sent to the email address of an external service provider. For more information about the function, see [Using the Spam Filter and Email Previews](#).

This function is available if you have signed a separate contract with the external service provider Return Path. To do so, contact Return Path at <https://returnpath.com/request-a-demo/> or email [sap@returnpath.com](mailto:sap@returnpath.com).

### Procedure

1. **Request API credentials from Return Path.**

To do so, contact your Return Path account team or submit a support ticket. Make sure you specify that you are using SAP Marketing Cloud as your provider in the ticket.

SAP currently supports the following Return Path features with SAP Marketing Cloud:

- Inbox Preview
- Spam Filter Check

Once Return Path has created the account, you receive the following information:

- API Key
- API Secret

2. **Create an incident for the system in which you want to activate the email content check and enter the SAP component CEC-MKT-MEM.**

Enter the following information:

- Description: *Request for individual activation of SAP\_COM\_1035*
- Your credential data, which you received from Return Path (API Key and API Secret, ideally via Secure Store)

For information about securely transferring your Return Path credentials, see SAP Note [1773689](#).



### 4.4.3.7 Integration with Litmus for Marketing Emails

You can use this integration to see if your email is displayed correctly on various devices, email clients and browsers.

#### Use

To check an email, its content is sent to the external service provider Litmus. If you want to use the feature, you need a separate contract with Litmus, which you can apply directly from inside the Content Studio app or from the SAP App Center.

For more information about the function, see [Using the Preview with Litmus for an Email Lite](#).

For new customers from release 2005, the selection *Litmus Preview* is automatically displayed in the Content Studio app.

#### Procedure

Proceed as follows to make the selection visible to your users:

1. Log on as administrator.
2. Choose the *Maintain Business Role* tile.
3. Search for the role for which you want to enable access to the *Litmus Preview* function.
4. Choose the *Assigned Business Catalogs* tab and ensure the business catalog *Marketing - Content* (ID: SAP\_CEC\_BC\_MKT\_LIB1\_PC) is assigned.
5. Only if that business catalog is assigned, you can now add the new business catalog *Marketing - Content with Litmus* (ID: SAP\_CEC\_BC\_MKT\_CNT\_LTM\_PC).

#### i Note

If this business catalog *Marketing - Content with Litmus* is already assigned, delete it to remove the selection in the Content Studio app.

### 4.4.4 Enabling Geospatial Segmentation with here.com

Use the integration option to translate addresses to geo-coordinates and reverse, and to enable geospatial analysis for segmentation based on the connected maps.

For more information, see [Setting up the Geospatial Segmentation and Map Preview](#).

#### i Note

SAP only provides the interfaces and configuration options that allow you to connect the map visualization and geocoding services. The usage of here.com is not part of your end-user license agreement with SAP. It is your responsibility to check and/or adapt the default configuration.

## 4.4.5 Integration with Baidu Maps for Geospatial Segmentation (Deprecated)

The integration of Baidu Maps into Segmentation enables you to segment contacts in China by geographic location in a visualized way.

### i Note

To use this function, you must have contact location data in the form of geographic coordinates in SAP Marketing Cloud.

For a description of the function, see [Using Geospatial Segmentation with Baidu Maps \(Deprecated\)](#).

For setup instructions, see [Setting Up the Integration with Baidu Maps for Geospatial Segmentation \(Deprecated\) \[page 318\]](#).

### 4.4.5.1 Setting Up the Integration with Baidu Maps for Geospatial Segmentation (Deprecated)

Set up the integration of Baidu Maps into Segmentation.

#### Prerequisites

- You have applied for a Baidu account key to use Baidu Maps APIs through Baidu Maps Platform at <http://lbsyun.baidu.com>.
- You have a business role that contains the **Communication Management** (SAP\_CORE\_BC\_COM) business catalog. This business catalog is required for creating the communication system and communication arrangement.  
You can use the standard business role **Administrator** (SAP\_BR\_ADMINISTRATOR), which contains the **Communication Management** business catalog and other administration-related catalogs. Alternatively, you can create custom business roles using the *Maintain Business Roles* app.
- You have a business role that contains the **Marketing - Segmentation and Campaign Configuration** (SAP\_CEC\_BC\_MKT\_CPC\_PC) business catalog. This business catalog is required for activating the relevant segmentation profile in the *Segmentation Configuration* app.  
You can use the standard business role **Administrator - Marketing** (SAP\_BR\_ADMINISTRATOR\_MKT), which contains the **Marketing - Segmentation and Campaign Configuration** business catalog and other configuration-related catalogs. Alternatively, you can create custom business roles using the *Maintain Business Roles* app.
- You have a business role that contains the **Security** (SAP\_CORE\_BC\_SEC) business catalog. This business catalog is required for managing trusted sites.  
You can use the standard business role **Administrator** (SAP\_BR\_ADMINISTRATOR), which contains the **Security** business catalog and other administration-related catalogs. Alternatively, you can create custom business roles using the *Maintain Business Roles* app.

## Procedure

1. Create a communication system that represents the Baidu Maps system, as follows:
  - a. Log into SAP Fiori launchpad with a business role that contains the **Communication Management** (SAP\_CORE\_BC\_COM) business catalog.
  - b. Open the *Communication Systems* app.
  - c. Choose *New*.

The *New Communication System* dialog box appears.

- d. Enter a system ID and a system name. Choose *Create*.

Examples:

- **System ID:** BAIDU\_MAP
- **System name:** Baidu Map

The editing screen for the communication system appears.

- e. In the *Host Name* field, enter **www.baidu.com**, which is the host name of the Baidu Maps server. Choose *Save*.
- f. Create a user for the outbound communication, as follows:
  1. In the *User for Outbound Communication* section, choose + (Add).
  2. Set the authentication method to **None**. Choose *Create*.
- g. Save and activate the communication system.

Do not exit SAP Fiori launchpad.

2. Create a communication arrangement for communicating with Baidu Maps, as follows:
  - a. Open the *Communication Arrangements* app.
  - b. Choose *New*.

The *New Communication Arrangement* dialog box appears.

- c. Enter scenario SAP\_COM\_0075 and an arrangement name. Choose *Create*.

The editing screen for the communication arrangement appears.

- d. In the *Communication System* field, enter the communication system that you have created.
- e. In the *Baidu Account Key* field, enter your Baidu account key.
- f. In the *Baidu Map API* section, enter the required information, such as the path.
- g. Save and activate the communication arrangement.
- h. Exit SAP Fiori launchpad.

3. Activate the **All China Consumers (B2C)** segmentation profile, as follows:
  - a. Log into SAP Fiori launchpad with a business role that contains the **Marketing - Segmentation and Campaign Configuration** (SAP\_CEC\_BC\_MKT\_CPC\_PC) business catalog.
  - b. Open the *Segmentation Configuration* app.
  - c. Choose *Segmentation Profiles*.
  - d. Select *All China Consumers (B2C)* from the left pane.
  - e. Choose *Edit* in the *Use in Applications* section in the right pane.
  - f. Select the *Use* checkbox for the **Segmentation Model** application.
  - g. Choose *Save*.
  - h. Go back to the initial screen of the *Segmentation Configuration* app. Choose *Segmentation Objects and Attributes*.

- i. Search for the *China Consumer* (deprecated; SAP\_CE\_LOC\_CN; use the segmentation object *Contacts* instead) segmentation object. Select it after you find it.
  - j. From the details screen of the segmentation object on the right, select the SAP\_CONTACT\_INTERACTIONS\_CN data source. A details screen of the data source opens.
  - k. Switch to the edit mode by choosing *Edit*. Do the following on **both the IA\_LATITUDE\_BD09 and IA\_LONGITUDE\_BD09 properties**:
    - o Select the *Visible as Attribute* checkbox.
    - o From the *Attribute Group* dropdown list, select *Interactions*.
  - l. Choose *Save*.
  - m. Exit SAP Fiori launchpad.
4. Log into SAP Fiori launchpad with a business role that contains the **Security** (SAP\_CORE\_BC\_SEC) business catalog.
  5. Open the *Manage Content Security Policy* app.
  6. On the *Trusted Sites* tab, select *UI\_RESOURCES\_SCRIPTS*. A details pane opens on the right.
  7. On the *Managed by Customer* tab, add **https://api.map.baidu.com** as a trusted site. Save your changes.

## 4.4.6 SAP Jam Integration for Collaboration

The integration enables using SAP Jam in SAP Marketing Cloud to facilitate the collaboration when planning and executing marketing campaigns.

### Note

Keep in mind that you need the Social Collaboration Integration (SAP\_COM\_0026) for this integration. For more information, see [SAP Jam Collaboration](#) under *Administration*.

For information about how to set up the integration, see SAP Help Portal at [https://help.sap.com/viewer/p/SAP\\_JAM\\_COLLABORATION](https://help.sap.com/viewer/p/SAP_JAM_COLLABORATION), [Administration](#) [Administrator Guide \(HTML\)](#) [Integrations](#).

## Prerequisites

- SAP Jam is installed.
- SAP Jam server is defined in the configuration of the system.
- System user is also a SAP Jam user.
- For the SAP Jam widget to appear in the **Campaigns** app, ensure you maintain the source URL, for example, [https://\\*.sapjam.com/](https://*.sapjam.com/), in the whitelist of the Content Security Policy. For more information, see [Manage Content Security Policy](#).

## Features

- Create and link SAP Jam group or assign existing group.
- Post status changes to SAP Jam group.
- Automatic upload of export files to SAP Jam group.
- Display SAP Jam feed in a campaign.

### 4.4.7 Verifying Email Addresses Using a Partner Solution

Accurate email addresses are vital for email marketing campaigns. To verify email addresses you can use partner services, such as Neverbounce or others.

For more information, see the blog [SAP Marketing Cloud – Verifying E-Mail addresses with Neverbounce](#).

### 4.4.8 Integration with an External Coupon Service System

Integrate SAP Marketing Cloud with an external coupon service system. To use this integration, you must use the communication arrangement: `SAP_COM_0286`.

## Use

SAP Marketing Cloud allows you to integrate with an external coupon management service. Such a service is responsible for coupon code creation, validation and redemption. The offer and coupon functionality in SAP Marketing Cloud is also responsible for consistently publishing offers with and without assigned coupons using the different digital marketing channels.

Perform the following tasks to set up an external coupon service.

## Implement Outbound Interface

Implement the external coupon outbound interface, either directly in the external system, or by using a suitable integration middleware, such as SAP Cloud Integration (<https://cloudplatform.sap.com/capabilities/product-info.SAP-Cloud-Platform-Integration.cceaaf2b-8ceb-4773-9044-6d8dad7a12eb.html>), to map the SAP Marketing Cloud interface to the external system interface. The external coupon service must follow the REST application protocol via HTTPS and must accept JSON.

## Set Up a Communication Arrangement

1. Set up the communication system to define the endpoint of the external coupon service.  
Create a new communication system, which you can later use to establish communication arrangements.
  1. In the SAP Fiori launchpad, log on with a user that has administrator authorizations.
  2. Launch the *Communication Systems* app, and choose *New*.
  3. In the *New Communication System* window, define the *System ID* for the communication system, for example, Z\_EXTERNAL\_COUPON\_SRV. Define a *System Name*. You can freely define a name; note that the name is used when you create the communication arrangement. Choose *Create*.
  4. In the *Communication System* page, enter the following:
    1. Under *Technical Data*, specify the external system you want to use for the external coupon service. Indicate the pure host name, no path, no port. Define the HTTPS Port, default is 443. Note that the other properties under Technical Data are not relevant.
    2. Optionally, you can provide your contact information for the communication system you are defining.
    3. Define the users to be used for the communication:  
Under *User for Inbound Communication*, create a new technical user. This user will be able to access the corresponding inbound APIs to write Coupon Codes and Redemption Interactions.  
Under *User for Outbound Communication*, define the user to be used to access your external coupon service.
  5. Choose *Save* to save the new or edited communication system in an active status..

### i Note

You can now establish a communication arrangement with the created system. Use the *Maintain Communication Arrangements* app for this purpose.

2. Set up the communication arrangement.  
Create a communication arrangement and reference the communication system created in the first step.
  1. In the SAP Fiori launchpad, choose the *Communication Arrangements* app.
  2. Create a new communication arrangement.
  3. In the *New Communication Arrangement* dialog, under *Scenario*, use the value help to select the predefined scenario *Marketing - External Coupon Management Service Integration (SAP\_COM\_0286)*.
  4. Enter the required arrangement name.
  5. Choose *Create*.
  6. In the *Communication Arrangements* screen, do the following:
    1. Under *Common Data*, use the value help to select the communication system you have created in the *Communication Systems* app.
  7. Under *Outbound Services* specify the relative path for the outbound service. The host name is already defined in the communication system. The resulting URL to connect to the external service via HTTPS will be `https://<service_url>/<path>:<port>`.
  8. Choose *Save* to save the new or edited communication arrangement in an active status.

### i Note

You should only maintain one communication arrangement for the scenario SAP\_COM\_0286. Although you can create multiple communication arrangements, only the most recently activated communication arrangement is used when replicating coupons.

## Process Overview

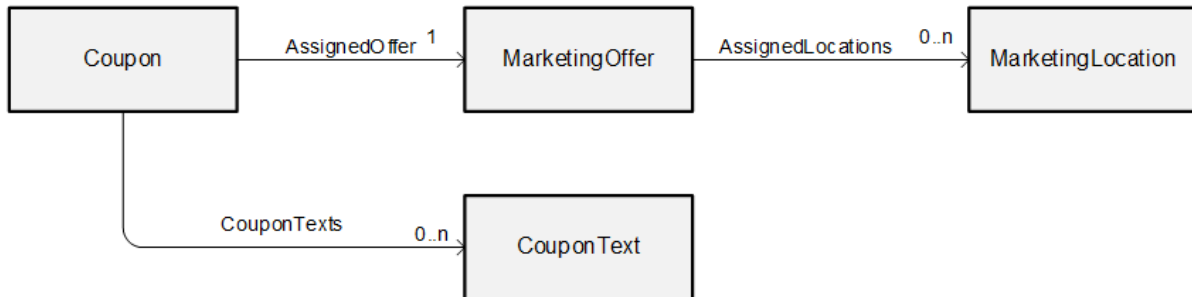
Once you have setup your external coupon service and the connection to it, the external coupon management process is performed as follows. You require the business catalog role Marketing Recommendation (SAP\_CEC\_BC\_MKT\_REC\_PC) to work with offers and coupons.

1. Plan your offer with coupon  
In the *Manage Offers* app, create an offer with a coupon feature. Maintain your offer details.
2. Create and assign a coupon for your offer  
In the coupon tab for an offer, assign an existing coupon by searching the coupon value help or create a new coupon directly by choosing *Create*. This creates a default coupon, navigates you to the coupon details and assigns the coupon to the offer in a single step.
3. Release the coupon and replicate it to the external system  
In the *Manage Coupons* app, in the section *General Information*, maintain the integration property. Set the value for *Replicate to External Coupon Service* to *Yes*.  
Choose *Release* and then *Replicate* to transfer the coupon and assigned offer information to the external system. In the dialog, define the initial number of coupon codes to be created by the external system.
4. Release the offer and use it in your execution process (such as in an email campaign)  
Navigate back to the offer and choose *Release*. You can now use the offer in execution processes. For example, you can run an email campaign with an email message that contains the offer. Note that the offer needs to have coupon codes and needs to also be visible during campaign execution.
5. Changes made to your offer are communicated to the external system  
If you change the offer (either in the preparation phase of the offer or after setting the status to Paused), these changes are also communicated to the external coupon service.
6. If needed, request more coupon codes during the lifetime of your offer  
During the lifetime of the offer with coupon, you can request additional coupon codes from the external coupon system. Open the app *Manage Coupons* and navigate to the details for the coupon object. In the section *Coupon Codes*, you can choose *Request Additional Codes*. On the dialog box, define the number of additional codes to be added to the existing number of codes. The properties *Total Number of Requested Codes* and *Number of Pending Codes* in the *Integration* section of the coupon header give an overview about replication progress.

For information about the overall process of offers with coupon and external coupon service, see [Offers with External Coupon Service](#).

## Implementing the External Coupon Service

To replicate coupon and offer information to an external coupon service, SAP Marketing Cloud calls a defined REST endpoint with a JSON payload via HTTPS. The payload is structured into four different objects, as you see in the following figure:



As long as the offer can be changed, SAP Marketing Cloud calls the REST endpoint.

A detailed overview about the properties, their data types and semantics can be found in the following table:

Object	Property	Data Type	Size	Semantic	Example Value
Coupon	CouponUUID	UUID	36	Internal identifier of the coupon header	6c0b84b7-5523-1e e8-9689-445f06b 876e8
	Coupon	String	32	User-defined identifier of the coupon	CPN_201835_000 1
	CouponOrigin	String	30	User-defined origin of the coupon identifier	DEMO
	CouponType	String	10	Indicates the type of the coupon	SINGLE – Single Coupon Code MULTI – Multiple Coupon Codes
	CouponContact RelationshipType	String	2	Indicates the relationship between a coupon code and the contact	01 – No Contact Assigned 03 – Contact Assigned Dynamically
	CouponStatus	String	2	Status of the coupon code	01 – In Preparation 02 – Released



Object	Property	Data Type	Size	Semantic	Example Value
	CouponCode ValidityDuration	Decimal	15/0	Duration of the validity of an individual code.	14
	CouponCode ValidityDurationUnit	String	3	Unit of the duration, defaulted to days	DAY
	CpnCodeValidity StartDelay	Decimal	15/0	Validity start delay of an individual code	1
	CpnCodeValidity StartDelayUnit	String	3	Unit of the validity start delay, defaulted to days	DAY
	TotalNumberOf Redemption	Integer	--	Total number of redemptions possible for this coupon	1.000
	TotalNumberOf RdmptnPerContact	Integer	--	Number of redemptions possible for each contact or code.	1
	NumberOfRequested CouponCodes	Integer	--	Number of codes to be generated by the external coupon system.	50.000
	SingleCoupon-Code	String	128		SUMMER-SALE
Coupon Texts	Language	String	2	ISO code of the language-dependent name and description	EN
	CouponName	String	120	Name of the coupon	
	CouponDescription	String	512	Description of the coupon	
Assigned Offer	MarketingOffer	String	10	Internal identifier of the offer	0000000815

Object	Property	Data Type	Size	Semantic	Example Value
	ExternalOffer	String	60	External identifier of the offer if it was created externally and not within SAP Marketing Cloud	
	ExternalOfferOrigin	String	30	Origin of the external offer identifier	
	OfferStatus	String	2	Status of the offer	00 – In Preparation 01 – Released 02 – Paused
	OfferValidity Start-DateTime	Date Time	--	ISO 8601-compliant timestamp in UTC	2018-06-13T22:00:00.000+0000
	OfferValidity End-DateTime	Date Time	--	ISO 8601-compliant timestamp in UTC	2018-06-13T22:00:00.000+0000
	OfferVisibility StartDateTime	Date Time	--	ISO 8601-compliant timestamp in UTC	2018-06-13T22:00:00.000+0000
	OfferVisibility End-DateTime	Date Time	--	ISO 8601-compliant timestamp in UTC	2018-06-13T22:00:00.000+0000
Assigned Locations	MarketingLocation	String	50	Identifier of the marketing location	
	MarketingLocation Origin	String	30	Origin of the location identifier	SAP_RETAIL_STORE

In the lifecycle of the offer with coupon, the following requests will be sent to the external coupon service. All requests always contain the complete set of properties with their current values.

1. A HTTP POST request for the initial replication of the coupon with the assigned offer data.
2. A HTTP PUT request for any changes to the assigned offer (such as adding assigned marketing locations, extending the validity period of the offer). The property NumberOfRequestedCouponCodes is 0 in this case.
3. A HTTP PUT request to request additional codes. The property NumberOfRequestedCouponCodes contains the actual number of requested codes and not the overall number of codes to be generated. If for example the initial replication requests 50.000 codes and an additional request for 10.000 codes is issued, the NumberOfRequestedCouponCodes is 10.000 for the second request for additional codes.

Example payload for the initial replication request for a coupon with 50.000 codes. The URL of the endpoint is defined in the communication arrangement / RFC destination.

### Sample Code

```
POST /coupons
{
  CouponUUID: "6c0b84b7-5523-1ee8-9add-d272824ef884",
  Coupon: "CPN ID 4711",
  NumberOfRequestedCouponCodes: 50000,
  ...
  CouponTexts: [{
    Language: "EN",
    CouponName: "Coupon Name",
    CouponDescription: "Coupon Description"
  }]
  AssignedOffer: {
    MarketingOffer: "OFFER ID 4711",
    ExternalOffer: "PMR ID 4711",
    ExternalOfferOrigin: "SAP_PMR",
    ...
    AssignedLocations: [{
      MarketingLocation: "LOC ID 4711",
      MarketingLocationOrigin: "SAP_RETAIL_STORE"
    }]
  }
}
```

Example payload for a subsequent request of 10.000 additional codes. The complete payload is sent again. The only difference here is the value of the NumberOfRequestedCouponCodes property.

### Sample Code

```
PUT /coupons
{
  CouponUUID: "6c0b84b7-5523-1ee8-9add-d272824ef884",
  Coupon: "CPN ID 4711",
  NumberOfRequestedCouponCodes: 10000,
  ...
  CouponTexts: [{
    Language: "EN",
    CouponName: "Coupon Name",
    CouponDescription: "Coupon Description"
  }]
  AssignedOffer: {
    MarketingOffer: "OFFER ID 4711",
    ExternalOffer: "PMR ID 4711",
    ExternalOfferOrigin: "SAP_PMR",
    ...
    AssignedLocations: [{
      MarketingLocation: "LOC ID 4711",
      MarketingLocationOrigin: "SAP_RETAIL_STORE"
    }]
  }
}
```

## CSRF protection

We provide CSRF protection according to the standard implemented at SAP: Any modifying request is rejected unless the header attribute `x-xsrf-token` is added with a valid token value. The client must be able to obtain a valid token using the following procedure:

- The token will be requested by a HTTP HEAD request to the default endpoint URL.
- This call will include the name/value pair "x-xsrf-token/fetch" in the request header.
- The response includes the name/value pair "x-xsrf-token/<validToken>". The valid token will be used for subsequent requests.

It is not mandatory for an external coupon service to implement CSRF protection as long as the HTTP HEAD request to fetch the CSRF token will not fail.

## Handling Errors

If an error occurs when communicating with the external coupon service, SAP Marketing Cloud expects a corresponding HTTP status code. Furthermore, details about the source of the error can be included as JSON in the body of the HTTP response. The JSON is parsed and processed further, for example the error message is shown in the [Manage Coupon](#) user interface when the replication of the coupon fails.

Example of an error response:

### Sample Code

```
{
  "error": {
    "status": 404,
    "message": "Coupon not found",
    "target": "/path_to_api_endpoint/object_id",
    "details": [{
      "message": "Detailed error message goes here"
    }]
  }
}
```

## Implementing Inbound Interfaces

In addition to the outbound interface used to create a coupon with offer information in an external coupon management system, the following SAP Marketing Cloud inbound APIs are also relevant to the overall process:

- Importing coupon codes into SAP Marketing Cloud
- Importing redemption interactions into SAP Marketing Cloud

### Importing Coupon Codes

The external coupon management service is responsible for generating coupon codes and transferring them back to SAP Marketing Cloud for distribution in the digital channels. This can be done using the Coupon OData API. For more information, see [Coupons \[page 1026\]](#).

## Sample Code

Example payload to create a single coupon code for a given coupon UUID

```
POST /sap/opu/odata/sap/API_MKT_COUPN_SRV/Coupons(guid'<coupon_uuid>')/
to_CouponCode
{
  "CouponCode": "Coupon Code, e.g. Web-Code",
  "CouponCodeSerialNumber": "Coupon Code Serial Number",
  "EANCodeImageURL": "Image URL to EAN code",
  "QRCodeImageURL": "Image URL to QR code"
}
```

## Importing Redemption Interactions

One integral part of an external coupon management service is the redemption of coupon codes. SAP Marketing Cloud can also use this information to optimize the distribution and communication of offers with coupons. For example, the offer recommendation will only recommend offers with assigned coupons when the redemption limit has not yet been reached.

The interaction API can be used to inform SAP Marketing Cloud about offer redemption events. For more information, see [Interactions \[page 615\]](#).

The following pre-defined interaction type is delivered with SAP Marketing Cloud:

OFFER\_REDEMPTION: Inform about a redemption of an offer with assigned coupon. The redemption counter in SAP Marketing Cloud will be increased in this use case.

## Sample Code

Example payload for an anonymous offer redemption interaction from an online shop

```
POST /API_MKT_INTERACTION_SRV/Interactions
{
  "InteractionTimeStampUTC": "/Date(1530626397595)/",
  "CommunicationMedium": "ONLINE_SHOP",
  "InteractionType": "OFFER_REDEMPTION",
  "InteractionIsAnonymous": true,
  "InteractionOffers": [{
    "MarketingOffer": "0000000815",
    "CouponCode": "CODE-0815-ABC"
  }]
}
```

For more information about set up of interaction channels, see [Managing Interaction Content](#).

## 4.4.9 Partner Extension: Integrate with Digital Market Intelligence

With the partner integration of SimilarWeb, you can see the web no app traffic of your competitors for each channel, such as direct, email, social. These insights help you to make better strategic decisions with regard to your own campaigns. Note that the extension is an offering from a partner of SAP.

To add the partner extension in SAP Marketing Cloud, do the following steps:

1. Open your system of SAP Marketing Cloud and go to the *Extensions* under your user details.
2. Add an extension with
  - o a section
  - o a title, for example, **SimilarWeb**
  - o and a URL. For an overview of all available URLs, see [Widgets Demo of SimilarWeb](#) ↗

Now, you should see an additional tab in the *Marketing Plans* app, when you open a single plan.

For more information, see [SimilarWeb](#) ↗ .

### Related Information

[UI Extensions](#)

## 4.4.10 Marketing Events

This section for marketing events explains the following:

- How to integrate events, registrants, and participants data from third-party event provider platforms, such as ON24 platform with SAP Marketing Cloud using integration flows. For more information, see [Integration with Marketing Events \[page 330\]](#).
- How to integrate events, registrants, and participants data from GoToWebinar platform with SAP Marketing Cloud using integration flows. For more information, see [Integration with GoToWebinar using SAP Cloud Integration Open Connectors \[page 332\]](#).

### 4.4.10.1 Integration with Marketing Events

The integration of events data from event provider platform, such as ON24 platform with SAP Marketing Cloud enables promotion of these events via campaigns and analysis of their success rate after event completion. This integration fetches and stores event data easily from the event provider platforms into SAP Marketing Cloud system.

To achieve this integration the following iFlows are provided:

- Fetch Marketing Events Data from Event Provider Platforms**

This integration flow calls the API endpoints from the event provider platform (ON24) to fetch the events data and map it to the SAP Marketing Cloud format. This fetched data is used by the application job to create the marketing event object in SAP Marketing Cloud system.

If the iFlow is run in full mode `header value delta = false`, then all the events from the specified start date plus 180 days are fetched and updated in SAP Marketing Cloud system. If the iFlow is run in delta mode `header value delta = true`, then all the events updated from the `lastrundatettime`, that is, header value plus 180 days will be fetched and updated in SAP Marketing Cloud system.
- Fetch Registrant Data from Event Provider Platforms**

This integration flow calls the API endpoints from the event provider platform (ON24) to get the registrants' data for an event and map it to the SAP Marketing Cloud format. This fetched data is used by the application job to update or created the following objects in SAP Marketing Cloud system:

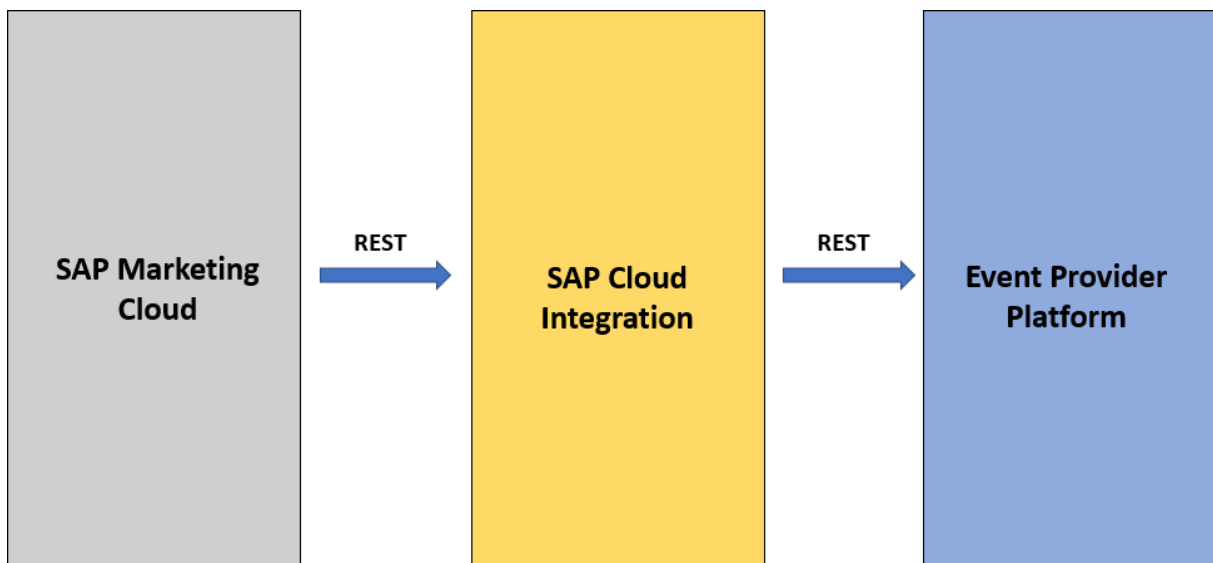
  - Registrants are updated or created as interaction contacts
  - Activities like registration, permission opt-in are created as interactions of respective contacts
- Fetch Participant Data from Event Provider Platforms**

This integration flow calls the API endpoints from the event provider platform (ON24) to get the participants (attendee) data for an event and map it to the SAP Marketing Cloud format. This fetched data is used by the application job to update or created the following objects in SAP Marketing Cloud system:

  - Participants are updated or created as interaction contacts
  - Engagement data of participants like polls and surveys are created as survey responses of respective contacts
- Fetch Participant Engagement Data from Event Provider Platforms**

This integration flow calls the API endpoints from the event provider platform (ON24) to import engagement data of participants such as polls and surveys metadata to SAP Marketing Cloud system. This metadata is used to create the poll or survey business object in SAP Marketing Cloud system. Survey responses to these polls and survey are captured using the '*Fetch Participant Data from Event Provider Platforms*' iflow .

The integration package runs on the SAP Cloud Integration tenant and fetches events data from event provider platforms and transform it into SAP Marketing Cloud format. The Network Security team takes responsibility for preparing the network environment across different systems and related security aspects.



For more information, see [Integrating Marketing Events Data with SAP Marketing Cloud](#).

## 4.4.10.2 Integration with GoToWebinar using SAP Cloud Integration Open Connectors

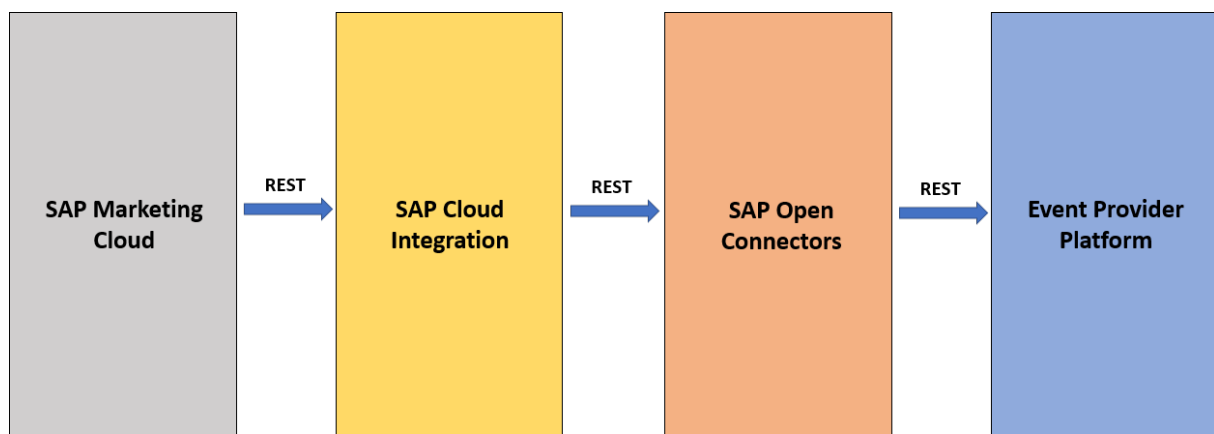
The integration of events data from GoToWebinar platform with SAP Marketing Cloud enables promotion of these events via campaigns and analysis of their success rate after event completion. This integration fetches and stores event data easily from the event provider platforms into SAP Marketing Cloud system.

To achieve this integration the following iFlows are provided:

- Fetch Marketing Events Data from GoToWebinar Platform - Events data is fetched from the GoToWebinar platform.
- Fetch Registration Data from GoToWebinar Platform - Registrants' information is fetched from the GoToWebinar platform, and registrants are created as interaction contacts on SAP Marketing Cloud system.
- Fetch Participation Data from GoToWebinar Platform - Participant data is fetched from the GoToWebinar platform. The participants are created as new interaction contacts. The participant details will also contain engagement data like polls and surveys. The engagement data of participants is created as interaction activities in SAP Marketing Cloud system.

The integration package runs on the SAP Cloud Integration tenant and fetches events data from event provider platforms and transform it into SAP Marketing Cloud format. The Network Security team takes responsibility for preparing the network environment across different systems and related security aspects.

SAP Cloud Integration Open Connectors are used to simplify the connectivity and provide seamless integration with GoToWebinar platform. If there are any updates from the event provider platform, they will be handled by SAP Cloud Integration Open Connectors.





## Limitations on GotoWebinar Integration

### Functional Limitations

The following data is not supported by GotoWebinar platform as their APIs do not fetch these details for a specific event or a participant:

- On-demand Duration: The time during which a participant viewed the recording of the event.
- Content Downloads: The number of different types of content downloaded by a participant.
- Participation Score: A measure for a participant's engagement in an event. This score is determined by comparing and rating concrete KPI values of a participant in an event.
- On-demand Recording Available From: The date from which the recording of the event is available for viewing.
- On-demand Recording Available Until: The date until which the recording of the event will be available for viewing.

#### i Note

GoToWebinar allows recurring events with multiple sessions, but multiple sessions are not supported with this integration. You should set up a single session between each SAP Marketing Cloud event and GoToWebinar webinar.

### Technical Limitations

The following technical limitations are found with integrating GoToWebinar platform data using SAP Cloud Integration Open Connector:

- The data being fetched from GoToWebinar platform and integrated in SAP Marketing Cloud by marketing application job cannot effectively run in incremental mode. Event data that is within the application job date range will be fetched irrespective of whether data has been changed since the last run or not. Due to this, the number of data fetch calls made using SAP Cloud Integration Open Connectors increases and the performance for application job execution is suboptimal.
- The registrant or participant details are fetched and integrated by calling the participant API individually for each registrant or participant. Due to this, the number of data fetch calls made using Open Connectors increases depending on the number of registrants or participants.

For more information, see [Integrating GoToWebinar Data with SAP Marketing Cloud using SAP Open Connectors](#).

## 4.5 Suite-Enabling Integrations

This section contains details of integration with applications in the SAP Suite, such as SAP Customer Experience, S/4HANA, CRM, ERP, and includes inbound, outbound, and bidirectional integration.

[Sales and Service \(Inbound\) \[page 334\]](#)

[Sales Automation \(Outbound\) \[page 352\]](#)

Set up the integration of a sales system with SAP Marketing Cloud

[Financial Data \[page 379\]](#)

[Survey Data \[page 383\]](#)

[Personalized Commerce \[page 385\]](#)

## 4.5.1 Sales and Service (Inbound)

The integrations below enable you to integrate sales and service data with your system:

- SAP ERP  
For more information, see [Integration with SAP ERP \[page 349\]](#).
- SAP Customer Relationship Management (CRM) and SAP Cloud for Customer  
Enables you, for example, to replicate SAP CRM and SAP Cloud for Customer business partners and business documents.  
For information about how to set up the integration, see [Integration with SAP CRM - Inbound Channel \[page 342\]](#) and [Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#).
- External Sales Systems  
Depending on the data model of the external sales system, business partners and business documents can be replicated.  
For more information, see [Integration with External Sales Systems - Inbound Channel \[page 346\]](#).
- SAP Customer Activity Repository retail applications bundle  
For more information, see [SAP Customer Activity Repository retail applications bundle \[page 352\]](#).

Available Integration Scenarios and their Business Partner Replication

Scenario	Business Partner Transfer	Set-Up Guide	Integration Technology
Marketing Collaboration with Sales Cloud	▶▶ <a href="#">SAP Cloud for Customer</a> ▶ <a href="#">SAP Marketing Cloud</a> ▶	<a href="#">Setting Up SAP Marketing Cloud Integration with SAP Cloud for Customer (1J9)</a>	SOAP
Lead2Cash	▶▶ <a href="#">SAP Master Data Service for Business Partners</a> ▶ <a href="#">SAP S/4HANA</a> , <a href="#">SAP Cloud for Customer</a> , <a href="#">SAP Marketing Cloud</a> etc. ▶	<a href="#">Integration of SAP Marketing Cloud with SAP Master Data Service for Business Partners</a>	SOAP
Marketing Collaboration with SAP CRM	▶▶ <a href="#">SAP CRM</a> ▶ <a href="#">SAP Marketing Cloud</a> ▶	<a href="#">Setting Up SAP CRM Integration with SAP Marketing Cloud (1NP)</a>	OData
Order Management	▶▶ <a href="#">SAP ERP</a> ▶ <a href="#">SAP Marketing Cloud</a> ▶ ▶▶ <a href="#">SAP S/4HANA</a> ▶ <a href="#">SAP Marketing Cloud</a> ▶	<a href="#">Setting Up SAP ERP Integration with SAP Marketing Cloud (1KW)</a> <a href="#">Setting Up SAP S/4HANA Cloud Integration with SAP Marketing Cloud (1UG)</a>	OData

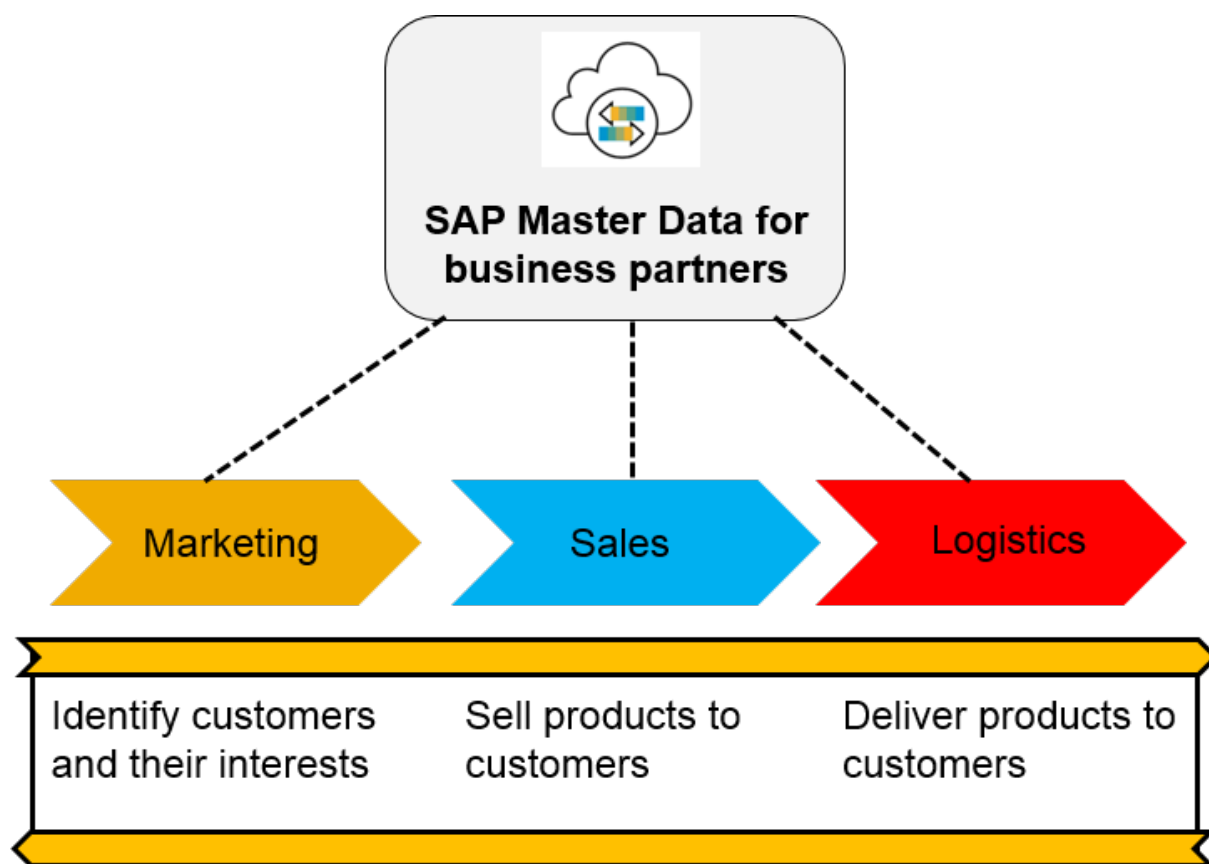
## 4.5.1.1 SAP Master Data Service for Business Partners

Integration of SAP Marketing Cloud with SAP Master Data service for business partners

You can use SAP Master Data service for business partners for storing all your business partners and their master data.

With the integration of SAP Master Data service for business partners, you can not only replicate business partners and their relationship to SAP Marketing Cloud, but also to other systems like SAP Cloud for Customer.

With SAP Master Data service for business partners, you can also leverage the multiple relationships a contact can have within or across different companies.



For the integration setup with SAP Master Data service for business partners, you have the following two options:

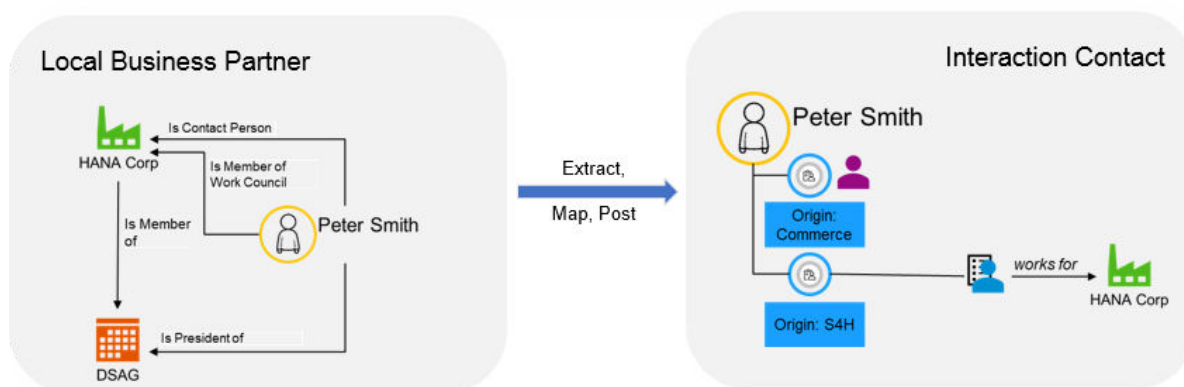
1. You are guided by the Cloud Integration Automation Service (see the SAP Help Portal at [Introduction to Cloud Integration Automation Service](#)) by making use of the [Maintenance Planner](#).
2. You use the integration setup of SAP Master Data service for business partners with SAP Marketing Cloud (see [SAP Master Data for business partners Integration with SAP Marketing Cloud](#), or [Integration of SAP Marketing Cloud with SAP Master Data service for business partners](#)).

This integration supports advanced B2B Marketing using business partner and business partner relationships between corporate contacts and accounts. It loads the business partner data from an SAP S/4HANA system or

an SAP Cloud for Customer system into SAP Marketing Cloud, extracting the local business partners and business partner relationships into interaction contacts. In addition, the following features are offered:

- Postal address of a contact within a company that is updated on a daily basis. If a contact has more than one relationship defined, for example marketing lead and president of DSAG, the relationship with the highest ID is selected.
- Extraction of future time slices of time-dependent business partner data. This allows you to target marketing activities like campaigns only at contacts that are relevant at that specific point in time.
- Several Extensibility Options (see [Extensibility for SAP Business Partner Integration](#))

The following graphic shows an overview of the integration:



For a full description of how to set up an integration with SAP S/4HANA Cloud, see [Setting Up SAP S/4HANA Cloud Integration with SAP Marketing Cloud \(1UG\)](#) or the corresponding section of the Cloud Integration Automation Service in [Maintenance Planner](#).

For a full description of how to set up an integration with SAP Cloud for Customer, see [Setting Up SAP Marketing Cloud Integration with SAP Cloud for Customer \(1J9\)](#) or the corresponding section of the Cloud Integration Automation Service in [Maintenance Planner](#).

## 4.5.1.2 Presales / Sales

Set up the integration of a sales system with SAP Marketing Cloud

SAP Marketing Cloud can be integrated with SAP CRM, SAP Cloud for Customer, or an external sales system via SAP Cloud Integration.

### Parallel Integration of Presales/Sales Systems

#### i Note

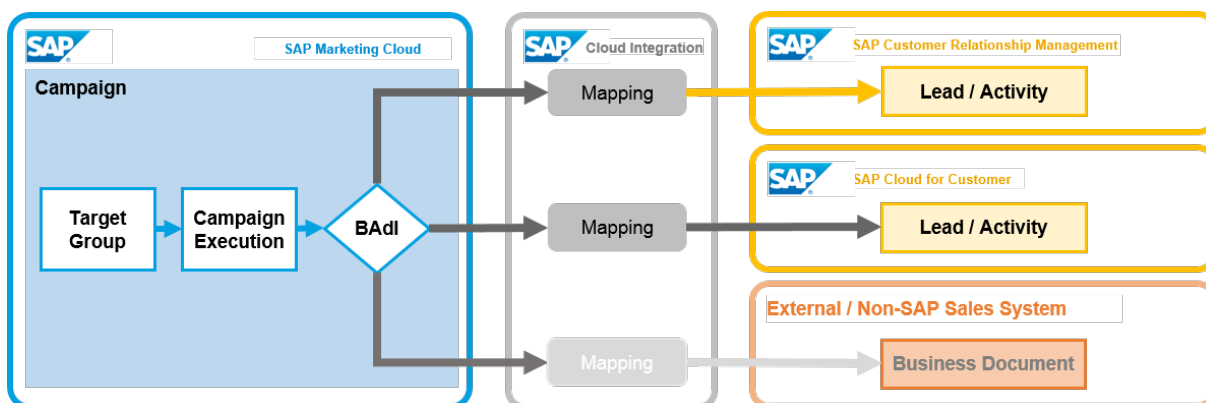
You can set up the integration with SAP CRM, SAP Cloud for Customer, and an external sales system simultaneously, but only with one system of a target system type at a time. That is, you can integrate one system for SAP CRM, one for SAP Cloud for Customer, and one external sales system.

In case you have activated more than one communication arrangement, that is, you have configured more than one target system, the Business Add-In (BAI) *Lead Management: Determine Target System Type* is performed. With the *Custom Logic* app, you can implement the BAI. You define the target system type (either SAP\_C4C or SAP\_CRM or SALES\_EXT) depending on different attributes of the contact that is

currently in process. The BAdI is performed once for each member of the target group. That is, you define in which target system the correspondings leads or activities are created. For more information, see [Custom Logic](#).

The following graphic provides you with an overview of the parallel integration options:

- Target Group members can be of different origin, that is, some from SAP Cloud for Customer, others from SAP Customer Relationship Management or External Sales Systems.
- BAdI:  
During campaign execution, the BAdI determines the target system in which a lead and/or activity shall be created according to the implemented default or custom logic.
- The BAdI default implementation has to be replaced by custom logic.



Integration takes place in two directions:

- Inbound, from presales / sales to marketing
- Outbound, from marketing to presales / sales

For more information, see:

- [Integration with SAP CRM - Inbound Channel \[page 342\]](#)
- [Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)
- [Integration with External Sales Systems - Inbound Channel \[page 346\]](#)
- [Integration with SAP CRM - Outbound Channel \[page 354\]](#)
- [Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#).
- [Integration with External Sales Systems - Outbound Channel \[page 374\]](#)

For more information about the integration setup of SAP Cloud for Customer with SAP Marketing Cloud, see the Integration Guide on SAP API Hub at [SAP Cloud for Customer Integration with SAP Marketing](#), or [Purpose](#).

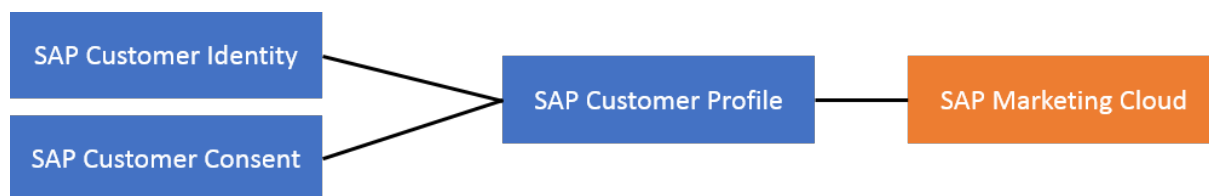
For more information about the integration setup of SAP Marketing Cloud and SAP CRM, see the Set-Up Instructions on SAP API Hub at [SAP CRM Integration](#) or [Overview](#).

## 4.5.1.2.1 SAP Customer Data Cloud and SAP Marketing Cloud

SAP Customer Data Cloud is a provider of customer identity management. With its solutions SAP Customer Identity, SAP Customer Consent, and SAP Customer Profile you can collect and replicate contact profiles to SAP Marketing Cloud.

The integration enables you to add SAP Customer Data Cloud first-party, permission-based user information into the SAP Marketing Cloud platform, and turn it into actionable data for audience segmentation, targeted marketing, and more. The integration is based on exporting data from SAP Customer Data Cloud, using the corresponding integration package on SAP Cloud Integration (see the SAP API Hub at [SAP Customer Data Cloud Integration with SAP Marketing Cloud](#)). SAP Customer Identity manages the customers, SAP Customer Consent manages the consent, SAP Customer Profile connects this data into various channels, including SAP Marketing Cloud.

For more information, see the SAP Customer Data Cloud documentation on the SAP Help Portal at [SAP Marketing Cloud Integration with SAP Customer Data Cloud](#).



## 4.5.1.2.2 Integration with SAP Cloud for Customer - Inbound Channel

Data transfer from sales to marketing.

By integrating SAP Marketing Cloud, and SAP Cloud for Customer, the bridge between marketing and sales is built so that processes can be harmonized across marketing and sales channels. With sharing the same business partner, and business document data, Marketing is able to deeper support sales in the process of converting potential buyers and interested persons to real buyers. The integration between SAP Marketing Cloud, and SAP Cloud for Customer supports the following business scenarios:

- Lead Transfer
- Call Qualification
- Marketing-Driven Sales Enablement

In addition, the transfer of campaign data is possible for started campaigns.

## Data Replication from Sales to Marketing (Inbound)

From SAP Cloud for Customer, the system replicates the following data to SAP Marketing Cloud via initial and delta load:

- Business Partners
  - Contacts
  - Accounts
  - Individual Customers
  - Business Partner Relationships
- Business Documents
  - Leads including product items
  - Opportunities including product items, and product categories
  - Activities of type *Phone Call, Appointment, Task, and Visit*

### i Note

In SAP Marketing Cloud, interactions are stored for tasks. So, from SAP Cloud for Customer, the replication of marketing-driven tasks is enabled. Tasks created in the sales system, cannot be replicated.

- Marketing Attribute Categories
  - Master data (marketing attribute sets and marketing attributes)
  - Assignments of marketing attribute sets including attribute values and attribute value descriptions to business partners  
For more information about the transfer of marketing attributes from SAP Cloud for Customer to marketing, see [Transferring Marketing Attributes \[page 341\]](#).
- Marketing Permissions  
In marketing, the permissions are stored on contact and account level. The replication takes place once via initial load. For more information about the processing of marketing permissions, see [Permission Marketing](#).
- Custom Fields for Interactions  
Custom fields for interactions that are created in SAP Cloud for Customer can also be transferred to marketing. To learn more about how to create custom fields in SAP Cloud for Customer, see [How to Extend SAP Cloud Integration](#). For more information on how to extend SAP Marketing Cloud, see the extensibility guide, [Custom Fields for the Integration with SAP Cloud for Customer](#).
- Custom Fields for Business Partners and Interaction Contacts

In marketing, you can analyze data imports via the [Import Monitor \[page 404\]](#).

## Integration

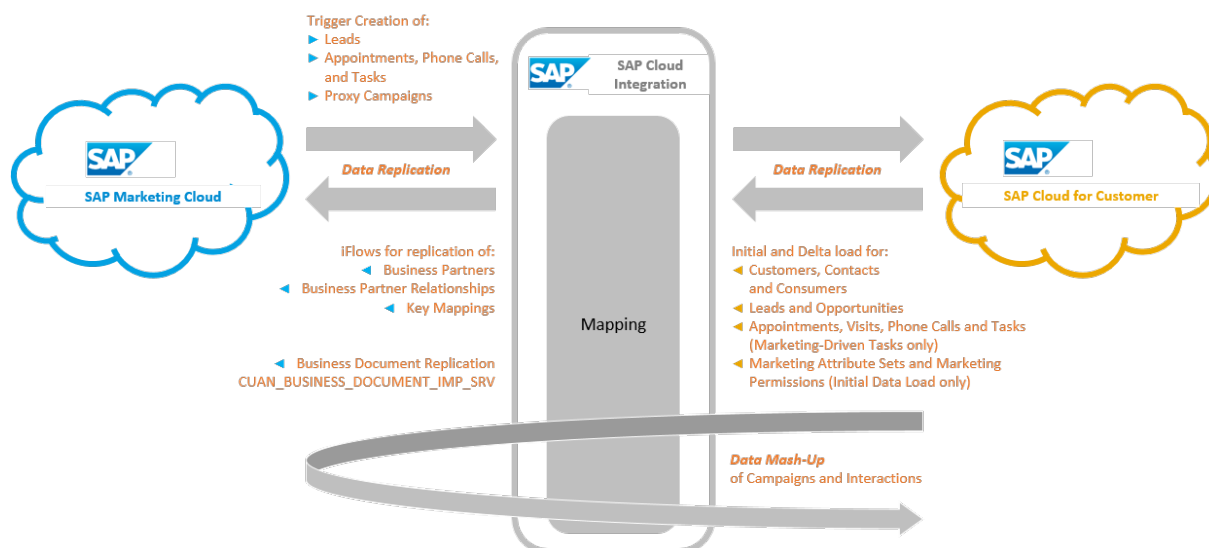
You can set up the integration between SAP Marketing Cloud with SAP Cloud for Customer via SAP Cloud Integration.

For the setup you have the following two options:

1. You are guided by the Cloud Integration Automation Service (see the SAP Help Portal at [Introduction to Cloud Integration Automation Service](#)) by making use of the [Maintenance Planner](#).

- You use the integration setup of SAP Cloud for Customer with SAP Marketing Cloud (see [SAP Cloud for Customer Integration with SAP Marketing](#), or [Setting Up SAP Marketing Cloud Integration with SAP Cloud for Customer \(1J9\)](#)).

The following figure shows an overall integration overview:



### Integration of SAP Marketing Cloud with Presales or Sales for Marketing-Driven Leads, Call Center Campaigns, and Activities

For more information about the campaign-based lead creation process, see [Handling Leads](#).

For more information about the process of lead creation via a call center campaign, see [Telephone Campaigns in SAP Cloud for Customer](#).

For more information about the process of campaign-based activity creation, see [Handling Activities](#).

## Navigation, and Display Options

- In SAP Marketing Cloud
 

Information about leads, activities, and phone calls created in SAP Cloud for Customer via marketing campaigns, and replicated data from SAP Cloud for Customer is displayed in marketing on the corresponding contact, or account in [Interactions](#).

Leads and opportunities, both marketing-driven and sales-created, are additionally visible on contact, and account level under [Leads](#) [Sales Pipeline](#). The system provides you with navigation links to SAP Cloud for Customer for the corresponding objects.

On the related campaign, the system displays the marketing-driven objects created in SAP Cloud for Customer under [Performance](#).

In the Lead Dashboard, KPIs for marketing-driven, and sales-created objects are displayed.
- In SAP Cloud for Customer
 

In SAP Cloud for Customer, you can display campaign data. For more information, see [Sales Insights on Marketing Campaigns \[page 363\]](#)



## Related Information

[Lead Campaigns](#)  
[Marketing-Driven Sales Enablement](#)  
[Call Qualification](#)  
[Displaying Lead Information for Contacts](#)  
[Displaying Lead Information for Accounts](#)  
[Lead Dashboard](#)  
[Marketing Attribute Categories](#)  
[Transferring Marketing Attributes \[page 341\]](#)  
[Business Documents \[page 661\]](#)

### 4.5.1.2.2.1 Business Partner Replication

#### i Note

You can replicate the business partners in two different ways:

1. The replication can be done based on SOA.  
For more information about SOA-based replication of Business Partners and Business Partner Relationships, see [Replicating Business Partner with SOAP](#).
2. You can use the integration with SAP Master Data service for business partners. For more information, see [SAP Master Data Service for Business Partners \[page 335\]](#).

You can also extend the SAP business partner integration. For more information, see [Extensibility for SAP Business Partner Integration](#).

### 4.5.1.2.2.2 Transferring Marketing Attributes

Transfer of marketing attributes and business partner assignments from SAP Cloud for Customer to SAP Marketing Cloud.

Marketing attributes and their assignments to business partners are transferred from SAP Cloud for Customer to SAP Marketing Cloud in two steps, as described in the following table:

Transfer of Marketing Attributes

Transfer of	From SAP Cloud for Customer	To SAP Marketing Cloud
Master Data	<ul style="list-style-type: none"><li>• Marketing Attribute Sets</li><li>• Marketing Attributes</li></ul>	Marketing Attribute Categories

Transfer of	From SAP Cloud for Customer	To SAP Marketing Cloud
Business Partner Assignments	Marketing Attributes Sets, including: <ul style="list-style-type: none"> <li>Marketing Attribute Values</li> <li>Marketing Attribute Value Descriptions</li> </ul>	Marketing Attribute Values

## Integration

In Marketing, marketing attributes categories and marketing attribute values are visible in *Personal Data* of contacts, accounts, or individual customers. Marketing attribute categories, and marketing attribute values can be used in segmentation.

### i Note

- Marketing attribute categories always have a text in the system language. If no text is transferred from SAP Cloud for Customer the system automatically creates a text in the system language from the ID.
- Ensure that all attributes in SAP Cloud for Customer are named differently. Attributes with the same name cause an error that can be monitored in the [Import Monitor \[page 404\]](#).
- To prevent from overwriting attribute values, do not use the same attribute in different attribute sets in SAP Cloud for Customer.

Changes of master data and business partner assignments in Sales are automatically transferred to Marketing.

## Related Information

[Segmentation](#)  
[Contacts](#)

### 4.5.1.2.3 Integration with SAP CRM - Inbound Channel

Data transfer from sales to marketing.

By integrating SAP Marketing Cloud with SAP CRM, you can trigger the creation of leads and activities in SAP CRM via a marketing campaign. Furthermore this integration enables the replication of business partner, and business document data from SAP CRM to SAP Marketing Cloud.

You can set up the integration of SAP CRM with SAP Marketing Cloud via SAP Cloud Integration.

The integration between SAP Marketing Cloud, and SAP CRM supports the following business scenarios:

- Lead Transfer
- Marketing-Driven Sales Enablement

## Data Replication from Sales to Marketing (Inbound)

From SAP CRM, the system replicates the following data to SAP Marketing Cloud via initial and delta load:

- Business Partners
  - Contacts
  - Accounts
  - Individual Accounts
- Business Documents
  - Leads
  - Opportunities
  - Activities of type *Planned Call*, *Appointment*, and *Task* (marketing-driven tasks only)

Leads, and activities created in SAP CRM via marketing campaign are created as business documents in SAP CRM.

### i Note

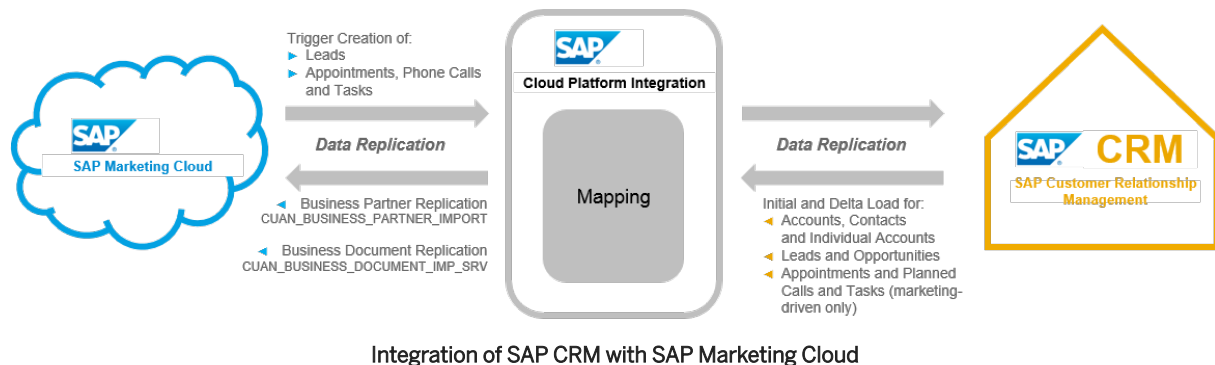
In SAP Marketing Cloud, interactions are stored for tasks. So, from SAP CRM, the replication of marketing tasks is enabled. Tasks created in the sales system, are not replicated.

- Marketing Attributes  
For more information about the configuration, see [Configuration of CRM Marketing Attribute Replication Integration Flow](#).
- Custom fields  
Custom fields created in SAP CRM can also be transferred to marketing. To learn more about how to create custom fields in SAP CRM, see [How to Extend SAP CRM: Purpose of this Document](#). For more information on how to extend SAP Marketing Cloud, see the extensibility guide, [Custom Fields for the Integration with SAP CRM](#).
- Marketing Permissions  
In marketing, the permissions are stored on contact and account level. The replication takes place once via initial load. For more information about the processing of marketing permissions, see [Permission Marketing](#).  
For information on how to set up the transfer of marketing permissions from SAP CRM, see [Initial Load of Marketing Permissions from SAP Customer Relationship Management to SAP Marketing Cloud](#), in the SAP Community.

In marketing, you can analyze data imports by the [Import Monitor \[page 404\]](#).

## Integration

The following figure shows an overall integration overview:



For a full description of the integration setup of SAP Marketing Cloud and SAP CRM, see the Set-Up Instruction on SAP API Hub at [SAP CRM Integration](#) or [Overview](#).

## Navigation, and Display Options

In SAP Marketing Cloud, information about leads, and activities created in SAP CRM via marketing campaigns, and replicated data from SAP CRM is displayed in Marketing on the corresponding contact, or account in [Interactions](#).

Leads and opportunities, both marketing-driven and sales-created, are additionally visible on contact, and account level under [Leads](#) [Sales Pipeline](#). The system provides you with navigation links to SAP CRM for the corresponding objects.

On the related campaign, the system displays the marketing-driven objects created in SAP CRM under [Performance](#).

In the Lead Dashboard, KPIs for marketing-driven, and sales-created objects are displayed.

## Related Information

- [Lead Campaigns](#)
- [Marketing-Driven Sales Enablement](#)
- [Displaying Lead Information for Contacts](#)
- [Displaying Lead Information for Accounts](#)
- [Lead Dashboard](#)
- [Handling Leads](#)
- [Handling Activities](#)
- [Business Documents \[page 661\]](#)

### 4.5.1.2.3.1 Business Partner Replication

The replication of business partners and relations from SAP CRM to SAP Marketing Cloud is processed by the following principle:

- Initial load of all SAP CRM accounts, contacts, individual accounts person to SAP Marketing Cloud according to your selection with the initial setup of the integration.
- Delta load of new SAP CRM accounts, contacts, individual accounts to SAP Marketing Cloud as soon as those objects are created.
- Delta load of changed SAP CRM accounts, contacts, individual accounts to SAP Marketing Cloud as soon as those objects are changed.

### 4.5.1.2.3.2 Transferring Marketing Attributes

Transfer of marketing attributes and business partner assignments from SAP CRM to marketing.

Marketing attributes and their assignments to business partners are transferred from SAP CRM to SAP Marketing Cloud in two steps, as described in the following table:

Transfer of Marketing Attributes

Transfer of	From SAP CRM	To SAP Marketing Cloud
Master Data	<ul style="list-style-type: none"><li>• Marketing Attributes</li></ul>	Marketing Attribute Categories
Business Partner Assignments	Marketing Attributes, including: <ul style="list-style-type: none"><li>• Marketing Attribute Values</li><li>• Marketing Attribute Value Descriptions</li></ul>	Marketing Attribute Values

## Integration

In Marketing, marketing attributes categories and marketing attribute values are visible in *Personal Data* of contacts, accounts, or individual customers. Marketing attribute categories, and marketing attribute values can be used in segmentation.

Changes of master data and business partner assignments in Sales are automatically transferred to Marketing.

## 4.5.1.2.4 Integration with External Sales Systems - Inbound Channel

Data transfer from external sales systems to marketing.

By integrating SAP Marketing Cloud with external sales systems, the bridge between marketing and sales is built so that processes can be harmonized across marketing and sales channels.

### ⚠ Caution

For the integration of SAP Marketing Cloud with external sales systems, we do not deliver standard content.

### i Note

You can also use the integration with Salesforce offered by Advantco International LLC. For more information, see [SAP Marketing Cloud Integration with Salesforce](#) .

## Inbound Processes from an External Sales System to Marketing

Depending on the data model of an external sales system, the following data can be replicated to SAP Marketing Cloud.

- Business Partners
- Business Documents, such as leads, opportunities, or activities

Business documents can be imported from external sales systems to SAP Marketing Cloud via the standard OData service *Import of Business Documents (Interactions)* (CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV).

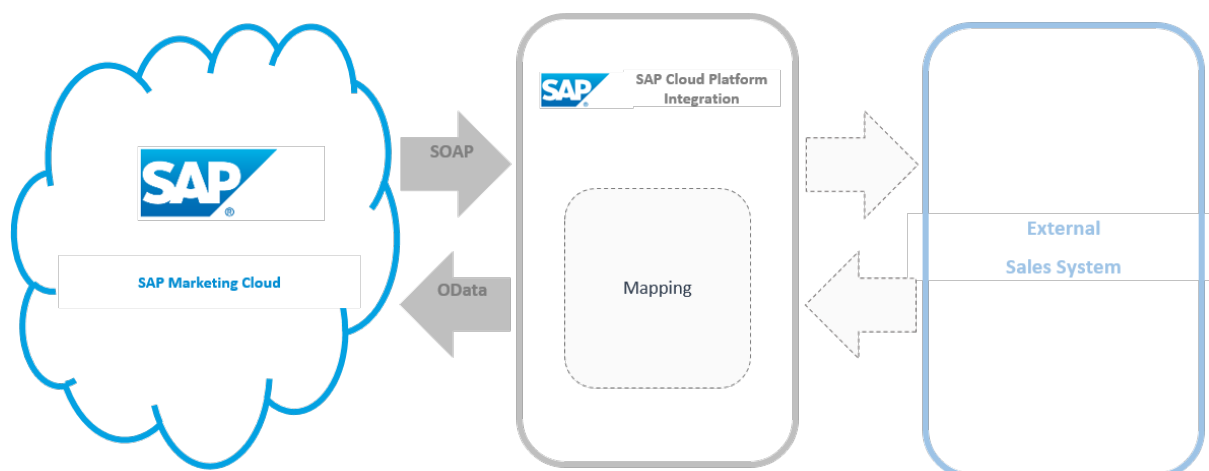
Business partners can be imported from external sales systems to SAP Marketing Cloud via the standard OData services *Marketing - Interaction Contacts* (API\_MKT\_INTERACTION\_CONTACT\_SRV), *Marketing - Contacts* ( API\_MKT\_CONTACT\_SRV), and *Marketing - Corporate Accounts* (API\_MKT\_CORPORATE\_ACCOUNT\_SRV). For more information on OData services, see SAP API Business Hub under [Artifacts](#) .

In marketing, you can analyze data imports via the [Import Monitor \[page 404\]](#).

## Integration

You can set up the integration between SAP Marketing Cloud with an external sales system via SAP Cloud Integration.

The following figure shows a possible overall integration overview based on a customer-owned integration setup:



Integration of SAP Marketing Cloud with External Sales Systems

#### 4.5.1.2.4.1 Setting Up the Connection Between Marketing and an External Sales System

Connect SAP Marketing Cloud with an external sales system.

Before doing the configuration in SAP Marketing Cloud, you need the administrator business user, which contains the business catalog SAP\_CORE\_BC\_COM (Communication Management), for example the business role SAP\_BR\_ADMINISTRATOR (Administrator).

As an administrator, you maintain the setup via the following apps under *Communication Management*:

- Maintain Communication User  
For more information, see [Creating a Communication User for Inbound Communication \[page 376\]](#).
- Communication Systems  
For more information, see [Setting Up a Communication System for the Integration of an External Sales System \[page 377\]](#).
- Communication Arrangements  
For more information, see [Setting Up a Communication Arrangement for the Integration of an External Sales System \[page 378\]](#).

### 4.5.1.3 Order Management

[Order Management Data Replication to SAP Marketing Cloud \[page 348\]](#)

Integration with SAP S/4HANA Cloud and SAP S/4HANA

[Integration with SAP ERP \[page 349\]](#)

Integration of SAP ERP with SAP Marketing Cloud using SAP Cloud Integration

## 4.5.1.3.1 Order Management Data Replication to SAP Marketing Cloud

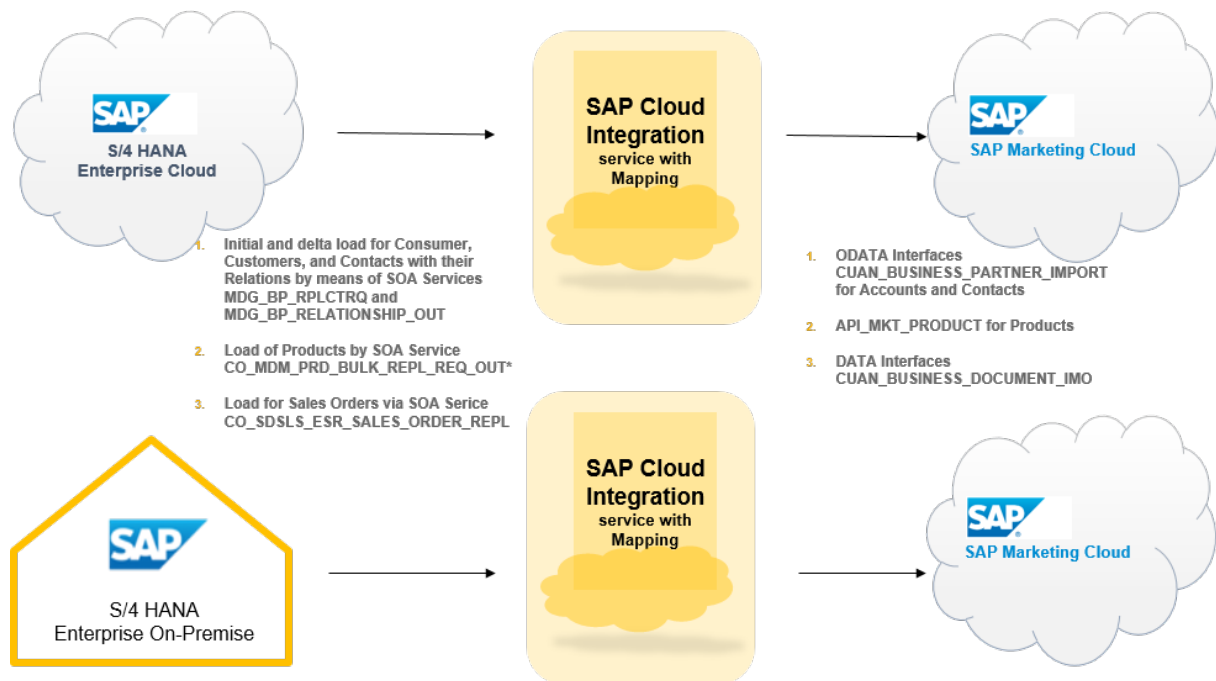
Integration with SAP S/4HANA Cloud and SAP S/4HANA

By integrating SAP Marketing Cloud and SAP S/4HANA Cloud or SAP S/4HANA, customers can use customer and contact data, as well as their relations and address data for Marketing campaigns.

Using the SAP Cloud Integration as standard middleware content, customers benefit from SAP cloud integration standards for security, performance, data integrity, and robustness.

On SAP S/4HANA-side the exchange is done via Business Partner SOA services MDGK\_BP\_RPLCTRQ and MDG\_BP\_RELATIONSHIP\_OUT.

On Marketing-side the interface CUAN\_BUSINESS\_PARTNER\_IMPORT is used, see [Import Business Partners \[page 574\]](#).



\*For integration of products via SOA Service CO\_MDM\_PRD\_BULK\_REPL\_REQ\_OUT S/4HANA FSP02 is required

### i Note

If you have an integration of SAP S/4HANA Enterprise with SAP Marketing Cloud in place, deletion or the end of purpose of a customer or corporate contact on the SAP S/4HANA Enterprise side is not automatically replicated to SAP Marketing Cloud. To ensure deletion in SAP Marketing Cloud, you have to do the following:

1. Run the application job *Flag Contact IDs for Deletion* with the following parameters: specify the *Origin of Contact* and the *ID of Contact*, and select the parameter *Origin IDs with Dep. IDs Too*.
2. Finally, you also have to run the application job *Delete Flagged Contact IDs*.



For the integration setup with SAP S/4HANA Cloud, you have the following two options:

1. You are guided by the Cloud Integration Automation Service (see the SAP Help Portal at [Introduction to Cloud Integration Automation Service](#)) by making use of the [Maintenance Planner](#).
2. You use the integration setup of SAP S/4HANA Cloud with SAP Marketing Cloud (see [SAP S/4HANA Cloud Integration with SAP Marketing Cloud](#), or [Setting Up SAP S/4HANA Cloud Integration with SAP Marketing Cloud \(1UG\)](#)).

For the integration setup with SAP S/4HANA, you can find a full description: [Setting Up SAP S/4HANA Integration with SAP Marketing Cloud \(23L\)](#).

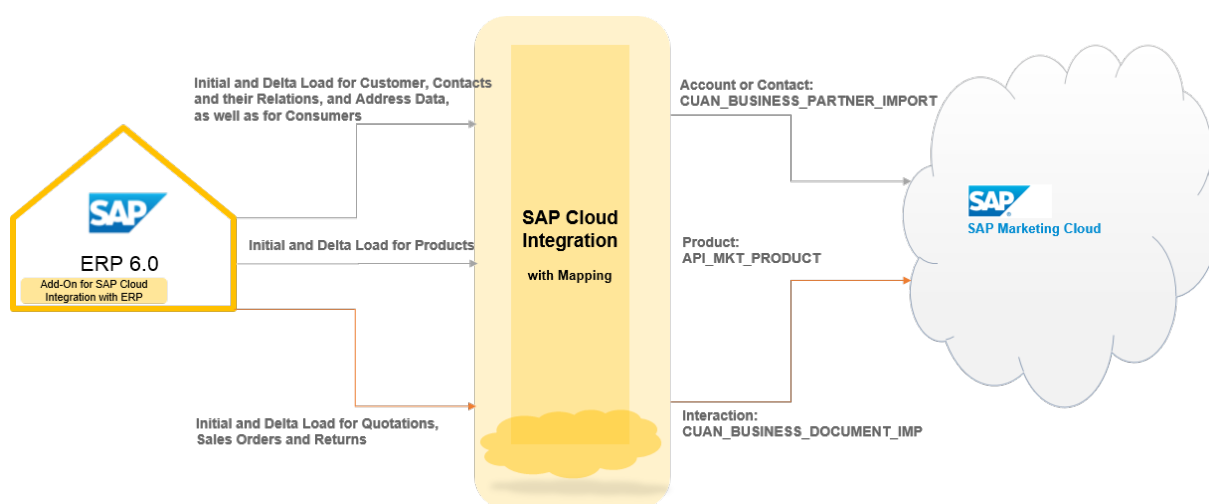
### 4.5.1.3.2 Integration with SAP ERP

Integration of SAP ERP with SAP Marketing Cloud using SAP Cloud Integration

By integrating SAP Marketing Cloud and SAP ERP, customers can use valuable data from the on premise SAP ERP system in Business-To-Business (B2B) and Business-To-Customer (B2C) business scenarios. This includes customer, contact and product data, as well as sales volume data like quotes, orders, and returns. Customers can, for example, launch campaigns for the customers and contacts, or use information on sales volume to determine the best customers for campaigns.

If specific contacts, customers, and consumers are set to **blocked** in the source system, the respective customers, contacts, and consumers are flagged with **end of purpose** in the marketing system.

They are then no longer visible and cannot be used in business processes in marketing.



Integration is done using SAP Cloud Integration middleware and the OData service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV, see [Import Business Partners \[page 574\]](#).

Extensibility enables customers to transfer additional data of customers, contacts, and sales orders from SAP ERP to SAP Marketing Cloud. Additional fields for this purpose must be enabled in the *Custom Fields* app for the service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV.

You can find a full description of how to set up this integration in: [Setting Up SAP ERP Integration with SAP Marketing Cloud \(1KW\)](#).

## 4.5.1.4 Integrating Service Tickets

Replicate Service Tickets from SAP Service Cloud to SAP Marketing Cloud.

By replicating service tickets from SAP Service Cloud to interactions in SAP Marketing Cloud, marketers can use them as signals in marketing. Via the replicated interactions, actions can be triggered in marketing, such as inviting people to surveys, safeguarding the overall customer satisfaction by awards or compensations. Furthermore, they can be used as interruption pointer for campaigns. Service tickets in marketing are important to understand your customer's business needs, for example to trigger campaigns for a new product launch or product replacements.

### Prerequisites

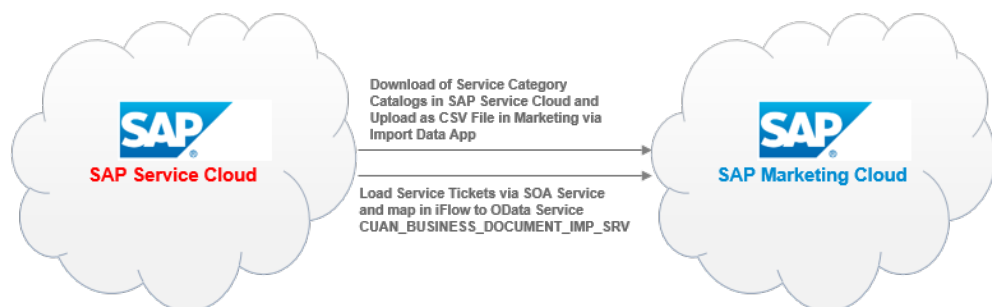
The integration of service tickets requires the following preliminary steps:

- Perform the integration scenario **SAP Cloud for Customer Integration with SAP Marketing Cloud**. For more information about the integration setup of SAP Cloud for Customer with SAP Marketing Cloud, see [SAP Cloud for Customer Integration with SAP Marketing Cloud](#). For setup instructions, see [Service Ticket Integration](#).
- In the *Manage Interaction Content* configuration app, define a new *Interaction Channel* for service tickets, named *Service*. Assign the *Communication Medium* `BUSINESS_DOCUMENT` and the *Interaction Type* `SERVICE_TICKET` to the interaction channel *Service*. For more information, see [Managing Interaction Content](#).

### Process

Customers can submit requests for service, for example to address a problem in SAP Service Cloud. These service tickets can be replicated to SAP Marketing Cloud and saved in interactions for further marketing actions.

The following graphic provides you with an overview about the necessary steps.



To integrate service tickets in SAP Marketing Cloud, perform the following steps:

- Download the service category catalog in SAP Service Cloud via Microsoft Excel® file download and upload it in SAP Marketing Cloud as product hierarchy using the [Data File Load](#) app. For more information about the import, see [Data File Load](#).
- Deploy the standard iFlow on SAP Cloud Integration that service tickets are mapped and loaded as interactions into SAP Marketing Cloud.

The following attributes of service tickets are replicated to marketing:

Mapping of Service Ticket Attributes from SAP Service Cloud to Properties of OData Service CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV of SAP Marketing Cloud

Service Ticket Attribute in SAP Service Cloud	CUAN_BUSINESS_DOCUMENT_IMP_SRV Property in SAP Marketing Cloud
Ticket Buyer Party	Contact ID
Ticket Creation Date Time	Time Stamp
Ticket ID	External ID
Ticket Priority Code	Interaction Priority
The <i>Internal Object Type</i> is not mapped to an attribute in SAP Service Cloud. It has the hard-coded value <code>SERVICE_TICKET</code> . In the integration scenario <b>SAP Cloud for Customer Integration with SAP Marketing Cloud</b> , this value can be adapted in the corresponding iFlow <a href="#">Replicate Service Ticket to SAP Marketing</a> .	Internal Object Type
The <i>External Object Type</i> is not mapped to an attribute in SAP Service Cloud. It has the hard-coded value <code>C4S_SERVICE_TICKET</code> .	External Object Type
Confirmation Issuing Status Code	Status Code
Information Life Cycle Status Code	External Status Code
Service Category	Product Category of type <b>Process Category</b>
Incident Category	Product Category of type <b>Incident Category</b>
Object Category	Product Category of type <b>Object Category</b>
Cause Category	Product Category of type <b>Cause Category</b>
Resolution Category	Product Category of type <b>Solution Category</b>
<ul style="list-style-type: none"> <li>○ Ticket Completion Time Point</li> <li>○ Ticket Creation Date Time</li> </ul>	Interaction Processing Duration  Is calculated from TicketCompletionTimePoint minus CreationDateTime and converted into seconds.

For more information about the structure of this service, see [Structure of OData Service CUAN\\_BUSINESS\\_DOCUMENT\\_IMP\\_SRV \[page 664\]](#).

Product hierarchies can be classified in terms of their usage in marketing, that is, whether they are defined for product categories or service categories.

The service categories of the service tickets are included as product categories in SAP Marketing Cloud. They can be classified with category types, that is, to distinguish service processes, root causes, or service solutions.

For more information about the attributes on product hierarchies and product categories, see [Product Hierarchies and Categories \[page 604\]](#).

### 4.5.1.5 SAP Customer Activity Repository retail applications bundle

The integration of SAP Customer Activity Repository enables you to import of POS transactions, such as sales order and sales returns as interactions.

You can use then this data in various process steps like segmentation, customer fact sheet, product recommendation, and trigger based campaigns.

As a prerequisite you have uploaded the relevant master data to SAP Marketing Cloud, such as contacts, products, and marketing locations.

#### i Note

Keep in mind that this integration works only with SAP Customer Activity Repository retail applications bundle (CARAB) 2.0 FP1.

For more information, see:

- [SAP Customer Activity Repository applications bundle](#)
- [Integration with SAP ERP \[page 349\]](#)
- OData services
  - [Products \[page 582\]](#)
  - [Product Hierarchies and Categories \[page 604\]](#)
  - [Contacts \[page 412\]](#)
  - [Import Business Partners \[page 574\]](#)

## 4.5.2 Sales Automation (Outbound)

Set up the integration of a sales system with SAP Marketing Cloud

SAP Marketing Cloud can be integrated with SAP CRM, SAP Cloud for Customer, or an external sales system via SAP Cloud Integration.

### Parallel Integration of Presales/Sales Systems

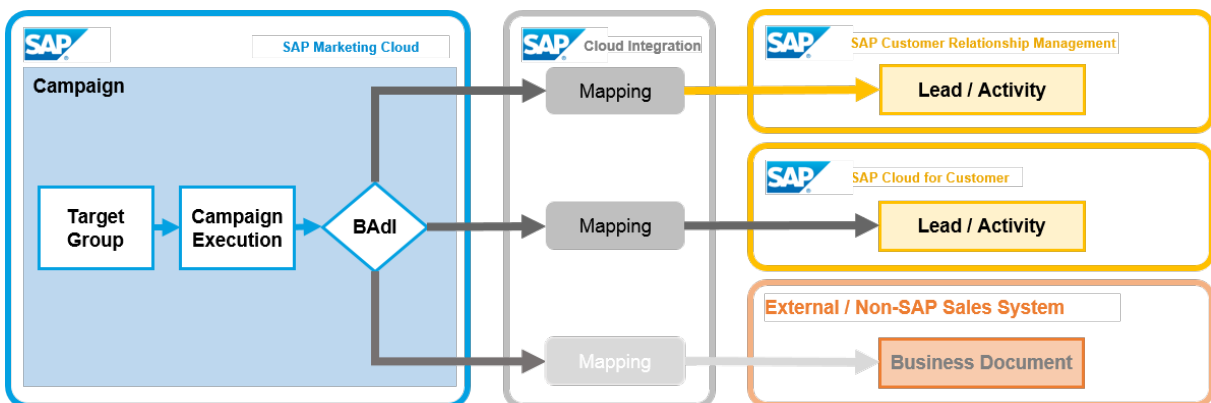
## i Note

You can set up the integration with SAP CRM, SAP Cloud for Customer, and an external sales system simultaneously, but only with one system of a target system type at a time. That is, you can integrate one system for SAP CRM, one for SAP Cloud for Customer, and one external sales system.

In case you have activated more than one communication arrangement, that is, you have configured more than one target system, the Business Add-In (BAI) *Lead Management: Determine Target System Type* is performed. With the *Custom Logic* app, you can implement the BAI. You define the target system type (either SAP\_C4C or SAP\_CRM or SALES\_EXT) depending on different attributes of the contact that is currently in process. The BAI is performed once for each member of the target group. That is, you define in which target system the correspondings leads or activities are created. For more information, see [Custom Logic](#).

The following graphic provides you with an overview of the parallel integration options:

- Target Group members can be of different origin, that is.. some from SAP Cloud for Customer, others from SAP Customer Relationship Management or External Sales Systems.
- BAI:  
During campaign execution, the BAI determines the target system in which a lead and/or activity shall be created according to the implemented default or custom logic.
- The BAI default implementation has to be replaced by custom logic.



Integration takes place in two directions:

- Inbound, from presales / sales to marketing
- Outbound, from marketing to presales / sales

For more information, see:

- [Integration with SAP CRM - Inbound Channel \[page 342\]](#)
- [Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)
- [Integration with External Sales Systems - Inbound Channel \[page 346\]](#)
- [Integration with SAP CRM - Outbound Channel \[page 354\]](#)
- [Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#).
- [Integration with External Sales Systems - Outbound Channel \[page 374\]](#)

For more information about the integration setup of SAP Cloud for Customer with SAP Marketing Cloud, see the Integration Guide on SAP API Hub at [SAP Cloud for Customer Integration with SAP Marketing](#), or [Purpose](#).

For more information about the integration setup of SAP Marketing Cloud and SAP CRM, see the Set-Up Instructions on SAP API Hub at [SAP CRM Integration](#) or [Overview](#).

## 4.5.2.1 Integration with SAP CRM - Outbound Channel

Data transfer from marketing to sales.

### Data Replication from Marketing to Sales (Outbound)

From SAP Marketing Cloud, the system triggers the creation of the following data in SAP CRM:

- Business Partners  
If the business partner is not known in SAP CRM, the system creates a business partner, account, contact, individual customer, in SAP CRM.

#### i Note

Only during the process of lead creation via marketing campaign, business partners are created in sales. Ensure that business partners in Marketing that are part of the lead creation process, the `Country` is filled. Otherwise, no business partner, and no lead is created in SAP CRM.

The address of a contact is not replicated into SAP CRM. The system uses the standard address of the related account.

Before creating business documents, the system creates accounts, and contacts in SAP CRM with the origin ID from Marketing.

- Business Documents
  - Leads  
The system enriches the transferred marketing data during each lead transfer. For more information see [Augmented Lead Context \[page 355\]](#).
  - Activities of type *Planned Call*, *Appointment*, and *Task*  
The system enriches the transferred marketing data during each activity transfer. For more information see [Augmented Activity Context \[page 359\]](#).

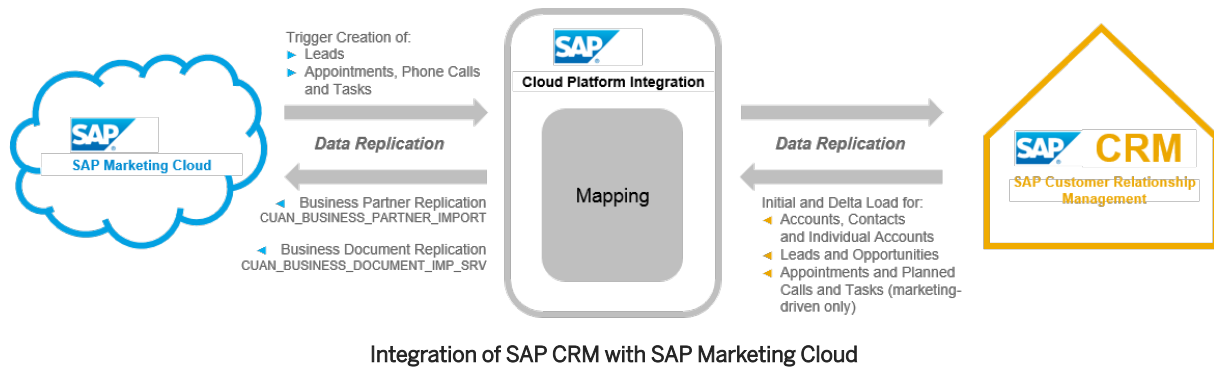
#### i Note

For leads and activities transferred to SAP CRM, the system also sends out the assigned marketing area. As no mapping is performed in SAP Cloud Integration, the marketing area gets lost during the confirmation process from SAP CRM to SAP Marketing Cloud.

- Custom Fields Created in Marketing.  
Custom fields created in marketing can also be transferred to SAP CRM. For more information on how to extend SAP Marketing Cloud, see the extensibility guide, [Custom Fields for the Integration with SAP CRM](#). To make marketing-created custom fields available in SAP CRM, you have to ensure that the standard mapping is extended in SAP Cloud Integration.

## Integration

The following figure shows an overall integration overview:



For a full description of the integration setup of SAP Marketing Cloud and SAP CRM, see the Set-Up Instruction on SAP API Hub at [SAP CRM Integration](#) or [Overview](#).

## Related Information

[Lead Campaigns](#)  
[Marketing-Driven Sales Enablement](#)  
[Displaying Lead Information for Contacts](#)  
[Displaying Lead Information for Accounts](#)  
[Lead Dashboard](#)  
[Handling Leads](#)  
[Handling Activities](#)  
[Business Documents \[page 661\]](#)

### 4.5.2.1.1 Augmented Lead Context

Enhancement of lead information to be transferred to sales.

Via augmented lead context, sales representatives are provided with context-related lead information that helps to prioritize the sequence of lead processing, and that allows more specified follow-up activities.

Augmented lead context is relevant for the integration with SAP CRM and SAP Cloud for Customer.

For a lead triggered by marketing, the sales representative wants to know the context of this creation, that can be a product:

- Requested as a sample by an interested party
- Clicked on by an interested party in a marketing email

The following use case illustrates the dependencies:

A customer requests a sample, or adds a product to a wishlist. In marketing, you can trigger lead creation in sales, including the product information (product item). The transfer of product information is based on the product of the trigger event of the marketing campaign for lead creation in sales.

### i Note

During an integration with SAP Cloud for Customer, the system only transfers product items for product origin `SAP_C4C_PRODUCT`.

To drive lead acceptance and conversion probability in sales, during lead transfers, additional and custom attributes from the following objects are transferred together with the lead:

- Predecessor interaction
- Business partner
- Campaign
- Lead score

The system transfers *Score Builder* scores that you can create by yourself, and the delivered *Account Engagement Score* based on a predictive model.

## Predecessor Interaction Information

During each transfer of leads to sales via marketing campaign, the system enriches the transferred content by predecessor interaction information.

This interaction contains a range of attributes listed below. Additionally, it contains the subnodes *Products* and *Item Of Interest* with a table of attributes each.

- Attributes
  - *Origin of Interaction Contact Data*
  - *External ID of Interaction Contact Data*  
Can contain the email address of a contact
  - *Interaction Reason*
  - *Description of Interaction Reason*
  - *Marketing Area ID*
  - *Communication Medium*
  - *Interaction Type*
  - *Description of Interaction Type*
  - *Interaction Content*
  - *Campaign ID*
  - *Interaction Content Subject*
  - *Object Type*
  - *Generic Object ID*
  - *Interaction Source System Type*
  - *Interaction Source System ID*
  - *Generic Object ID*



- *Business Document Status Code*
- *Uniform Resource Identifier*
- *UTC Time Stamp in Long Form (YYYYMMDDhhmmssmmuuun)*
- Subnodes
  - *Products*  
Contains the attributes `ProductId` and `ProductOrigin`; it can contain 0 to N values of this attribute combination.
  - *Item of interest*  
Contains the attributes `ItemOfInterest` and `ItemOfInterestName`; it can contain 0 to N values of this attribute combination.

### i Note

Predecessor interaction information of products is transferred with the lead depending on the assigned sales system type:

- Products of SAP CRM are replicated to SAP CRM.
- Products of SAP Cloud for Customer are replicated to SAP Cloud for Customer.
- For external systems, products to be transferred are not filtered. That is, all products, such as ERP products, are replicated to an external system.

As a prerequisite, there is a predecessor action, or trigger defined to the action *Create Lead*. The following options are possible:

- The predecessor interaction contains additional existing fields, such as *Product*.  
The system replicates additional fields during lead transfer, together with the lead. To add those fields to applicable fields in sales, mapping to sales fields must be defined.
- The predecessor interaction contains custom fields.  
The system replicates the custom fields during lead transfer. To add those fields to applicable fields in sales, mapping to sales fields must be defined.

### i Note

Only the attributes and **not the subnodes** can be extended by custom fields in SAP Marketing Cloud.

## Business Partner Information

For lead transfers where no corresponding business partner exists in sales, the system transfers additional existing, and custom fields of the account, contact, or consumer.

During an integration with SAP Cloud for Customer, also for the creation of call qualification leads in sales, the system transfers additional existing, and custom fields of the account, contact, or consumer.

To add those fields to applicable fields in sales, mapping to sales fields must be defined.

## Campaign Information

The system replicates custom fields on campaigns to the outbound message of the lead, so that this information can be mapped to the business document in sales. To add those fields to applicable fields in sales, mapping to sales fields must be defined.

## Lead Scores

During each transfer of leads to sales via marketing campaign, or via lead transfer, the system enriches the transferred content for leads by lead score information (ID, name, value) for each contact of the assigned target group. That is, scores are added to the lead outbound message.

During an integration with SAP Cloud for Customer, the system also enriches the transferred content for call qualification leads by lead score information (ID, name, value) for each contact of the assigned target group.

To add those fields to applicable fields in sales, mapping to sales fields must be defined.

Lead scores can be calculated regularly. As an administrative user or analyst, you can determine in the [Rule Model](#) how often a score is persisted, how many versions exist, when to delete older versions. The system replicates the most recent persisted version.

### i Note

As a prerequisite for transfer of scores, in the [Score Builder](#), ensure the following:

- Set [Client Application](#) to [Augmented Lead Context](#)
- In the rule set of the score, leave [Applicable For](#) empty.
- The score must be persisted. This means the score must be saved either daily, weekly, or monthly.
- In case there is more than one rule model for a score, the system considers the last changed rule model.

## Related Information

[Score Builder](#)

[Custom Fields in Campaign](#)

[Custom Fields](#)

[Integration with SAP CRM - Outbound Channel \[page 354\]](#)

[Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)

## 4.5.2.1.2 Augmented Activity Context

Enhancement of activity information to be transferred to sales.

Via augmented activity context, sales representatives are provided with context-related activity information that helps to prioritize the sequence of activity processing, and that allows more specified follow-up actions.

For an activity triggered by marketing, the sales representative wants to know the context of this creation that can be a product:

- Requested as a sample by an interested party
- Clicked on by an interested party in a marketing email

To drive activity acceptance in sales, during activity transfers, additional, and custom attributes from the following objects are transferred together with the activity:

- Predecessor interaction
- Campaign

### Predecessor Interaction Information

During each transfer of leads to sales via marketing campaign, the system enriches the transferred content by predecessor interaction information.

This interaction contains a range of attributes listed below. Additionally, it contains the subnodes *Products* and *Item Of Interest* with a table of attributes each.

- Attributes
  - *Interaction Contact Origin*
  - *Interaction Contact ID*  
Can contain the email address of a contact
  - *Interaction Reason*
  - *Interaction Reason Name*
  - *Interaction Contact Marketing Area*
  - *Communication Medium*
  - *Interaction Type*
  - *Interaction Type Name*
  - *Content Data*
  - *Campaign ID*
  - *Interaction Content Subject*
  - *Interaction Content Object Source*
  - *Interaction Content Object ID*
  - *Source System Type*
  - *Source System*
  - *Interaction Source Object Additional ID*
  - *Interaction Source Object Status*
  - *Interaction Source Data URL*
  - *Interaction Source Time Stamp UTC*

- Subnodes
  - *Products*  
Contains the attributes `ProductId` and `ProductOrigin`; it can contain 0 to N values of this attribute combination.
  - *Item of interest*  
Contains the attributes `ItemOfInterest` and `ItemOfInterestName`; it can contain 0 to N values of this attribute combination.

### i Note

Predecessor interaction information of products is transferred with the activity depending on the assigned sales system type:

- Products of SAP CRM are replicated to SAP CRM.
- Products of SAP Cloud for Customer are replicated to SAP Cloud for Customer.
- For external systems, products to be transferred are not filtered. That is, all products, such as ERP products, are replicated to an external system.

As a prerequisite, there is a predecessor action, or trigger defined to the action for activity creation, such as [Create Task](#). The following options are possible:

- The predecessor interaction contains additional existing fields.  
The system replicates additional fields during activity transfer, together with the activity. So, the sales representative is provided with additional information and is enabled to take further actions on customers. For more information, see also the use case description [Creating Tasks in Sales for Missing Marketing Permissions](#).  
To add those fields to applicable fields in sales, mapping to sales fields must be defined in SAP Cloud Integration.
- The predecessor interaction contains custom fields.  
The system replicates the custom fields during activity transfer. To add those fields to applicable fields in sales, mapping to sales fields must be defined in SAP Cloud Integration.

### i Note

Only the attributes and **not the subnodes** can be extended by custom fields in SAP Marketing Cloud.

## Campaign Information

The system replicates custom fields on campaigns to the activity outbound message, so that this information can be mapped to the business document in sales. To add those fields to applicable fields in sales, mapping to sales fields must be defined in SAP Cloud Integration.

## Related Information

[Custom Fields in Campaign Custom Fields](#)

## 4.5.2.2 Integration with SAP Cloud for Customer - Outbound Channel

Data transfer from marketing to sales.

### Data Replication from Marketing to Sales (Outbound)

From SAP Marketing Cloud the creation of the following data in SAP Cloud for Customer can be triggered:

- Business Partners  
If the business partner is not known in SAP Cloud for Customer business partner in status *In Preparation* (account, contact, individual customer) is created.

#### i Note

Business partners in status *In Preparation* are only created in sales during the process of lead creation via marketing campaign.

Before creating business documents, the system creates accounts and contacts in SAP Cloud for Customer with the origin ID from marketing.

- Business Documents
  - Leads  
The system enriches the transferred marketing data during each lead transfer. For more information see [Augmented Lead Context \[page 355\]](#).

#### i Note

For leads and activities transferred to SAP Cloud for Customer, the system also sends out the assigned marketing area. As no mapping is performed in SAP Cloud Integration, the marketing area gets lost when a change in sales is replicated back to marketing.

- Campaigns  
The system transfers basic campaign data to SAP Cloud for Customer for started campaigns.  
With the *Custom Logic* app, you can implement the Business Add-In (BAI) *BAI: Filter for Campaigns to Be Replicated to* SAP Cloud for Customer (BAI\_CUAN\_LM\_CAMPAIGN\_REPLICATION) to adapt the delivered filters of campaign selection for the transfer to SAP Cloud for Customer.  
For the initial load of campaigns to SAP Cloud for Customer, you define an application job in the *Marketing Application Jobs* app. With the application job template *Campaigns: Transfer Campaigns to Sales*, you specify filters to select campaigns for the transfer to SAP Cloud for Customer. After a downtime in the running system, you can also perform this job.  
For more information, see [Campaigns: Transfer Campaigns to Sales](#).  
Filtering of campaigns is done automatically by the system configuration in the Business Add-In (BAI) *BAI: Filter for Campaigns to Be Replicated to* SAP Cloud for Customer (BAI\_CUAN\_LM\_CAMPAIGN\_REPLICATION).

CUAN\_LM\_CAMPAIGN\_REPLICATION). The standard implementation in the BAdI permits the transfer of *Blank Campaigns*, *Email Campaigns*, and *Mobile Campaigns*. The implementation rejects, for example, *Transfer Leads Campaigns*, *Facebook Campaigns* and *Trigger-Based Campaigns*.

- Custom Fields Created in Marketing  
Custom fields created in marketing can also be transferred to SAP Cloud for Customer. For more information on how to extend SAP Marketing Cloud, see the extensibility guide, [Custom Fields for the Integration with SAP Cloud for Customer](#). To make marketing-created custom fields available in SAP Cloud for Customer, you have to ensure that the standard mapping is extended in SAP Cloud Integration .

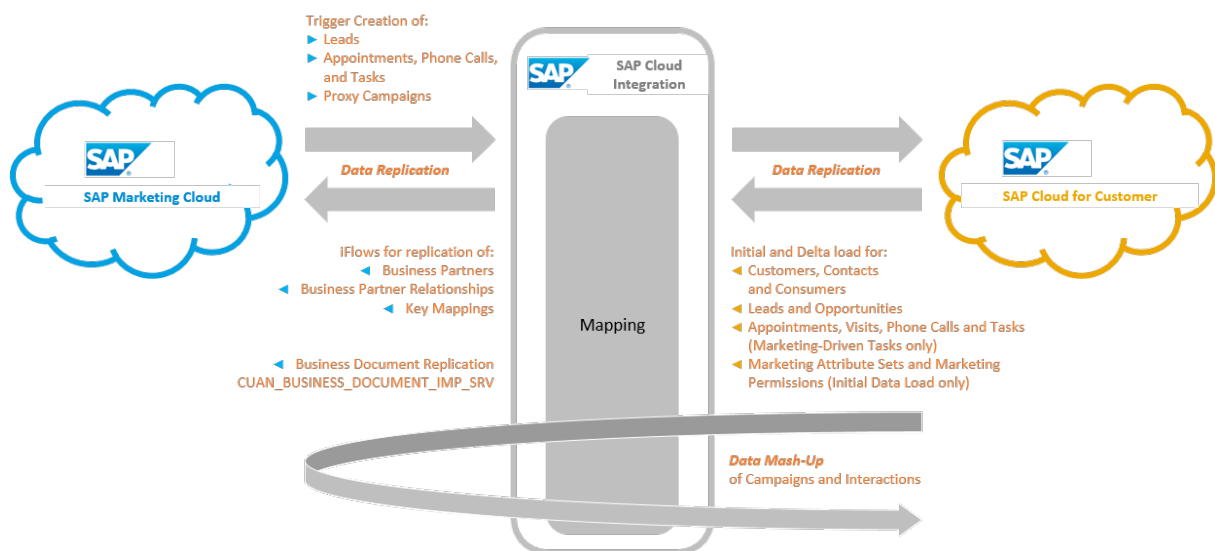
## Integration

You can set up the integration between SAP Marketing Cloud with SAP Cloud for Customer via SAP Cloud Integration.

For the setup you have the following two options:

1. You are guided by the Cloud Integration Automation Service (see the SAP Help Portal at [Introduction to Cloud Integration Automation Service](#)) by making use of the [Maintenance Planner](#).
2. You use the integration setup of SAP Cloud for Customer with SAP Marketing Cloud (see [SAP Cloud for Customer Integration with SAP Marketing](#), or [Setting Up SAP Marketing Cloud Integration with SAP Cloud for Customer \(1J9\)](#)).

The following figure shows an overall integration overview:



Integration of SAP Marketing Cloud with Presales or Sales for Marketing-Driven Leads, Call Center Campaigns, and Activities

For more information about the campaign-based lead creation process, see [Handling Leads](#).

For more information about the process of lead creation via a call center campaign, see [Telephone Campaigns in SAP Cloud for Customer](#).

For more information about the process of campaign-based activity creation, see [Handling Activities](#).

## Business Partner Replication

SAP Marketing Cloud is able to trigger the creation of business partners of type *Account*, *Contact* and *Individual Customer* in SAP Cloud for Customer via lead creation. In case no corresponding business partner exists in SAP Cloud for Customer, a new business partner with the status *In Preparation* is created.

## Related Information

[Lead Campaigns](#)

[Handling Leads](#)

[Marketing-Driven Sales Enablement](#)

[Call Qualification](#)

[Displaying Lead Information for Contacts](#)

[Displaying Lead Information for Accounts](#)

[Lead Dashboard](#)

[Business Documents \[page 661\]](#)

### 4.5.2.2.1 Sales Insights on Marketing Campaigns

Provide campaign data to sales representatives in SAP Cloud for Customer.

## Use Case

As a sales representative, you want to prepare for a dedicated customer visit, and get an overview about those campaigns that affect your area of responsibility. Therefore, you want to see all campaigns that include at least one of the accounts, or contacts you are responsible for to be able to reinforce the message of the campaigns. Furthermore, you are interested in collaboration with marketing to share campaign-related information.

The following pieces of information coming from a campaign may be valuable for you:

- Which campaigns are stopped, or ongoing?  
You can display details of marketing campaigns, such as name, description, type, and status.
- Which accounts, and contacts of my area of responsibility are targeted by those campaigns?  
You can display those contacts, and accounts that are affected by a dedicated marketing campaign.
- Which information is sent by email to a contact?  
You can open emails a contact has received.
- Which campaign-related information is provided by marketing experts?  
SAP Jam allows you to support online communication, such as discussions with marketing, or other sales people, or sharing collaterals for example. During campaign replication, the system also transfers the ID of the attached SAP Jam group to SAP Cloud for Customer. As a sales representative, you can access the group in the sales system.

## Prerequisites

To provide sales representatives with marketing information on campaigns ensure the following:

- Integration with SAP Cloud for Customer, that is the transfer of contacts and accounts, is set up.
- Campaign data is transferred to SAP Cloud for Customer.
- You have configured SAP Jam. For more information see [https://help.sap.com/viewer/user\\_help](https://help.sap.com/viewer/user_help).
- You have integrated the relevant campaign with SAP Jam. In SAP Jam, you have invited additional users to your group to share information with them.

## Navigation to Marketing

In SAP Cloud for Customer, the replicated campaign with the addressed contacts or account is displayed. As a sales representative, you can navigate to:

- Contacts in marketing
- Campaign in marketing

## Collaboration via SAP Jam

Sales representatives and marketing experts collaborate via the same SAP Jam group, that is, they can share campaign collaterals on the SAP Jam group.

In the relevant marketing campaign, under **► Collaboration ► SAP Jam ►**, the marketing expert creates or assigns an SAP Jam group. Per campaign, only one group can be assigned, but a group can be reused in multiple campaigns.

- The feed of the SAP Jam group is displayed on the marketing campaign in SAP Cloud for Customer.
- Sales representatives and marketing experts can create posts from within the feed.

New SAP Jam groups, posts, and feeds created in marketing, sales, or directly in SAP Jam, are always synchronous in those systems.

## Display of Emails in Browser

In SAP Cloud for Customer, the sales representative can display individual emails including personalization attributes that are sent to a contact via a marketing email campaign. That is, as a sales representative you have the same view on the emails as the addressee. So, you are enabled to trigger suitable further sales actions.

For more information about how to insert the *View in Browser* link in email templates, see [Using Links in Emails and Email Templates](#).



## Related Information

[Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)

[Campaigns: Transfer Campaigns to Sales](#)

[Augmented Lead Context \[page 355\]](#)

[Campaign Details](#)

### 4.5.2.2.2 Augmented Lead Context

Enhancement of lead information to be transferred to sales.

Via augmented lead context, sales representatives are provided with context-related lead information that helps to prioritize the sequence of lead processing, and that allows more specified follow-up activities.

Augmented lead context is relevant for the integration with SAP CRM and SAP Cloud for Customer.

For a lead triggered by marketing, the sales representative wants to know the context of this creation, that can be a product:

- Requested as a sample by an interested party
- Clicked on by an interested party in a marketing email

The following use case illustrates the dependencies:

A customer requests a sample, or adds a product to a wishlist. In marketing, you can trigger lead creation in sales, including the product information (product item). The transfer of product information is based on the product of the trigger event of the marketing campaign for lead creation in sales.

#### i Note

During an integration with SAP Cloud for Customer, the system only transfers product items for product origin `SAP_C4C_PRODUCT`.

To drive lead acceptance and conversion probability in sales, during lead transfers, additional and custom attributes from the following objects are transferred together with the lead:

- Predecessor interaction
  - Business partner
  - Campaign
  - Lead score
- The system transfers *Score Builder* scores that you can create by yourself, and the delivered *Account Engagement Score* based on a predictive model.

## Predecessor Interaction Information

During each transfer of leads to sales via marketing campaign, the system enriches the transferred content by predecessor interaction information.

This interaction contains a range of attributes listed below. Additionally, it contains the subnodes *Products* and *Item Of Interest* with a table of attributes each.

- Attributes
  - *Origin of Interaction Contact Data*
  - *External ID of Interaction Contact Data*  
Can contain the email address of a contact
  - *Interaction Reason*
  - *Description of Interaction Reason*
  - *Marketing Area ID*
  - *Communication Medium*
  - *Interaction Type*
  - *Description of Interaction Type*
  - *Interaction Content*
  - *Campaign ID*
  - *Interaction Content Subject*
  - *Object Type*
  - *Generic Object ID*
  - *Interaction Source System Type*
  - *Interaction Source System ID*
  - *Generic Object ID*
  - *Business Document Status Code*
  - *Uniform Resource Identifier*
  - *UTC Time Stamp in Long Form (YYYYMMDDhhmmssmmuuun)*
- Subnodes
  - *Products*  
Contains the attributes `ProductId` and `ProductOrigin`; it can contain 0 to N values of this attribute combination.
  - *Item of interest*  
Contains the attributes `ItemOfInterest` and `ItemOfInterestName`; it can contain 0 to N values of this attribute combination.

### i Note

Predecessor interaction information of products is transferred with the lead depending on the assigned sales system type:

- Products of SAP CRM are replicated to SAP CRM.
- Products of SAP Cloud for Customer are replicated to SAP Cloud for Customer.
- For external systems, products to be transferred are not filtered. That is, all products, such as ERP products, are replicated to an external system.

As a prerequisite, there is a predecessor action, or trigger defined to the action *Create Lead*. The following options are possible:

- The predecessor interaction contains additional existing fields, such as *Product*.  
The system replicates additional fields during lead transfer, together with the lead. To add those fields to applicable fields in sales, mapping to sales fields must be defined.
- The predecessor interaction contains custom fields.

The system replicates the custom fields during lead transfer. To add those fields to applicable fields in sales, mapping to sales fields must be defined.

#### i Note

Only the attributes and **not the subnodes** can be extended by custom fields in SAP Marketing Cloud.

## Business Partner Information

For lead transfers where no corresponding business partner exists in sales, the system transfers additional existing, and custom fields of the account, contact, or consumer.

During an integration with SAP Cloud for Customer, also for the creation of call qualification leads in sales, the system transfers additional existing, and custom fields of the account, contact, or consumer.

To add those fields to applicable fields in sales, mapping to sales fields must be defined.

## Campaign Information

The system replicates custom fields on campaigns to the outbound message of the lead, so that this information can be mapped to the business document in sales. To add those fields to applicable fields in sales, mapping to sales fields must be defined.

## Lead Scores

During each transfer of leads to sales via marketing campaign, or via lead transfer, the system enriches the transferred content for leads by lead score information (ID, name, value) for each contact of the assigned target group. That is, scores are added to the lead outbound message.

During an integration with SAP Cloud for Customer, the system also enriches the transferred content for call qualification leads by lead score information (ID, name, value) for each contact of the assigned target group.

To add those fields to applicable fields in sales, mapping to sales fields must be defined.

Lead scores can be calculated regularly. As an administrative user or analyst, you can determine in the [Rule Model](#) how often a score is persisted, how many versions exist, when to delete older versions. The system replicates the most recent persisted version.

#### i Note

As a prerequisite for transfer of scores, in the [Score Builder](#), ensure the following:

- Set [Client Application](#) to [Augmented Lead Context](#)
- In the rule set of the score, leave [Applicable For](#) empty.
- The score must be persisted. This means the score must be saved either daily, weekly, or monthly.

- In case there is more than one rule model for a score, the system considers the last changed rule model.

## Related Information

[Score Builder](#)

[Custom Fields in Campaign](#)

[Custom Fields](#)

[Integration with SAP CRM - Outbound Channel \[page 354\]](#)

[Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)

### 4.5.2.2.3 Augmented Activity Context

Enhancement of activity information to be transferred to sales.

Via augmented activity context, sales representatives are provided with context-related activity information that helps to prioritize the sequence of activity processing, and that allows more specified follow-up actions.

For an activity triggered by marketing, the sales representative wants to know the context of this creation that can be a product:

- Requested as a sample by an interested party
- Clicked on by an interested party in a marketing email

To drive activity acceptance in sales, during activity transfers, additional, and custom attributes from the following objects are transferred together with the activity:

- Predecessor interaction
- Campaign

## Predecessor Interaction Information

During each transfer of leads to sales via marketing campaign, the system enriches the transferred content by predecessor interaction information.

This interaction contains a range of attributes listed below. Additionally, it contains the subnodes *Products* and *Item Of Interest* with a table of attributes each.

- Attributes
  - *Interaction Contact Origin*
  - *Interaction Contact ID*  
Can contain the email address of a contact
  - *Interaction Reason*
  - *Interaction Reason Name*

- *Interaction Contact Marketing Area*
- *Communication Medium*
- *Interaction Type*
- *Interaction Type Name*
- *Content Data*
- *Campaign ID*
- *Interaction Content Subject*
- *Interaction Content Object Source*
- *Interaction Content Object ID*
- *Source System Type*
- *Source System*
- *Interaction Source Object Additional ID*
- *Interaction Source Object Status*
- *Interaction Source Data URL*
- *Interaction Source Time Stamp UTC*
- Subnodes
  - *Products*  
Contains the attributes `ProductId` and `ProductOrigin`; it can contain 0 to N values of this attribute combination.
  - *Item of interest*  
Contains the attributes `ItemOfInterest` and `ItemOfInterestName`; it can contain 0 to N values of this attribute combination.

### **i** Note

Predecessor interaction information of products is transferred with the activity depending on the assigned sales system type:

- Products of SAP CRM are replicated to SAP CRM.
- Products of SAP Cloud for Customer are replicated to SAP Cloud for Customer.
- For external systems, products to be transferred are not filtered. That is, all products, such as ERP products, are replicated to an external system.

As a prerequisite, there is a predecessor action, or trigger defined to the action for activity creation, such as [Create Task](#). The following options are possible:

- The predecessor interaction contains additional existing fields.  
The system replicates additional fields during activity transfer, together with the activity. So, the sales representative is provided with additional information and is enabled to take further actions on customers. For more information, see also the use case description [Creating Tasks in Sales for Missing Marketing Permissions](#).  
To add those fields to applicable fields in sales, mapping to sales fields must be defined in SAP Cloud Integration.
- The predecessor interaction contains custom fields.  
The system replicates the custom fields during activity transfer. To add those fields to applicable fields in sales, mapping to sales fields must be defined in SAP Cloud Integration.

### i Note

Only the attributes and **not the subnodes** can be extended by custom fields in SAP Marketing Cloud.

## Campaign Information

The system replicates custom fields on campaigns to the activity outbound message, so that this information can be mapped to the business document in sales. To add those fields to applicable fields in sales, mapping to sales fields must be defined in SAP Cloud Integration.

## Related Information

[Custom Fields in Campaign](#)

[Custom Fields](#)

[Integration with SAP CRM - Outbound Channel \[page 354\]](#)

[Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)

### 4.5.2.2.4 Sales Insights on Marketing Permissions and Subscriptions

Provide marketing permissions and subscriptions and enable editing in SAP Cloud for Customer.

## Use Case

As a sales representative, you want to access and change marketing permissions and subscriptions that are created in marketing.

For presales and sales employees, when contacting potential and existing customers via phone or email, it is important to know for which address it is allowed to reach the customers. Especially for call center activities, the knowledge about the permissions is essential.

### i Note

This scenario is only supported for desktops computers and notebooks. Smartphones are not supported and there can be restrictions on tablets.

## Permission Marketing in SAP Marketing Cloud

As a marketing expert, you can maintain permissions and subscriptions on contacts or accounts on the respective user interface, on the [Permission Marketing](#) tab.

In an integrated system landscape, a sales person can edit marketing permissions, in SAP Cloud for Customer. For this use case, SAP Cloud for Customer requests the marketing permissions from SAP Marketing Cloud and provides the corporate account or contact user interface from marketing for editing marketing permissions by a sales person.

## Marketing Permission in SAP Cloud for Customer

In SAP Cloud for Customer, marketing permissions are displayed on SAP Cloud for Customer contacts or accounts under [MARKETING PERMISSIONS](#).

By [Edit Permissions and Subscriptions](#), on a sales contact or account, you can navigate to the corresponding marketing contact or account user interface, to the [Permission Marketing](#) tab. You can edit the marketing permission in the marketing system. The system updates the permission in sales, accordingly.

### i Note

By default, in SAP Cloud for Customer the [MARKETING PERMISSIONS](#) facet does not show the [General](#) section and the URL for editing the permissions. To make it visible, proceed as follows:

1. Choose ► [Adapt](#) > [Edit Master Layout](#) >].
2. Go to [MARKETING PERMISSIONS](#).
3. Add section [General](#).
4. In section [General](#), choose mashup [Edit Permissions and Subscriptions](#).
5. Choose ► [Adapt](#) > [End Layout Changes](#) >].

So, permissions in SAP Cloud for Customer and in SAP Marketing Cloud are synchronous. Sales employees in SAP Cloud for Customer are provided with the same view at the permissions as their marketing counterparts.

### → Tip

It might be required to replicate existing permissions via initial load from SAP Cloud for Customer to SAP Marketing Cloud, and after that, editing permissions shall only be done in marketing.

## Related Information

[Permission Marketing in the Contact Profile](#)

[Integration with SAP Cloud for Customer - Inbound Channel \[page 338\]](#)

[Integration with SAP Cloud for Customer - Outbound Channel \[page 361\]](#)

## 4.5.2.2.5 Sales Insights on Marketing Account and Contact Factsheet

Define a HTML or URL mashup to embed the apps Contact Profile and Spotlighting Accounts in an iFrame in SAP Cloud for Customer.

### Use Case

As a sales representative, you want to get insights about the engagement and activities of your accounts and contacts across multiple communication channels.

### Prerequisites

To provide sales representatives with marketing information on accounts and contacts, ensure the following:

- Integration with SAP Cloud for Customer, that is the transfer of contacts and accounts, is set up.
- Sales representatives must have a user in SAP Marketing Cloud and the authorizations to display the factsheets.

### Navigation to SAP Marketing Cloud information in SAP Cloud for Customer

1. In SAP Cloud for Customer navigate to an account or contact.
2. Navigate to your custom tab, for example the Marketing Factsheet.
3. Log on to the SAP Marketing Cloud to display the account or contact factsheet.

### Set-up Procedure for Mashup

The following steps have to be performed in the SAP Marketing Cloud and the SAP Cloud for Customer in order to enable Sales Insights on Marketing Account and Contact Factsheet.

### Settings in SAP Cloud for Customer

#### Settings in SAP Cloud for Customer for HTML or URL Mashup

1. Navigate to [Administrator](#) > [Mashup Authoring](#).
2. Choose [New](#) > [HTML Mashup](#).  
If you want to create a URL mashup, choose [New](#) > [URL Mashup](#).



3. In the *General Information* section, verify that *With Port Binding* is preselected.
4. Choose ► *Port Binding Type* ► *Additional Account Information* ▾.
5. Enter a name and a description for the mashup, for example **Marketing Factsheet**.
6. Under *Configuration Information*, select *Type URL*.
7. Enter the URL **https://<host>/ui?sap-ushell-config=headerless#MarketingContact-displayFactSheet?OriginID=SAP\_C4C\_BUPA**.  
Host here refers to the SAP Marketing Cloud host.

#### i Note

Do not select *Extract Parameter*.

8. Under *Request Parameters*, choose *Add Row*.
9. As Parameter, enter *InteractionContactID*.
10. Open the value help and select *Parameter Binding AccountInternalID* and set the request parameter to *Mandatory*.
11. Click *Preview* to display the end result of the mashup.  
To test the mashup, you can enter sample values for the parameters, and click *Update Parameter Values* to the right of the *HTML Code Editor*.

#### i Note

If you change the code, you need to click *Preview* again to display the updated result of the mashup.

12. Specify the *Height* with the recommended value *600 px* accordingly.
13. *Save* and *Activate* the mashup.

For more information about creating HTML mashups, see [Create HTML Mashups](#).

### Adding the Mashup to the *Account* and *Contact UI*

1. Go to your user profile, and from the dropdown list select *Start Adaptation*. The system opens in the *Adaptation Mode*.
2. Open an account. In the side pane, select ► *Add* ► *Tab* ▾.
3. Go to the tab bar of the main screen and select the tab you just created.
4. Select the blue icon on the new tab. The system highlights the area with a red border to indicate that you can make changes.
5. In the side pane, click the reverse arrow icon twice to navigate from the *Form Pane* to the *UI Component* view.
6. Select *Add Mashup* to open a new window.
7. Depending on your selection in step 2, set the filter to *HTML Mashups* or *URL Mashup*.  
For the *HTML Mashups*, carry out the following additional steps:
  1. Select the row that contains the required mashup to display the *Properties* header.
  2. Tick the checkbox for the same mashup to make the *Properties* editable.
  3. Tick the checkbox *Full Width* in *Properties* and set the *Height(%)* to 100.  
The newly added mashup will occupy the full height of the screen.
8. Choose *Apply*.
9. To save your settings, go to your profile and select *End Adaptation*.

*Add mashup to contact UI:* If you want to add the marketing factsheet to the contact UI, start the *Adaptation Mode*, open a *Contact* and add a new tab. Then repeat the steps 3 to 12.

For more information about adding mashups on screen, see [Add Mashups on Screens](#)

## Settings in SAP Marketing Cloud

1. Assign the business catalog `SAP_CEC_BC_MKT_CFS2_PC` to your user.  
With this catalog, you can access to the apps *Contact Profile* and *Spotlighting Accounts*. As a reference, have a look at the user `SALES_REP_MKT_INFO`, that has the *Business Role* `SAP_BR_SALES_REP_MKT_INFO` and the *Business Catalog* `SAP_CEC_BC_MKT_CFS2_PC` assigned.
2. As an administrator, check if the SAP Cloud for Customer UI host is added in the app *Maintain Clickjacking Protection Whitelist* in SAP Marketing Cloud.  
The host is required to be able to click inside the Marketing UI in the iFrame in SAP Cloud for Customer.
3. If the host is not listed, add it.

## Technical Background

Using this URL will automatically navigate to/embed either the contact or account UIs in their legacy or Fiori incarnation depending on the business role catalog that is assigned to the user .

### 4.5.2.3 Integration with External Sales Systems - Outbound Channel

Data transfer from marketing to external sales systems.

By integrating SAP Marketing Cloud with external sales systems, the bridge between marketing and sales is built so that processes can be harmonized across marketing and sales channels.

#### ⚠ Caution

For the integration of SAP Marketing Cloud with external sales systems, we do not deliver standard content.

#### i Note

- You can set up the integration with internal and an external sales system simultaneously, but only with one system of a target system type at a time. That is, you can integrate one system for SAP CRM, one for SAP Cloud for Customer, and one external sales system.
- In case you have activated more than one communication arrangement, that is, you have configured more than one target system, the Business Add-In (BAI) *Lead Management: Determine Target System Type* is performed. With the *Custom Logic* app, you can implement the BAI. You define in which target system the corresponding leads are created.
- You can also use the integration with Salesforce offered by Advantco International LLC. For more information, see [SAP Marketing Cloud Integration with Salesforce](#) .

## Outbound Processes from Marketing to an External Sales System

Depending on the data model of an external sales system, the following data can be replicated to an external sales system

- Business Documents  
Leads are transferred to the external sales system.  
The system enriches the transferred marketing data during each lead transfer. For more information see [Augmented Lead Context \[page 355\]](#).

### i Note

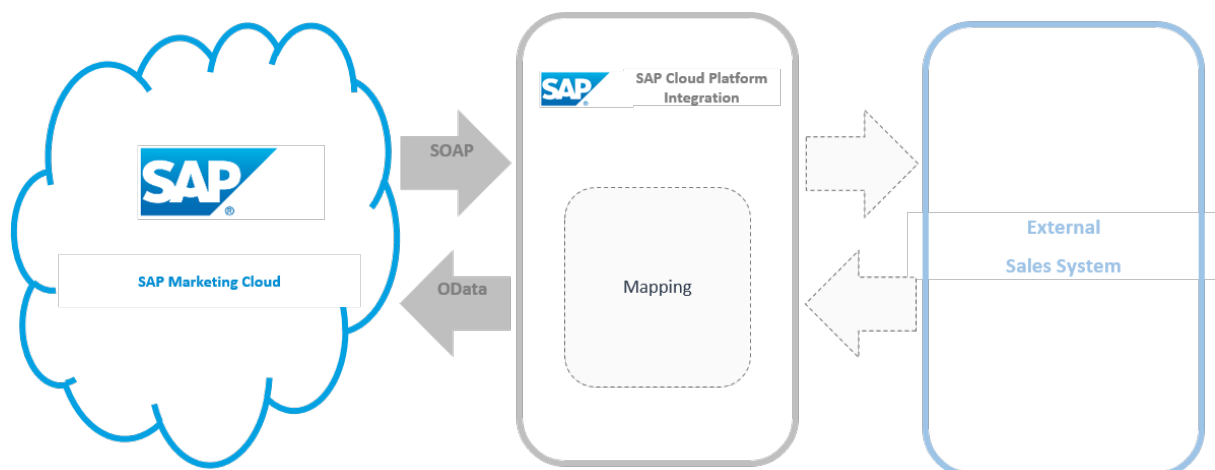
For leads transferred to an external sales system, the system also sends out the assigned marketing area. As no mapping is performed in SAP Cloud Integration, the marketing area gets lost when a change in sales is replicated back to marketing.

- Custom fields created in marketing  
Custom fields created in marketing can be transferred to the external sales system within the lead creation process.

## Integration

You can set up the integration between SAP Marketing Cloud with an external sales system via SAP Cloud Integration.

The following figure shows a possible overall integration overview based on a customer-owned integration setup:



Integration of SAP Marketing Cloud with External Sales Systems

### 4.5.2.3.1 Setting Up the Connection Between Marketing and an External Sales System

Connect SAP Marketing Cloud with an external sales system.

Before doing the configuration in SAP Marketing Cloud, you need the administrator business user, which contains the business catalog SAP\_CORE\_BC\_COM (Communication Management), for example the business role SAP\_BR\_ADMINISTRATOR (Administrator).

As an administrator, you maintain the setup via the following apps under *Communication Management*:

- Maintain Communication User  
For more information, see [Creating a Communication User for Inbound Communication \[page 376\]](#).
- Communication Systems  
For more information, see [Setting Up a Communication System for the Integration of an External Sales System \[page 377\]](#).
- Communication Arrangements  
For more information, see [Setting Up a Communication Arrangement for the Integration of an External Sales System \[page 378\]](#).

#### 4.5.2.3.1.1 Creating a Communication User for Inbound Communication

Define a user for inbound communication.

The communication user defined in the SAP Marketing Cloud system is used for inbound communication and for the processing of messages in the system. Technically the user is needed to call OData services in SAP Marketing Cloud from SAP Cloud Integration.

1. Log on to your SAP Marketing Cloud system.
2. On the launchpad, select the *Maintain Communication Users* tile
3. Choose *New* to create a new user (for example, MKT\_COM\_USR) or select an existing user.
4. Assign a password for the user if you would like to use the basic authentication, or assign SAP Cloud Integration, integration service public client certificate to the user for certificate-based authentication:
  1. Basic authentication: Enter password in the *Password* field

#### i Note

Basic authentication with a user and password is possible, but not recommended in a productive environment for security reasons.

2. Certificate based authentication: Choose *Upload Certificate* (exported from SAP Cloud Integration, integration service keystore. The link to download certificate is provided by SAP Operations in the initial tenant provisioning mail).
5. Choose *Save*.
6. Note down the user data for further steps.

#### i Note

In the SAP Cloud Integration system, you need this technical user to call the OData services.

## 4.5.2.3.1.2 Setting Up a Communication System for the Integration of an External Sales System

Connect the communication user with the communication arrangement.

The *Communication System* is used to define the host name of the SAP Cloud Integration tenant and to assign users for the inbound (from SAP Cloud Integration to SAP Marketing Cloud) and outbound (from SAP Marketing Cloud to SAP Cloud Integration) communication.

To create the communication system for SAP Cloud Integration, proceed as follows:

1. Log on to your SAP Marketing Cloud system.
2. On the launchpad, open the *Communication Systems* app.
3. To create a new system, choose *New*.
4. In *New Communication System*, enter a system ID and its name, and choose *Create*.
5. In the *Technical Data* section, enter the details of your SAP Cloud Integration tenant in the *Host Name* field (see SAP Cloud Integration tenant provisioning email).  
As a host is irrelevant to the inbound communication, enter dummy to assign a dummy host.
6. Assign the communication user created earlier to this communication system, as follows:
  - In the *User for Inbound Communication* section, choose + (Add).
  - In *New Inbound Communication User*, select the authentication method as *User Name and Password* and enter the user created earlier.
7. To create a new outbound user, in the *User for Outbound Communication* section, choose + (Add).  
In *New Outbound User*, select the authentication method:
  - Basic Authentication

Authentication Method	Username and Password
User Name	P-user that has access to the SAP Cloud Integration service tenant.
Password	Password for the user

- Certificate-Based Authentication

Authentication Method	SSL Client Certificate
Certificate Type	Exported from SAP Cloud Integration, integration service keystore. The link to download certificate would be provided by SAP Operations in initial tenant provisioning mail.

8. Choose *Create*.
9. Save and activate the communication system.

### 4.5.2.3.1.3 Setting Up a Communication Arrangement for the Integration of an External Sales System

Set up the communication with SAP Cloud Integration.

*Communication Arrangements* need to be activated in SAP Marketing Cloud for communication with OData APIs. The communication arrangement in the SAP Marketing Cloud defines all relevant information for the communication with SAP Cloud Integration. It contains the communication system, and inbound and outbound authentication.

The communication arrangement *SAP\_COM\_0017: Marketing - Presales/Sales Integration* needs to be activated.

To set up the communication arrangement, proceed as follows:

1. Log on to SAP Marketing Cloud with a user that has administrator authorizations.
2. From the *SAP Fiori launchpad*, choose the *Communication Arrangements* app.
3. To create a new communication arrangement for the communication scenario *SAP\_COM\_0017 (Marketing - Presales/Sales Integration)*, choose *New*.  
Select the scenario and enter an arrangement name. Choose *Create*.
4. In the *Common Data*, choose the communication system that you created previously.
5. In *Additional Properties*, select the *Target System Type*. Choose *SALES\_EXT*.

#### i Note

In case a communication arrangement with *Target System Type SALES\_EXT* already exists, you cannot activate another one.

6. Select values for *First Origin of Contact ID*, *Second Origin of Contact ID*, *Third Origin of Contact ID*.  
New additional properties are defined to be able to configure *Origins of Contact IDs* for target system type *SAP\_EXTERNAL*. The combination of *ID\_ORIGIN* and *ID* must uniquely identify a contact in SAP Marketing Cloud. Depending on the usage of number ranges assigned to business partners in the external sales system, more than one *ID\_ORIGIN* can be necessary.  
Valid values can be defined in the *Origin of Contact* app under **Implementation Cockpit > Manage Your Solution > Configure Your Solution > Marketing/Contacts and Profiles > OriginContactID > Configure > .**  
The rules above are effective independent of the existence of communication arrangements with different target system types.

#### i Note

It is only possible to activate a communication arrangement with *Target System Type = SALES\_EXT*, if the following *Additional Properties* or *Outbound Services* are maintained as follows:

- The additional property *Campaign Transfer to Sales* is inactive.
- The additional property *First Origin of Contact ID* is available.
- The outbound service *Export of Activities* is inactive.
- The outbound service *Export of Campaigns* is inactive.

7. In *Inbound Communication*, enter the previously created user. The authentication method is filled automatically.
8. In *Outbound Communication*, the SAP Cloud Integration service user that was assigned to *Communication System* is automatically added or the default client certificate with certificate download link is displayed.

- in *Outbound Services*, ensure that only *Export of Leads* is activated.

Field Name	Entry
Service Status	Checked (Active)
Application Protocol	SOAP
Port	443
Path	/cxf/<ExternalSalesSystemService>
Service URL	Will be automatically populated.

9. Choose *Save*.

## 4.5.3 Financial Data

[Integration with SAP ERP for Spend Planning \[page 379\]](#)

With the following instructions you are able to integrate spend data from your system with SAP ERP. A campaign is represented by a project and a spend item as a WBS element. You use these WBS elements as account reference for further processing within SAP ERP.

### 4.5.3.1 Integration with SAP ERP for Spend Planning

With the following instructions you are able to integrate spend data from your system with SAP ERP. A campaign is represented by a project and a spend item as a WBS element. You use these WBS elements as account reference for further processing within SAP ERP.

#### Prerequisites

- You have SAP ERP 6.0 EHP 4 or higher.
- You have set up the SAP Marketing Cloud - *SAP ERP Actual and Committed Spend* integration package. For more information, see <https://api.sap.com/package/SAPS4HANAMarketingCloudSAPERPActualandCommittedSpendIntegration>.

## Setting Up the Communication with SAP ERP

To set up the communication between SAP Marketing Cloud and SAP ERP, perform the following steps:

1. Create the system for outbound communication using the *Communication Systems* configuration application. Enter the following data:

<i>System Name</i>	Name of the SAP ERP system, for example, <b>ABC</b> .
<i>Host Name</i>	Host address of the SAP ERP system, for example, <b>ldiabc.corp.com</b> .
<i>Authentication Method</i>	Select an authentication method.

2. Create a communication arrangement with a certificate-based authentication for the outbound scenario.
3. Create a communication arrangement for the *Marketing - Planning Spend Integration* (SAP\_COM\_0018) scenario using the *Communication Arrangements* configuration application. Enter the following data:

▶ <i>Common Data</i> ▶ <i>Arrangement Name</i> ▶	SAP_COM_0018
▶ <i>Common Data</i> ▶ <i>Communication System</i> ▶	The SAP ERP system created in the <i>Communication Systems</i> configuration application in the previous step.
▶ <i>Outbound Communication</i> ▶ <i>User Name/Certificate</i> ▶	Select certificate-based authentication for the outbound scenario. Download the certificate. It will be used later while setting up SAP ERP and configuring the security.
▶ <i>Outbound Services</i> ▶ <i>Port</i> ▶	Port for the communication.



You must set up the following SOAP services in the order defined below and specify the path for each of them:

1. Create service, for example, `/sap/bc/srt/xip/sap/ecc_projecterpcrtrc1/<client of SAP ERP system>/<name of service/binding that will be used when configuring Web Services>/<name of service/binding that will be used when configuring Web Services>`
2. Update service, for example, `/sap/bc/srt/xip/sap/ecc_projectupdrc/<client of SAP ERP system>/<name of service/binding that will be used when configuring Web Services>/<name of service/binding that will be used when configuring Web Services>`
3. Get service, for example, `/sap/bc/srt/xip/sap/ecc_projecterpidqr1/<client of SAP ERP system>/<name of service/binding that will be used when configuring Web Services>/<name of service/binding that will be used when configuring Web Services>`

The path for each service is defined during the SAP ERP setup.

For example, `<client of SAP ERP system>` could be 100, `<name of service/binding that will be used when configuring Web Services>` could be `cuan_msm`. The name you define in this step must be the same as the name defined during the SAP ERP setup.

---

For more information, search for *Maintain Communication Arrangements* and *Maintain Communication Systems* on SAP Help Portal at <http://help.sap.com> under the *SAP S/4HANA Cloud* product.

## Setting Up SAP ERP

1. Configure the Web service runtime as target system connection. For more information, search for *Configuring the Web Service Runtime* on SAP Help Portal at <http://help.sap.com> under the *SAP NetWeaver* product.
2. Configure your security settings for the service provider and service consumer using the SOA Manager. For more information, search for *Runtime Configuration with the SOA Manager* on SAP Help Portal at <http://help.sap.com> under the *SAP NetWeaver* product.
3. Configure service definitions in the Web service configuration transaction (SOAMANAGER) for the following:
  - `ProjectERPCreateRequestConfirmation_In_V1 (ECC_PROJECTERPCRTRC1)`

- ProjectERPUpdateRequestConfirmation\_In (ECC\_PROJECTUPDRC)
- ProjectERPByIDQueryResponse\_In\_V1 (ECC\_PROJECTERPIDQR1)

Configure the services as follows:

1. In the *Service and Binding Name* step, enter the same name in the *Service Name* and *New Binding Name* fields. This must be the same name as the one defined in step 3 of the communication setup with SAP ERP under **► Outbound Services ► Path ▾**.
2. In the *Provider Security* step, under *Transport Level Security*, select the *SSL (https)* radio button, and under *Transport Channel Authentication*, select *X.509 SSL Client Certificate*.
3. Go through the other steps without specifying any values and complete the configuration.

You will find the *WSDL URL for Binding* in the *WSDL Generation for Binding* of each service.

For more information, search for *Configuring a Service Provider* on SAP Help Portal at <http://help.sap.com> under the *SAP NetWeaver* product.

As in SAP Marketing Cloud, prefix `1_CUAN_MSM_<CampaignID>` is given for creation of projects and `1/<CampaignID>` for WBS elements in SAP ERP, no predefined coding mask is required for project coding key 1.

Project profile CUAN01 has to be configured in SAP ERP.

## Checking for Errors

You can use the *Application Log* application to check if there are errors with the integration. You can use the CUAN category and the *Marketing Spend Integration in External System* (CUAN\_MSM\_SPEND\_DISTR) subcategory.

You can also use the *Message Dashboard* application to monitor if there are errors with the integration and to reprocess the integration. For more information, see [Data Exchange Messages for Spend Planning Integration](#).

For a complete description of the configuration settings required for the integration scenario, see the setup guide at <https://api.sap.com/shell/discover/contentpackage/SAPS4HANAMarketingCloudSAPERPAActualandCommittedSpendIntegration?section=DOCUMENTS>.

### 4.5.3.1.1 Importing Actual and Committed Spend from SAP ERP

You can import actual and committed spend associated to campaigns from SAP ERP and make it available in the *Spend* area of the *Campaigns* application.

## Prerequisites

- You have enabled the integration of spend data with SAP ERP. For more information, see [Integration with SAP ERP for Spend Planning \[page 379\]](#).

- You have set up the SAP Marketing Cloud - *SAP ERP Actual and Committed Spend* integration package. For more information, see <https://api.sap.com/package/SAPS4HANAMarketingCloudSAPERPActualandCommittedSpendIntegration>.
- You have configured the communication scenario *Marketing - Business Data Integration* in SAP Marketing Cloud.
- You have installed the latest version of the SAP HYBRIS C4C ERP INTEGR product in SAP ERP that contains the COD\_ERP\_INT 6.00 component. For information about this product and the CODERINT 600 add-on, see the corresponding documentation on SAP Support Portal at <http://support.sap.com>  
 ▶ [Download Software](#) ▶ [By Alphabetical Index \(A-Z\)](#) ▶ [C](#) ▶ [SAP HYBRIS C4C ERP INTEGR](#) and [https://rapid.sap.com/bp/#/browse/categories/lines\\_of\\_business/areas/marketing/packageversions/BP\\_CLD\\_MKT](https://rapid.sap.com/bp/#/browse/categories/lines_of_business/areas/marketing/packageversions/BP_CLD_MKT)  
 ▶ [Solution Scope](#) ▶ [Integration](#) ▶ [SAP ERP Actual and Committed Spend Integration \(19Y\)](#).
- You have the authorization to run the CUAN\_ERP\_MSM\_EXTRACT\_ACTUAL report in SAP ERP.

## Context

You can import actual and committed spend data from SAP ERP using this integration.

## Procedure

To import the actual and committed spend, run the CUAN\_ERP\_MSM\_EXTRACT\_ACTUAL report.

## Results

The actual and committed spend associated to campaigns are shown in the *Spend* area of the *Campaigns* application.

For a complete description of the configuration settings required for the integration scenario, see the setup guide at <https://api.sap.com/shell/discover/contentpackage/SAPS4HANAMarketingCloudSAPERPActualandCommittedSpendIntegration?section=DOCUMENTS>.

### 4.5.4 Survey Data

The documentation explains the following topics:

- How to integrate SAP Qualtrics Surveys with SAP Marketing Cloud using integration flows and you intend to send surveys using the SAP Marketing Cloud system. For more information, see [Integration with SAP Qualtrics Surveys \[page 384\]](#).

- How to integrate SAP Qualtrics Surveys with SAP Marketing Cloud using integration flows and you intend to send surveys outside of the SAP Marketing Cloud system. For more information, see [Integration with SAP Qualtrics Surveys Using Actions Functionality \[page 384\]](#).

[Integration with SAP Qualtrics Surveys \[page 384\]](#)

[Integration with SAP Qualtrics Surveys Using Actions Functionality \[page 384\]](#)

### 4.5.4.1 Integration with SAP Qualtrics Surveys

Integration of SAP Qualtrics Surveys with SAP Marketing Cloud using SAP Cloud Integration: [SAP Qualtrics Surveys Integration with SAP Marketing Cloud](#).

By supporting the integration of SAP Qualtrics Surveys with SAP Marketing Cloud, customers can benefit from the features of Qualtrics. This integration fetches and stores data easily from SAP Qualtrics Surveys into SAP Marketing Cloud system.

To achieve this integration, the following iFlows are provided:

- Create Survey Data in SAP Marketing Cloud.
- Create Survey Subscription in Qualtrics.
- Mapping Qualtrics Surveys Data for SAP Marketing Cloud.

Watch this 4-minute video, which provides you a step-by-step integration overview, which helps you to integrate SAP Qualtrics Surveys with SAP Marketing Cloud

For more information, see [Integrating SAP Qualtrics Surveys with SAP Marketing Cloud](#).

### 4.5.4.2 Integration with SAP Qualtrics Surveys Using Actions Functionality

Integration of SAP Qualtrics Surveys with SAP Marketing Cloud using SAP Cloud Integration Suite:

By supporting the integration of SAP Qualtrics Surveys with SAP Marketing Cloud, customers can benefit from the features of Qualtrics. With this integration, you can:

- distribute surveys from an external system like Qualtrics or a website
- fetch survey responses of your existing customers
- fetch survey responses and create a contact profile for nonexistent contacts in your system

To achieve this integration, the following integration flows are provided:

- Create Survey Data and Response in SAP Marketing Cloud.
- Load Buffered Survey Data to SAP Marketing Cloud.

Watch this 3-minute video, which provides you a functional and integration overview of the integration using the Actions functionality.

For more information, see [Integrating SAP Qualtrics Surveys with SAP Marketing Cloud Using Actions Functionality](#).

## 4.5.5 Personalized Commerce

With the following personalized Commerce options, you can tailor your Commerce implementation to suite your customers.

### [Integration with SAP Commerce \[page 385\]](#)

Support omnichannel activities by integrating SAP Marketing Cloud with SAP Commerce Cloud.

### [Consuming Recommendation Models Using an OData Service \[page 385\]](#)

The `API_MKT_RECOMMENDATION_SRV` and `PROD_RECO_RUNTIME_SRV` public OData services enable customer channels to receive recommendations generated by *Recommendation*.

### [Offer Discovery \[page 386\]](#)

Discover suitable offer content for consumers.

### [Exporting Offline Sales Data \[page 386\]](#)

Export offline sales data from SAP Marketing Cloud and make it available in other applications.

### 4.5.5.1 Integration with SAP Commerce

Support omnichannel activities by integrating SAP Marketing Cloud with SAP Commerce Cloud.

For more information, see [Integration with SAP Commerce Cloud \[page 62\]](#).

### 4.5.5.2 Consuming Recommendation Models Using an OData Service

The `API_MKT_RECOMMENDATION_SRV` and `PROD_RECO_RUNTIME_SRV` public OData services enable customer channels to receive recommendations generated by *Recommendation*.

You can enable customer channels to receive recommendations generated by *Recommendation* using the following services:

#### `API_MKT_RECOMMENDATION_SRV`

The `API_MKT_RECOMMENDATION_SRV` public OData service for *Recommendations* allows a client system to obtain product or offer recommendations from the SAP Marketing Cloud using the SAP Business Technology Platform. The service is easy to consume and enables you to benefit from the following:

- Built-in redundancy in the event of SAP Marketing Cloud unavailability.

If the SAP Marketing Cloud is unresponsive when a request for a recommendation is submitted, a comprehensive fallback process is initiated. The process begins by trying to retrieve a personalized recommendation from the cache using the user's ID (if available) and the leading items associated with the request. If that fails, a second restricted attempt is made using similar users (for example, a target group) and the leading items associated with the request. If that fails, a third generic request is submitted using the leading items exclusively. This process of submitting personalized, restricted, and generic requests continues using subsets of the leading items from the most to least recent until a recommendation is returned.

- Enriched recommendation results.  
The API retrieves product master data from the SAP Marketing Cloud. The data enriches the recommendation results obtained, for example, by providing product images and descriptions.

For more information, see [Recommendations \(SAP Business Technology Platform\) \[page 923\]](#).

## PROD\_RECO\_RUNTIME\_SRV

The PROD\_RECO\_RUNTIME\_SRV public OData service for *Recommendations* allows a client system to obtain product recommendations from the SAP Marketing Cloud.

For more information, see [Recommendations \[page 939\]](#).

### 4.5.5.3 Offer Discovery

Discover suitable offer content for consumers.

The personalized offer recommendations are based on eligibility and validity. The personalization is determined by geo location, offer attributes, and scores.

For more information, see:

- SAP SCN [Offer Recommendation in SAP Commerce](#).
- [Discover Offers \[page 1008\]](#).

### 4.5.5.4 Exporting Offline Sales Data

Export offline sales data from SAP Marketing Cloud and make it available in other applications.

This integration provides a foundation for repurposing offline sales data from SAP Marketing Cloud. It acts as middleware, using the API\_MKT\_INTERACTION\_SRV and API\_MKT\_CONTACT\_SRV services to retrieve information from SAP Marketing Cloud. The integration flow exports the data in a comma-separated value (CSV) file, and then sends it to an SFTP server. From there, you can perform custom development in your target system to upload and use the data.

For a complete description of the configuration settings required for integration, see the [Introduction](#).

For more information, see the offline sales [Integration package](#).

# 5 Integration APIs

Are you trying to pull or push information for individual marketing entities such as campaigns, target groups, or contacts? There is a wide range of public APIs available to enable you to integrate with SAP Marketing Cloud. Refer to the following table to quickly find the information that will help you get started, no matter what your level of knowledge.

## ⚠ Caution

The API services available in SAP Marketing Cloud must not be used for mass read (GET) operations. In other words, you cannot use them for extracting all available data, for example, to extract millions of contacts or interactions from your marketing system.

Questions This Guide Answers	Read Me
Which API should I use if I want to integrate a third-party data source that provides, for example, <b>agreement</b> , <b>campaign</b> , or <b>contact</b> information.	<a href="#">Quick Guide - Which API for Which Entity [page 389]</a>
What do I need to know before using the marketing APIs?	<a href="#">Getting Started [page 387]</a>
What are SAP APIs?	<a href="https://developers.sap.com/topics/api.html">https://developers.sap.com/topics/api.html</a> 📄
How do I navigate the SAP API Business Hub?	<a href="https://api.sap.com/getting-started">https://api.sap.com/getting-started</a> 📄
I'm new to OData. How does it work?	<a href="http://www.odata.org">http://www.odata.org</a> 📄

## 5.1 Getting Started

This section contains information to help you get started quickly, including **communication prerequisites** for integrating with APIs, deep-dive **videos** that will help you find your way around the SAP API Business Hub and understand the data load concepts, as well as some useful **best practices** and recommendations for efficient integration and data load.

### [Videos - Best Practices for Data Load \[page 388\]](#)

These short videos provide valuable insight into the data load concepts, and include useful tips for before and after you load data into your marketing system. The videos are available in English only.

### [Quick Guide - Which API for Which Entity \[page 389\]](#)

SAP Marketing Cloud offers a wide range of services. But which one is right for your purposes? Take a few minutes to browse this table according to the entity type you want to import.

### [Consuming the Integration APIs \[page 395\]](#)

#### [Extending the Integration APIs \[page 396\]](#)

The following table indicates which APIs can be extended and provides links to further information.

#### [Optimize Performance During OData Service Calls \[page 397\]](#)

This section describes how to call an OData service in a way that ensures a high degree of system security and performance. The description uses API\_MKT\_INTERACTION\_SRV as an example, but the method applies to all OData services in SAP Marketing Cloud.

#### [Best Practices and Recommended Package Sizes \[page 400\]](#)

This section contains best practices for optimizing data load of **master data** entities, recommended package sizes, and some troubleshooting tips.

#### [Import Monitor \[page 404\]](#)

Monitor and explore data imports that are triggered by OData or upload services from external systems.

#### [Data Load Monitor \[page 407\]](#)

Monitor all import messages and keep track of their status.

#### [HTTP Response Status Codes \[page 408\]](#)

Every HTTP request that is received by a server is responded to with a 3-digit HTTP status code. They are grouped into five classes.

## 5.1.1 Videos - Best Practices for Data Load

These short videos provide valuable insight into the data load concepts, and include useful tips for before and after you load data into your marketing system. The videos are available in English only.

### Data Load Videos (English Only)

#### Planning and Configuring the Contact Data Load

This 5-minute video explains what origin IDs are, the significance of the configuration settings you can make, and how these affect the contact match and merge process.

#### Configuring Origins

This 2-minute video explains important points to consider when you configure your data sources and why it is important to have one origin per data source.

#### How Imported Data Is Processed

This 6-minute video explains the match and merge process that contact data undergoes whenever new data is uploaded.



## Data Lifecycle Management - Decluttering Your System

This 5-minute video explains how and why you should declutter your system regularly of marketing data that no longer serves any useful purpose.

### i Note

The application job used as an example in this video has changed and is no longer valid, but the deletion concept recommended in the video still applies.

## Analytical List Pages in SAP Marketing Cloud

This 3-minute video walks you through the highly-configurable functions of analytical list pages (ALPs). Based on the example of the Browse Contact Origin Data app, explore the insights you can gain from ALPs.

## Data Load Monitor

This 3-minute video explains the importance of monitoring imports, and of regularly analyzing and fixing common error causes. It shows how the *Data Load Monitor* can support you in safeguarding the quality of data imports.

## Related Information

[Video Library](#)

## 5.1.2 Quick Guide - Which API for Which Entity

SAP Marketing Cloud offers a wide range of services. But which one is right for your purposes? Take a few minutes to browse this table according to the entity type you want to import.

### Quick Guide - Which API Should I Use?

You can search, sort, and filter the table to view the data as best suits your requirements. For example, you can quickly search by **Entity Type** to find the recommended service or other import options, which you can use in exceptional cases, for example for a one-time import for test purposes.

Integration Services in SAP Marketing Cloud

Area	Entity Type	Recommended Service	Other Import Options
Contacts and Profiles	<p>Contacts</p> <ul style="list-style-type: none"> <li>Marketing Permissions</li> <li>Marketing Subscriptions</li> </ul>	<p>Public OData API (<code>API_MKT_CONTACT_SRV</code> Version 4) for reading and writing master data about Contacts. Contacts are natural persons who interact with your company.</p> <p><a href="#">Contacts [page 412]</a></p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <p>You must use version 4 of this API service if you implement <i>Contact-to-Account Relationships</i>.</p> </div>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Contacts</li> </ul>
Contacts and Profiles	<p>Interaction Contacts</p> <ul style="list-style-type: none"> <li>Marketing Permissions</li> <li>Marketing Subscriptions</li> </ul>	<p>Public OData API (<code>API_MKT_INTERACTION_CONTACT_SRV</code> Version 3) for Interaction Contacts. Interaction Contact is a generic term to group all natural persons (contacts, consumers, or suspects), companies and “unknowns”, who interact with your company.</p> <p><a href="#">Interaction Contacts [page 469]</a></p>	
Contacts and Profiles	<p>Corporate Accounts</p> <ul style="list-style-type: none"> <li>Marketing Permissions</li> <li>Marketing Subscriptions</li> </ul>	<p>Public OData API (<code>API_MKT_CORPORATE_ACCOUNT_SRV</code> Version 3) for reading and writing master data about Corporate Accounts only. Corporate accounts are companies or organizations that interact with your company.</p> <p><a href="#">Corporate Accounts [page 512]</a></p>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Corporate Accounts</li> </ul>
Contacts and Profiles	<p>Products</p>	<p>Public OData API (<code>API_MKT_PRODUCT_SRV</code>) for Products.</p> <p><a href="#">Products [page 582]</a></p>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Products</li> </ul>
Contacts and Profiles	<p>Product Hierarchies and Categories</p>	<p>Public OData API (<code>API_MKT_PROD_CAT_HIERARCHY_SRV</code>) for Product Hierarchies and Categories.</p> <p><a href="#">Product Hierarchies and Categories [page 604]</a></p>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Product Categories</li> </ul>

Area	Entity Type	Recommended Service	Other Import Options
Contacts and Profiles	Interactions	Public OData API ( <code>API_MKT_INTERACTION_SRV</code> ) for Interactions.  <a href="#">Interactions [page 615]</a>	<ul style="list-style-type: none"> <li><a href="#">Business Documents [page 661]</a></li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <p>For <b>business documents</b> (leads, opportunities, sales orders, and so on), we recommend that you use the API Service <code>CUAN_BUSINESS_DOCUMENT_IMP_SRV</code>, since it provides an <b>upsert</b> function and updates an already existing entry depending on time-stamp information.</p> </div> <ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Interactions</li> </ul>
Contacts and Profiles	Business Documents from SAP Cloud for Customer	Public OData API ( <code>CUAN_BUSINESS_DOCUMENT_IMP_SRV</code> ) for importing business documents, such as leads and opportunities, from external SAP or non-SAP systems to SAP Marketing Cloud. Use this version of the service when you want to import business documents related to Offers and Coupons. Lower versions are not suitable for this purpose.  <a href="#">Business Documents [page 661]</a>	
Contacts and Profiles	Contacts, Corporate Accounts or Relationships from SAP ERP, SAP CRM, or S/4HANA On Premise	<code>CUAN_BUSINESS_PARTNER_IMPORT_SRV</code> for importing business partner data from external source systems, like, for example, SAP ERP, SAP CRM, SAP S/4HANA On Premise.  <a href="#">Import Business Partners [page 574]</a>	
Contacts and Profiles	Account Team Members (for Companies)	Public OData API ( <code>API_MKT_CORPORATE_ACCOUNT_SRV</code> Version 3) for reading and writing master data about Corporate Accounts only. Corporate accounts are companies or organizations that interact with your company.  <a href="#">Corporate Accounts [page 512]</a>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Account Team Members</li> </ul>

Area	Entity Type	Recommended Service	Other Import Options
Contacts and Profiles	Account Team Members (for Contacts)	Public OData API ( <code>API_MKT_CONTACT_SRV</code> Version 4) for reading and writing master data about Contacts. Contacts are natural persons who interact with your company.  <a href="#">Contacts [page 412]</a>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Account Team Members</li> </ul>
Contacts and Profiles	Interests	Public OData API ( <code>API_MKT_INTEREST_SRV</code> )  <a href="#">Interest Items [page 648]</a>	<a href="#">Manage Interests</a> app - <a href="#">Manage Interests</a>
Contacts and Profiles	Agreements	Public OData API ( <code>API_MKT_AGREEMENT_SRV</code> ) for agreements. An agreement can be any kind of customer contract, for example, a sales contract or a contract that comprises specific services.  <a href="#">Agreements [page 681]</a>	
Contacts and Profiles	Marketing Locations	Public OData API ( <code>API_MKT_LOCATION</code> ) for Marketing Locations. A marketing location is any physical or virtual location where a marketing activity can be conducted.  <a href="#">Marketing Locations [page 710]</a>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Import Marketing Locations</li> </ul>
Contacts and Profiles	Scores	Public OData API ( <code>API_MKT_SCORE_SRV</code> ) for Scores  <a href="#">Scores [page 700]</a>	
Contacts and Profiles	Marketing Permissions and Marketing Subscriptions	Public OData API ( <code>API_MKT_CONTACT_SRV</code> Version 4) for reading and writing master data about Contacts. Contacts are natural persons who interact with your company.  <a href="#">Contacts [page 412]</a>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Permissions and Subscriptions</li> </ul>
Contacts and Profiles	Marketing Permissions and Marketing Subscriptions	Public OData API ( <code>API_MKT_INTERACTION_CONTACT_SRV</code> Version 3) for Interaction Contacts. Interaction Contact is a generic term to group all natural persons (contacts, consumers, or suspects), companies and "unknowns", who interact with your company.  <a href="#">Interaction Contacts [page 469]</a>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Permissions and Subscriptions</li> </ul>
Contacts and Profiles	Marketing Permissions and Marketing Subscriptions	Public OData API ( <code>API_MKT_CORPORATE_ACCOUNT_SRV</code> Version 3) for reading and writing master data about Corporate Accounts only. Corporate accounts are companies or organizations that interact with your company.  <a href="#">Corporate Accounts [page 512]</a>	<ul style="list-style-type: none"> <li><a href="#">Data File Load</a> - Permissions and Subscriptions</li> </ul>

Area	Entity Type	Recommended Service	Other Import Options
Commerce Marketing	Offers	Use the public OData API <code>CUAN_OFFER_IMPORT_SRV</code> to upload (import) offers from external sources.  <a href="#">Import Offers [page 973]</a>	
Commerce Marketing	Read Offers	Public OData API ( <code>API_MKT_OFFER_SRV</code> ) for Offers  <a href="#">Read Offers [page 1002]</a>	
Commerce Marketing	Discover Offers	Use the API OData service <code>CUAN_OFFER_DISCOVERY_SRV</code> for SAP Marketing Cloud Offers to find suitable offers for a consumer.  <a href="#">Discover Offers [page 1008]</a>	
Commerce Marketing	Coupons	Public OData API ( <code>API_MKT_COUPON_SRV</code> ) for Coupons.  <a href="#">Coupons [page 1026]</a>	
Commerce Marketing	Recommendations Interaction Data	OData service ( <code>PROD_RECO_RUNTIME_SRV</code> ) for posting interactions to an SAP HANA database.  <a href="#">Recommendations Interaction Data [page 971]</a>	
Extensibility	Custom Business Objects	Import data into a Custom Business Object by using an OData service  <a href="#">Import of Data into Custom Business Object [page 1057]</a>	<ul style="list-style-type: none"> <li>• <a href="#">Data File Load</a> - Custom Business Objects</li> </ul>
Marketing Planning and Performance	Custom Dimensions	<a href="#">Custom Dimension Values</a> - Import Custom Dimensions	
Marketing Planning and Performance	Actual and Committed Spend	<a href="#">Actual and Committed Spend Data [page 1041]</a> You can upload actual and committed spend data from an external ERP system into SAP Marketing Cloud using the <code>CUAN_ACTUAL_IMPORT_SRV</code> OData service.	<ul style="list-style-type: none"> <li>• <a href="#">Financial Data [page 379]</a> - SAP ERP Integration is the preferred method wherever possible.</li> <li>• <a href="#">Data File Load</a> - Actual and Committed Spend</li> </ul>

### i Note

If you use the data file load for occasional imports in parallel to a scheduled job using the OData service, you must consolidate the data to ensure data reconciliation and consistency.

Area	Entity Type	Recommended Service	Other Import Options
Campaign Management	Campaigns	Public OData API ( <a href="#">API_MKT_CAMPAGN_SRV</a> ) for Campaigns  <a href="#">Campaigns [page 767]</a>	
Campaign Management	Campaign Execution Plans	<a href="#">CUAN_MPO_IMPORT_SRV</a>  <a href="#">Campaign Execution Plans [page 763]</a>	
Campaign Management	Campaign Message Content and Personalized Email Content	Public OData API ( <a href="#">API_MKT_CAMPAGN_MESSAGE_SRV</a> ) for exporting and importing message content in multiple languages.  <a href="#">Campaign Message Content and Personalized Email Content [page 793]</a>	
Campaign Management	Campaign Performance	Public OData API ( <a href="#">API_MKT_CMPGN_SUCCESS_IMPORT</a> ) for importing aggregated success data for Campaigns.  <a href="#">Campaign Success Data [page 812]</a>	<ul style="list-style-type: none"> <li>• <a href="#">Data File Load</a> - Campaign Performance</li> </ul>
Campaign Management	Survey	OData API ( <a href="#">CUAN_SURVEY_IMPORT_SRV</a> ) that supports operations on survey metadata and survey responses.  <a href="#">Survey [page 886]</a>	<ul style="list-style-type: none"> <li>• <a href="#">Data File Load</a> - Survey Metadata</li> <li>• <a href="#">Data File Load</a> - Survey Response</li> </ul>
Campaign Management	Marketing Events	Public OData API ( <a href="#">API_MKT_EVENT_SRV</a> ) for importing events data from third-party event provider platforms.	
Campaign Management	External Landing Pages (Landing Pages)	Public OData API ( <a href="#">API_MKT_LANDING_PAGE</a> ) for writing external landing pages to the SAP Marketing Cloud system.  <a href="#">External Landing Pages [page 743]</a>	
Campaign Management	External Landing Pages (Forms)	Public OData API ( <a href="#">API_MKT_LANDING_PAGE</a> ) for writing external forms to the SAP Marketing Cloud system.  <a href="#">External Landing Pages [page 743]</a>	
Campaign Management	External Landing Page Value Help	Public OData API ( <a href="#">API_MKT_LANDING_PAGE_VALUEHELP</a> ) for retrieving attribute values used in landing pages.  <a href="#">External Landing Page Value Help [page 749]</a>	

Area	Entity Type	Recommended Service	Other Import Options
Marketing Analytics	Audiences	Cluster reporting results or assign budgets to audiences.  <a href="#">Audiences</a>	
Marketing Analytics	Brands	Add and edit brands, and import brand data from a comma-separated value (CSV) file. You can also delete values of brands that are not used in any business objects, such as budget plans.  <a href="#">Brands</a>	
Segmentation	Target Groups	Public OData API ( <code>API_MKT_TARGET_GROUP_SRV</code> ) for Target Groups  <a href="#">Target Groups [page 755]</a>	<a href="#">Export Target Groups and Target Group Member Data [page 761]</a>
Campaign Management	Export Definitions	Public OData API ( <code>API_MKT_EXPORT_DEFINITION</code> ) for Export Definitions.  An export definition is a template for structuring the export of target group member data, included in a target group or a campaign, to CSV files.  <a href="#">Read Content of Export Files in Campaigns [page 901]</a>	
Marketing Planning and Performance	Marketing - Programs	Public OData API ( <code>API_MKT_PROGRAM_SRV</code> ) for marketing programs.  <a href="#">Marketing Programs [page 1045]</a>	

## 5.1.3 Consuming the Integration APIs

### Overview

The SAP Marketing Cloud public APIs conform to the OpenAPI specifications. All the OData APIs are listed on the SAP API Business Hub at <https://api.sap.com/>. You can test these APIs on the SAP API Business Hub.

### Setting Up Communication with SAP Marketing Cloud

To set up a communication system and communication arrangement, you require the business catalog role Communication Management (`SAP_CORE_BC_COM`) assigned to your user. For more information, see:

- [Communication Management](#)
- [How to Create Communication Users](#)
- [How to Create Communication Systems](#)
- [How to Create a Communication Arrangement](#)

## Extending SAP Marketing Cloud

SAP S/4 HANA Cloud extension procedures are applicable for extension of SAP Marketing Cloud as well. For more information, see [Extending SAP S/4HANA Cloud](#).

## Configuring the Extension Application Connectivity to SAP Marketing Cloud

The following SAP S/4 HANA Cloud extension procedures are applicable for SAP Marketing Cloud as well.

- [Using Basic Authentication](#)
- [Using Client Certificate Authentication](#)
- [Using SAML Bearer Assertion Authentication](#)

### 5.1.4 Extending the Integration APIs

The following table indicates which APIs can be extended and provides links to further information.

For detailed information about:

- Adding custom fields to the SAP Marketing Cloud APIs and the relevant business contexts to use, see [Custom Fields](#).
- Adding custom fields in an integration package, for example adding a custom field to a replicated business object such as a business partner, see [SAP Marketing Cloud Scenario-Based Extensibility](#).

Extensible API Services in SAP Marketing Cloud

Area	API Service	Business Context
Contact Profiling	Contacts	<ul style="list-style-type: none"> <li>• Marketing: Contact</li> <li>• Marketing: Contact and Corporate Account</li> <li>• Marketing: Marketing Attributes for Contacts</li> </ul>
Contact Profiling	Interaction Contacts	<ul style="list-style-type: none"> <li>• Marketing: Contact</li> <li>• Marketing: Contact and Corporate Account</li> </ul>
Contact Profiling	Corporate Accounts	<ul style="list-style-type: none"> <li>• Marketing: Corporate Account</li> <li>• Marketing: Contact and Corporate Account</li> </ul>



Area	API Service	Business Context
Contact Profiling	Products	<ul style="list-style-type: none"> <li>Marketing: Product</li> </ul>
Contact Profiling	Product Categories	<ul style="list-style-type: none"> <li>Marketing: Product Category</li> </ul>
Contact Profiling	Interactions	<ul style="list-style-type: none"> <li>Marketing: Interaction</li> </ul>
Contact Profiling	Agreements	<ul style="list-style-type: none"> <li>Marketing: Agreement</li> <li>Marketing: Agreement Terms (time dep.)</li> </ul>
Contact Profiling	Marketing Locations	<ul style="list-style-type: none"> <li>Marketing: Marketing Location</li> </ul>
Campaign Management	Campaigns	<ul style="list-style-type: none"> <li>Marketing: Campaign</li> </ul>
Campaign Management	Campaign Success Data	<ul style="list-style-type: none"> <li>Marketing: Campaign Performance Actual Measure</li> <li>Marketing: Campaign Performance Dimension</li> <li>Marketing: Campaign Performance Target Measure</li> </ul>
Campaign Management	Marketing Events	<ul style="list-style-type: none"> <li>Marketing: Marketing Events</li> </ul>
Commerce Marketing	Import Offers	<ul style="list-style-type: none"> <li>Marketing: Offer Header</li> <li>Marketing: Offer Content</li> </ul>
Commerce Marketing	Read Offers	<ul style="list-style-type: none"> <li>Marketing: Offer Header</li> <li>Marketing: Offer Content</li> </ul>
Commerce Marketing	Discover Offers	<ul style="list-style-type: none"> <li>Marketing: Offer Header</li> <li>Marketing: Offer Content</li> </ul>
Commerce Marketing	Coupons	<ul style="list-style-type: none"> <li>Marketing: Coupon Code</li> </ul>
Marketing Planning and Performance	Marketing Programs	<ul style="list-style-type: none"> <li>Marketing: Program</li> </ul>

## 5.1.5 Optimize Performance During OData Service Calls

This section describes how to call an OData service in a way that ensures a high degree of system security and performance. The description uses API\_MKT\_INTERACTION\_SRV as an example, but the method applies to all OData services in SAP Marketing Cloud.

### Importing Data into SAP Marketing Cloud

1. Request an x-CSRF token and a session cookie by calling the metadata document, for example, `https://<mkt.com>/sap/opu/odata/sap/api_mkt_interaction_srv/$metadata`.
2. In the `get request` header, you must add the parameter name `x-csrf-token` and the value `Fetch`, as shown in the code snippet. This get request returns the x-CSRF token and session cookie in the response.
3. Create the payload with the data you want to post.

4. Post the data via the corresponding endpoint and send the x-CSRF token and the session cookie that you received in step 1.  
In the post request:
  - In the parameter `x-csrf-token` enter the value from the token you received in step 1.
  - In the parameter `Content-type`, enter the value `application/json`.
  - Add the session cookie you received from the `get metadata` request, for example `https://<mkt.com>/sap.opu.odata/sap/api_mkt_interactions_srv/InteractionsDeepInsert`.
5. You should terminate the session cookie by calling the logoff service. For example, `https://<mkt.com>/sap/public/bc/icf/logoff`. By doing this, you ensure that the session cookie and the x-CSRF token are no longer valid. In this `get request`, you have to add:
  - The parameter 'x-csrf-token' and the value you received in step 1.
  - The session cookie you received from the `get metadata` request.

### Important Points to Note

- The session cookie will automatically terminate after 30 minutes idle time.
- You should reuse the session cookie and the x-CSRF token for as long as you can. In other words, you should try to avoid exceeding 30 minutes idle time.
- By reusing the session cookie, you avoid having additional calls to generate a new cookie every time. This leads to improved performance because you have to execute the `get` call only once.
- By terminating the session cookie, you secure the system because the cookie and CSRF token can no longer be used.

## Example

### Code Syntax

```

*****
* Establish the connection
* get request to fetch the CSRF Token and session cookie
*****
lv_header_field-name = 'x-csrf-token'.
lv_header_field-value = 'Fetch'.
INSERT lv_header_field INTO TABLE lt_header_fields.
TRY.
  IF ( lt_cookies IS INITIAL ).
    cl_cuan_http_helper=>s_get_instance( )->http_call(
      EXPORTING
        iv_destination    = '<SM59_ENTRY>'
        iv_url             = '/sap/opu/odata/sap/api_mkt_interaction_srv/
$metadata'
        iv_method          = 'GET'
        it_header_fields  = lt_header_fields
      IMPORTING
        ev_status_code    = lv_status_code
        ev_x_csrf_token   = lv_x_csrf_token
        et_cookies        = lt_cookies
    )
  CLEAR lt_header_fields.
  lv_header_field-name = 'x-csrf-token'.
  lv_header_field-value = lv_x_csrf_token.
  INSERT lv_header_field INTO TABLE lt_header_fields.
  lv_header_field-name = 'Content-Type'.
  lv_header_field-value = 'application/json'.
  INSERT lv_header_field INTO TABLE lt_header_fields.

```

```

ENDIF
CATCH cx_cuan_cpred_error INTO exc_cpred_error.
* Error handling
ENDTRY.

```

## Code Syntax

```

*****
* Create payload and
* send it via post request to SAP Marketing Cloud
*****
WHILE lv_true = abap_true.
* creating the payload with your data and store it in variable lv_body
*...
*...
TRY.
    cl_cuan_http_helper=>s_get_instance( )->http_call(
        EXPORTING
            iv_destination    = '<SM59_ENTRY>'
            iv_url            = '/sap/opu/odata/sap/api_mkt_interaction_srv/
InteractionsDeepInsert'
            iv_method         = 'POST'
            iv_body_send      = lv_body
            it_header_fields  = lt_header_fields
            it_cookies        = lt_cookies
        IMPORTING
            ev_body_receive   = lv_body_receive
            ev_status_code    = lv_status_code
    ).
CATCH cx_cuan_cpred_error INTO exc_cpred_error.
* Error handling
ENDTRY.
IF ( lv_status_code = '403' ).
* generate new CSRF-Token and session cookie and try again go to step 1
*...
*...
ENDIF.
ENDWHILE.

```

## Code Syntax

```

*****
* Terminate the session cookie
*****
TRY.
    cl_cuan_http_helper=>s_get_instance( )->http_call(
        EXPORTING
            iv_destination    = 'E3W'
            iv_url            = '/sap/public/bc/icf/logoff'
            iv_method         = 'GET'
            iv_body_send_as_string = lv_basis
            it_header_fields  = lt_header_fields
            it_cookies        = lt_cookies
        IMPORTING
            ev_body_receive   = lv_body_receive
            ev_status_code    = lv_status_code
    ).
CATCH cx_cuan_cpred_error INTO exc_cpred_error.
* Error handling
ENDTRY.

```

## 5.1.6 Best Practices and Recommended Package Sizes

This section contains best practices for optimizing data load of **master data** entities, recommended package sizes, and some troubleshooting tips.

### General Recommendations

Object Type	Recommendations
All entities	<ul style="list-style-type: none"><li>• When you upload entities synchronously, parallel upload of entities is not allowed since this can lead to data inconsistencies.</li><li>• As a general principle, upload master data object types before transactional data object types. For example, when you upload data for multiple object types initially, you should do so in the following upload sequence:<ol style="list-style-type: none"><li>1. Product categories</li><li>2. Products</li><li>3. Interaction Contacts</li><li>4. Interactions</li><li>5. Marketing Permissions</li><li>6. Marketing Subscriptions</li></ol></li></ul>
Contacts	<p><b>Synchronous and Asynchronous Processing</b></p> <ul style="list-style-type: none"><li>• When you import contacts using an OData service, the data is processed <b>asynchronously</b> by default. This means that when you trigger a contact import, in most cases an OK response, such as a receipt notification, is returned almost immediately. An exception to this are data uploads that might contain severe errors, such as parse or format errors. These do not return an OK response but an error message. The data you upload lands in a staging area, where it is then further processed. To view the processing status of data uploads and to check for errors or success messages, you must open the <i>Import Monitor</i> app. In the event of errors, you can restart or discard the import from the import log. For more information, see <a href="#">Import Monitor [page 404]</a>.</li><li>• You can force imports to be processed synchronously by selecting the flag <code>SAP-CUAN-ForceSynchronousProcessing</code>. In this case, if an error is detected, an error message is returned immediately.</li><li>• In a <b>synchronous</b> import of contacts, only contacts with errors are aborted. All contacts within the same package without errors are imported successfully. You should refer to the <i>Import Monitor</i> to check for errors, correct the errors and post these contacts again.</li><li>• In an <b>asynchronous</b> import of contacts, in the event of errors, you can correct the errors and restart the import from the Import Monitor directly.</li><li>• Do not mix different types of services for operations involving the same data source for the same business entity. For example, when importing contacts from a web shop, do not use one service for a PUT operation and a different service to PATCH contacts. You should, however, migrate from CUAN_IMPORT to the API* services.</li></ul>

Object Type	Recommendations
Interactions	<p data-bbox="400 360 847 383"><b>Synchronous and Asynchronous Processing</b></p> <ul data-bbox="408 405 1394 819" style="list-style-type: none"> <li data-bbox="408 405 1394 506">• Import packages that contain up to <b>999</b> interactions are processed <b>synchronously</b>. Packages with <b>1000 interactions or more</b> are typically posted to the staging area and are processed <b>asynchronously</b>.</li> <li data-bbox="408 517 1394 618">• In a <b>synchronous</b> import of interactions, success notifications are returned immediately. In the event of errors, only a warning notification is returned, indicating that there were problems. To see exact details of any errors that occurred, you must open the <i>Import Monitor</i> app.</li> <li data-bbox="408 629 1394 819">• In an <b>asynchronous</b> import of interactions, processing is done via the staging area. An OK response is returned in most cases, indicating that the import has landed in the staging area. To see exact details of the import status, that is, whether there were errors or whether the import was successful, you must open the <i>Import Monitor</i> app. An exception to this are data uploads that might contain severe errors, such as parse or format errors. These immediately return an error message from Gateway.</li> </ul> <div data-bbox="448 842 1394 1095" style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p data-bbox="472 853 552 875"><b>i Note</b></p> <p data-bbox="472 909 1370 1066">The system automatically tries to restart the import of any interactions that have been blocked and written to the staging area. This takes place every minute for up to 9999 attempts, taking roughly one week. You can manually process interactions with errors by correcting the errors, for example, by changing the configuration. For more information, see <a href="#">Import Monitor [page 404]</a>.</p> </div> <ul data-bbox="408 1111 1394 1352" style="list-style-type: none"> <li data-bbox="408 1111 1394 1200">• The entity <code>InteractionDeepInsert</code> is available to enable better performance for imports. We recommend that you use this entity when importing interaction data. For more information, see <a href="#">Interactions [page 615]</a>.</li> <li data-bbox="408 1211 1394 1352">• For performance reasons, we recommend that you do not import in BATCH. However, if you do use BATCH imports, you should be aware that a 20x HTTP response code is always returned, whether errors are detected or not. You should always refer to the <i>Import Monitor</i> app to check the status messages of your imports. Additionally, you should also check the response message in case a technical error occurred. These are not listed in the <i>Import Monitor</i>.</li> </ul>

## Object Type

## Recommendations

### Marketing Permissions and Marketing Subscriptions

- When you import marketing permissions or marketing subscriptions using the OData services `API_MKT_CONTACT` (version 0004), `API_MKT_CORPORATE_ACCOUNT` (version 0003) or `API_MKT_INTERACTION_CONTACT` (version 0003), data is processed asynchronously by default. During asynchronous processing, you will receive an OK response, such as a receipt notification. Only in the event that the payload is not supplied correctly, an error message and an http code > 400 is returned. Such errors are:
  - Payload cannot be parsed because of an error
  - Format errors
  - Key fields are not supplied properlyYou can monitor the asynchronous processing by launching the *Import Monitor* App. In the case of errors, you can restart or discard the messages from the import log.
- When you use synchronous processing and the data could not be processed properly, error messages are returned immediately to the caller. An error message is returned and the http code will be larger than 400. You will not find these entries in the *Import Monitor* app.

#### **i** Note

Be aware that in case of errors, the whole payload is rejected and nothing is posted. If you want to use synchronous processing, you have to set the field `SAP-CUAN-ForceSynchronousProcessing` to true. You also have to take into account that there might be timeout when sending large payloads with synchronous calls, resulting in data loss. This is avoided during asynchronous processing.

- **→ Recommendation**

We recommend asynchronous processing for mass data.

- As marketing permissions and marketing subscriptions are separate entities as of version 0003 of the API services, you have to populate both nodes if you want to send marketing permissions and marketing subscriptions.

- **→ Recommendation**

For the initial load, we recommend that you choose a two-step approach: Loading contacts in a first step and loading marketing permissions and marketing subscriptions in a second step after the initial load of contacts has been finished. If you want to send contacts and marketing permissions and marketing subscriptions at the same time, ensure that you send all entities within one request, which will ensure the processing in a proper sequence.

- To avoid locking problems, all marketing permissions and marketing subscriptions belonging to a contact should be part of one package.

## Recommended Package Sizes

Object Type	Package Sizes
All entities	<ul style="list-style-type: none"> <li>The maximum supported OData request size is <b>100</b> megabytes.</li> <li>Depending on your package size, we recommend using at most 5 parallel sessions to load the data. If you load asynchronously, try increasing the number of concurrent sessions incrementally. Start with just two, then three, and so on.</li> </ul>
Contacts	<p><b>OData</b></p> <ul style="list-style-type: none"> <li>The recommended package size is 1000 entities per request.</li> </ul> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p><b>i Note</b></p> <p>If you send contacts together with their marketing permissions and subscriptions (which is what we recommend), you have to make sure that the sum of all entities does not exceed 1000. For example, 100 contacts with their 900 permissions/subscriptions amounts to 1000 entities. Therefore, you should calculate the average number of permissions and subscriptions per contact in advance.</p> </div> <ul style="list-style-type: none"> <li>More than 10000 will result in error.</li> <li>We recommend to use the default <b>asynchronous</b> processing mode for initial loads.</li> </ul>
Interactions	<ul style="list-style-type: none"> <li><b>OData API_MKT_INTERACTION</b> The general recommendation is 1000-5000 interactions including sub-nodes. The fewer sub-nodes and the less data contained in the sub-nodes, the more interactions can be imported. See also the information above <a href="#">about asynchronous processing [page 401]</a>. We recommend that you do not import single data records. Where possible, interaction imports should always be bundled.</li> </ul> <p><b>Maximum Package Sizes</b></p> <ul style="list-style-type: none"> <li>Without subnodes or long Edm Strings: Max. 50000</li> </ul> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p><b>i Note</b></p> <p>You can check the content of Edm Strings in the metadata, which is linked in the section <a href="#">Technical Field Documentation</a> for every API service.</p> </div> <ul style="list-style-type: none"> <li>Without products, 2 interests (on average): 20000</li> <li>With 5 products, 5 Interests (no long Edm Strings): 5000</li> <li>With 5 products, 5 Interests (with long Edm Strings): 1000 - 2000</li> </ul> <ul style="list-style-type: none"> <li><b>CSV</b> <ul style="list-style-type: none"> <li>CSV imports are only recommended for test purposes and for small volumes of data.</li> <li>Max. 10000 (memory restricted)</li> </ul> </li> </ul>
Products	Maximum package size: 10000
Product Categories	Maximum package size: 10000
Agreements	Maximum package size: 10000

Object Type	Package Sizes
Interests	500 (2 Languages)
Marketing Permissions and Marketing Subscriptions	<div style="background-color: #f0f0f0; padding: 10px;"> <p><b>i Note</b></p> <p>Performance basically depends on which other processes are running in parallel in marketing.</p> <ul style="list-style-type: none"> <li>• Do not use a packages size that exceed 1000 data sets.</li> <li>• Use parallel processing. Sending multiple requests in parallel will increase the performance. We recommend sending 10 to 20 parallel requests.</li> <li>• More than 10000 entries per package will result in an error.</li> </ul> </div>

## Related Information

[Import Monitor \[page 404\]](#)

[HTTP Response Status Codes \[page 408\]](#)

## 5.1.7 Import Monitor

Monitor and explore data imports that are triggered by OData or upload services from external systems.

With this app, administrative users can monitor the data import from outside SAP Marketing Cloud. You can import data via upload services, or OData services.

Within the integration of marketing with sales, data are transferred from SAP Cloud for Customer, and SAP CRM to SAP Marketing Cloud via SAP Cloud Integration. The data import into SAP Marketing Cloud is done by OData services.

As a marketing administrative user, you can handle import notifications that are caused by the import of business partner or business document data from SAP Cloud for Customer, and SAP CRM to SAP Marketing Cloud. SAP Marketing Cloud generates a list of notifications with the related status that denotes the progress of a data import:

### Notifications

You can select an import notification from a list that provides all notifications including important metadata, such as service name, date, size, status.



The system lists all import notifications grouped by their status:

- **In Process:** The notification is not yet finalized. Data import is not complete so far.
- **Error:** The import notification has caused an error, for example because of mapping errors. Data import has not taken place.



- **Success:** The notification is processed. Data import is finalized.



### i Note

You can find detailed information for import notifications under [Messages](#) , the data records are available under [Data](#) .

## Details of an Import Notification

For each import notification, the system provides the following detailed grouped information for the *File Import*:

- **Import Notification**
  - **Size:** The number of data records processed in an import notification
  - **Interface:** The interface that has triggered the data import
  - **Service Name:** The service, for example an OData service that has triggered the import of data, such as business partners, business documents, or marketing attributes
  - **Source System:** Source system of the data records to be transferred
  - **Created By:** The technical user used for import processing.
  - **Force Synchronous Processing:** Indicates whether data is stored for the import notification or not, and whether a restart of the notification is possible or not:
    - **Yes:** No data is available to inspect and a restart is not possible
    - **No:** Data is available to inspect and a restart is possible.
  - Reference message that allows you to identify the message in all involved systems, such as middleware or sending system, with different monitoring tools:
    - SAP Cloud Integration (Middleware)  
In SAP Cloud Integration, choose [Operations](#), and under [Monitor Message Processing](#) click on [All Integration Flows](#). Enter the Reference Message ID under [Application Message ID](#). The system displays the message. If [Message Tracing](#) is activated, you can also display the message payload.
    - Source system SAP Cloud for Customer  
In SAP Cloud for Customer, navigate to the [Web Service Message Monitoring](#) under [Administration](#). Open the Advanced Search, enter the Reference Message ID in search field [Message ID](#), and choose [Go](#). The system displays the original message.
    - Source System SAP CRM  
To find the original message in source system SAP CRM you have to carry out multiple steps:
      - In SAP Cloud Integration, choose [Operations](#), and under [Monitor Message Processing](#) click on [All Integration Flows](#). Enter the Reference Message ID under [Application Message ID](#). The system displays the message.
      - Click on [Message Processing Log](#), and search for string `com.sap.sod.utils.idoc.soap.idocassign`. The system displays one or more entries, as follows:  
`com.sap.sod.utils.idoc.soap.idocassign0= [00000000000006115,00000000000081447]`  
The second number displayed in the square brackets is the IDoc number.
      - In the SAP CRM system, call transaction [IDoc List](#) (WE05), and search for the corresponding IDoc with the IDoc number copied from SAP Cloud Integration.
- **Timestamps**  
Date and time, when the notification was generated or changed.

- *External*: Timestamp of the arrived import notification as set from sender system; Local time of the sender.
- *External (UTC)*: Timestamp of the arrived import notification as set from sender system; Universal date and time.
- *Created*: Timestamp of the arrived import notification as set from receiving system; Local time of the user.
- *Changed*: Timestamp of the latest change of the import notification as set from the receiving system; Local time of the user.
- *Status*
  - Status of the notification
  - Number of messages
- Messages  about the performed notification
- Data  for all import notifications with status *Error*.

### **i** Note

Errors, that is, data records that cannot be saved in SAP Marketing Cloud, can be caused by the following reasons:




- Data is locked by another user
- Customizing data is missing, such as origins of contact IDs, because the relevant BC set was not unpacked
- Mapping errors occurred in SAP Cloud Integration




By restarting or discarding, the notification errors can be resolved. For mapping errors, you can only discard the notification, because the error must be resolved in the relevant SAP Cloud Integration system.

## Features

The following features are available:

Features Overview of Import Monitor

Feature	Description
Multiple Selection 	You can select several import notifications, and restart or discard them in one step.
Search 	You can search for import notifications by entering the user name, or the notification ID.
Sort 	You can sort the import notifications by: <ul style="list-style-type: none"> <li>● Service name</li> <li>● Source system</li> <li>● Data and time, ascending, or descending</li> </ul>

Feature	Description
Restart	<p>You can select a notification, and <a href="#">Restart</a> it to recheck to processed data.</p> <p><b>Example:</b></p> <p>For notifications with status <a href="#">Error</a> you can trigger actions to resolve the error. After the correction, you can restart the notification, that is, the processing of the imported data, to check whether your correction was successful. The notification status then turns to <a href="#">Successful</a>.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <p>If no restart is possible for the current import notification you can correct error only externally, that is, in the source system, or in the middleware system.</p> </div>
Discard	You can <a href="#">Discard</a> a notification that is no longer necessary or valid.
Share 	You can share the notification via email, or on SAP Jam.
Data 	<p>You find data records of the current data import, with status <a href="#">Error</a>.</p> <p>By <a href="#">Show Full Record</a>, all fields of a data record are revealed. You can switch between the different data records using <a href="#">Previous Record</a>, and <a href="#">Next Record</a>.</p>
Messages 	<p>You find all messages generated by the current import notification.</p> <p>The messages contain descriptions of the activities performed for a notification, classified by their severity: <a href="#">Information</a>, <a href="#">Warning</a>, or <a href="#">Error</a>. You find detailed information for a message, if available, under <a href="#">More Information</a>.</p>

## 5.1.8 Data Load Monitor

Monitor all import messages and keep track of their status.

The [Data Load Monitor](#) app enables you to optimize your data imports by enabling you to:

- View import messages across the landscape and decide what action to take.
- Correct errors in the Marketing or in the source systems in a timely fashion and restart imports.
- Analyze imports and messages by multiple dimensions, and quickly resolve issues.
- Identify system issues that may have been previously hidden.

- View various status messages such as success messages, errors, and warnings to identify issues with data mapping or system configuration.

## Video (English Only)

This short video shows how the Data Load Monitor can support you in safeguarding the quality of data imports.

## Analyzing Import Errors

The Data Load Monitor collects all import messages (errors, warnings, and success messages) and displays the number of times individual messages occur across all imports. We recommend that you use the app to preform error analysis on a regular basis, eliminate frequent causes of error, and so optimize the quality of your data imports. Error analysis can be done in 4 simple steps:

1. Gain an overview of the errors and warning occurring in your imports by setting the filter options according to your requirements.

### i Note

There are a large number of filter options available in the Compact Filters, for example, import header, import service used, or source system ID. Note, however, that you cannot search on individual payload content. Use the *Import Monitor* app if you want to analyze specific payloads.

2. Analyze the list of errors and warnings and decide which errors can be fixed in the Marketing system, and which errors must be fixed in the source system.
3. When you have fixed all possible errors, filter the list for all messages that have been fixed and choose *Restart All* to restart these imports.
4. You can then use the *Discard All* function to discard all other entries in the list.

## Related Information

[Best Records](#)

[Videos - Best Practices for Data Load \[page 388\]](#)

## 5.1.9 HTTP Response Status Codes

Every HTTP request that is received by a server is responded to with a 3-digit HTTP status code. They are grouped into five classes.

The class of a status code can be quickly identified by its first digit:

- 1xx: Informational
- 2xx: Success
- 3xx: Redirection
- 4xx: Client Error
- 5xx: Server Error

### i Note

HTTP errors are often caused by incorrect URLs, interconnected proxy servers, or by slow processing in a system. Some typical 4xx and 5xx error codes are described in the next section.

## 5.1.9.1 Client Errors (4xx)

Client errors, or HTTP status codes from 400 to 499, are the result of HTTP requests sent by a HTTP client. Even though these types of errors are client-related, it is often useful to know which error code a user is encountering to determine if the potential issue can be fixed by server configuration.

Error Code	Description
400 Bad Request	The 400 status code, or Bad Request error, means the HTTP request that was sent to the server has invalid syntax.
401 Unauthorized	The 401 status code, or an Unauthorized error, means that the user trying to access the resource has not been authenticated or has not been authenticated correctly. This means that the user must provide credentials to be able to view the protected resource. An example scenario where a 401 Unauthorized error would be returned is if a user tries to access a resource that is protected by HTTP authentication if enters invalid username and password.
403 Forbidden	The 403 status code, or a Forbidden error, means that the user made a valid request but the server is refusing to serve the request, due to a lack of permission to access the requested resource.
404 Not Found	The 404 status code, or a Not Found error, means that the user is able to communicate with the server but it is unable to locate the requested resource.

## 5.1.9.2 Server Errors (5xx)

Server errors, or HTTP status codes from 500 to 599, are returned by server when it is aware that an error has occurred or is otherwise not able to process the request.

Error Code	Description
500 Internal Server Error	The 500 status code, or Internal Server Error, means that server cannot process the request for an unknown reason.
502 Bad Gateway	The 502 status code, or Bad Gateway error, means that the server is a gateway or proxy server, and it is not receiving a valid response from the backend servers that should actually fulfill the request.
503 Service Unavailable	The 503 status code, or Service Unavailable error, means that the server is overloaded or under maintenance. This error implies that the service should become available at some point.
504 Gateway Timeout	The 504 status code, or Gateway Timeout error, means that the server is a gateway or proxy server, and it is not receiving a response from the backend servers within the allowed time period.

## 5.2 Contact Profiling

The following integration APIs are available in the context of contacts:

### [Contacts \[page 412\]](#)

Public OData API (`API_MKT_CONTACT_SRV` Version 4) for reading and writing master data about Contacts. Contacts are natural persons who interact with your company.

### [Interaction Contacts \[page 469\]](#)

Public OData API (`API_MKT_INTERACTION_CONTACT_SRV` Version 3) for Interaction Contacts. Interaction Contact is a generic term to group all natural persons (contacts, consumers, or suspects), companies and “unknowns”, who interact with your company.

### [Corporate Accounts \[page 512\]](#)

Public OData API (`API_MKT_CORPORATE_ACCOUNT_SRV` Version 3) for reading and writing master data about Corporate Accounts only. Corporate accounts are companies or organizations that interact with your company. The public API for Corporate Account supports operations on the `Corporate Account Business Object` and the `Marketing Permissions Business Object`.

### [Business Partners from SAP Cloud for Customer \[page 555\]](#)

Import business partners from SAP Cloud for Customer via `CUAN_BUSINESS_PARTNER_IMPORT_SRV` to marketing.

### [Import Business Partners \[page 574\]](#)

`CUAN_BUSINESS_PARTNER_IMPORT_SRV` for importing business partner data from external source systems, like, for example, SAP ERP, SAP CRM, SAP S/4HANA On Premise.

### [Products \[page 582\]](#)

Public OData API (`API_MKT_PRODUCT_SRV`) for Products.

### [Product Hierarchies and Categories \[page 604\]](#)

Public OData API (`API_MKT_PROD_CAT_HIERARCHY_SRV`) for Product Hierarchies and Categories.

### [Interactions \[page 615\]](#)

Public OData API (`API_MKT_INTERACTION_SRV`) for Interactions.

### [Interest Items \[page 648\]](#)

Public OData API (`API_MKT_INTEREST_SRV`) for InterestItems. An interest represents the content or subject of a contact's interaction.

### [Business Documents \[page 661\]](#)

Public OData API (`CUAN_BUSINESS_DOCUMENT_IMPORT_SRV`) for importing business documents, such as leads and opportunities, from external SAP or non-SAP systems to SAP Marketing Cloud. Use this version of the service when you want to import business documents related to Offers and Coupons. Lower versions are not suitable for this purpose.

### [Agreements \[page 681\]](#)

Public OData API (`API_MKT_AGREEMENT_SRV`) for agreements. An agreement can be any kind of customer contract, for example, a sales contract or a contract that comprises specific services.

### [Scores \[page 700\]](#)

Public OData API (`API_MKT_SCORE_SRV`) for Scores

### [Marketing Locations \[page 710\]](#)

Public OData API (`API_MKT_LOCATION`) for Marketing Locations. A marketing location is any physical or virtual location where a marketing activity can be conducted.

### [Classifications \(Deprecated\) \[page 722\]](#)

Public OData API (`API_MKT_ML_CLASSIFICATION`, deprecated) for reading and writing data about classifications. A classification is the truth about whether a certain event in the past or not. You define this event yourself.

### [Marketing Attribute Categories \[page 735\]](#)

OData API (`API_MKT_ATTRIBUTE_CATEGORY`) for writing master data about marketing attribute categories. Marketing attribute categories are freely-definable classifications of information that can be assigned to customers, for instance, to store their hobbies or education history.

### [Import Monitoring \[page 740\]](#)

Public OData API (`API_MKT_IMPORT_MONITORING`) for reading messages output for a specific data import using the import header ID. This service can be used by all API services whose imports are processed via the staging area.

## 5.2.1 Contacts

Public OData API (`API_MKT_CONTACT_SRV` Version 4) for reading and writing master data about Contacts. Contacts are natural persons who interact with your company.

### i Note

We recommend that you use the current version 4 of this service. Do **not** revert to using version 3, once you start using version 4 since this may result in data inconsistencies. However, if you want to continue using one of the previous versions, you'll find the help links here:

- Contacts API, Version 3: [Contacts API, Version 0003](#)
- Version 2 of Contacts, Interaction Contacts, and Corporate Accounts: [Contact, Interaction Contact, Corporate Account API, Version 0002](#)

## Technical Data

### ⚠ Caution

The API services available in SAP Marketing Cloud must not be used for mass read (GET) operations. In other words, you cannot use them for extracting all available data, for example, to extract millions of contacts or interactions from your marketing system.

Name of the Service	API_MKT_CONTACT
Authorizations	The following business catalog roles are required: <ul style="list-style-type: none"><li>• For version 4: <code>SAP_CEC_BC_MKT_API_IC4_PC</code></li><li>• For version 3: <code>SAP_CEC_BC_MKT_API_IC3_PC</code></li><li>• For version 2: <code>SAP_CEC_BC_MKT_API_IC2_PC</code></li></ul>
Communication Scenario ID	SAP_COM_0207
Component for Incidents	<ul style="list-style-type: none"><li>• CEC-MKT-DM-IC (Interaction Contacts)</li><li>• CEC-MKT-DM-PER (Permissions and Subscriptions)</li></ul>
	<h3>i Note</h3> <p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p>
OData Version	2.0
Root URI	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004</code>



Service Metadata URI `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/$metadata`



Field Extensibility Supported Yes. For more information, search for **extensibility** in [Structure of OData Service API\\_MKT\\_CONTACTS \[page 418\]](#).

#### i Note

You need to open the collapsible sections of the document first.

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$metadata?sap-documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Contacts Details Page</a> 	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li><li>Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Contacts API</a> 	General access link takes you directly to the <i>Contacts</i> metadata file. One-time registration or logon is required.

#### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
<code>sap:updatable</code>	read-only field
<code>sap:creatable</code>	read-only field
<code>nullable</code>	mandatory field

[Basic Concepts \[page 414\]](#)

The public API for Contacts `API_MKT_CONTACT_SRV` supports operations on the `Interaction Contact Business Object` and the `Marketing Permissions Business Object`. There is no separate public OData API for **marketing permissions**. The corresponding entity is part of this service since marketing permissions are always stored for a certain interaction contact.

[Structure of OData Service API\\_MKT\\_CONTACTS \[page 418\]](#)

This document describes the structure of the Public OData API service `API_MKT_CONTACT`.

[Payload Examples \[page 445\]](#)

Payload examples for `API_MKT_CONTACT`.

[Function Imports \[page 464\]](#)

Function imports are used to perform custom operations on an entity, which are typically not provided by OData operations. This section contains payload examples for the following function imports:

## 5.2.1.1 Basic Concepts

The public API for Contacts `API_MKT_CONTACT_SRV` supports operations on the `Interaction Contact Business Object` and the `Marketing Permissions Business Object`. There is no separate public OData API for **marketing permissions**. The corresponding entity is part of this service since marketing permissions are always stored for a certain interaction contact.

### i Note

For generally applicable recommendations and best practices, make sure you refer to the section [Best Practices and Recommended Package Sizes \[page 400\]](#).

## Switching to Version 4

Version 0004 of `API_MKT_CONTACT_SRV` is the **prerequisite** if you implement the B2B function `Contact-to-Account Relationships`. Regardless of whether you implement `Contact-to-Account Relationships`, we strongly recommend that you use version 0004 of this service for importing contacts.

If you switch from a lower version of the service to version 0004, please note the following:

- If you migrate from `API_MKT_CONTACT_SRV_0002` to `API_MKT_CONTACT_SRV_0004`, be aware that version 0002 had only one entity **MarketingPermission** for both `Permissions` and `Subscriptions`, whereas in version 0004 there are separate entities for `Permissions` and `Subscriptions`.
- Email, phone, mobile and fax IDs can now only be imported as `AdditionalID` entities. In previous versions, these sub-entities were imported as part of the `OriginData` entity. So what happens to existing IDs that you previously loaded using another service? The table explains what happens to existing email, fax, phone, and mobile IDs when you start importing data with version 0004.

Operations with API\_MKT\_CONTACT\_0004

If You Perform This Operation	Are Existing IDs Deleted?
PUT on OriginData	Yes
PUT on OriginData and AdditionalIDs	Yes
PUT on OriginData and AdditionalIDs and Function Import ContactOriginDeleteAdditionalIDs	Yes
PUT on AdditionalIDs	No
PUT on AdditionalIDs and Function Import ContactOriginDeleteAdditionalIDs	Yes
PATCH on OriginData	No
PATCH on OriginData and AdditionalIDs	No
PATCH on OriginData and AdditionalIDs and Function Import ContactOriginDeleteAdditionalIDs	Yes
PATCH on AdditionalIDs	No
PATCH on AdditionalIDs and Function Import ContactOriginDeleteAdditionalIDs	Yes
Function Import ContactOriginDeleteAdditionalIDs	Yes

Sample Use Cases

If You Want To	The Recommended Method Is
Perform a full update of a contact in a running system	PATCH on OriginData, Function Import ContactOriginDeleteAdditionalIDs, and PATCH on AdditionalIDs
Add an additional ID to an existing contact, for example, a cookie ID	PUT on AdditionalID
Read contact data out of the system	GET Request on relevant entity

## Processing Info and Best Practices

- The minimum data required when importing contacts is an ID, an ID Origin, a timestamp, and **at least one other attribute**.
- **When to use PUT and PATCH:**
  - PUT requests are most suitable for an initial data import, for example, when you want to create a new contact. A PUT request requires that you always send all properties. Any properties that you omit are overwritten by blank entries. That is, any existing entries are deleted. If no record is found, a new record is created. In other words, the PUT request functions as a full **upsert**.
  - We recommend that you use PATCH requests for all other imports. A PATCH request updates only the properties provided in the request body and leaves everything untouched that was not provided. So,

you can omit all properties that are not to be changed. Like the `PUT` request, if no record is found, a new record is created with the available properties. In other words, the `PATCH` request functions as a delta **upsert**.

- An additional advantage of using `PATCH` is that you specify your own sequence ID. For this reason, it is more flexible than a `PUT` operation, where the sequence ID is set by default and cannot be changed.
- Basically, since you can use `PATCH` with the same payload as you would use for `PUT`, the `PATCH` operation is more universal and you can work with it exclusively.
- We recommend that you don't mix `PUT` and `PATCH` operations. Doing so can lead to unwanted results since a `PUT` operation is processed before a `PATCH`.
- **Do not combine a DELETE operation with other OData operations in one changeset.**  
We recommend that you do not combine the OData operations `PUT`, `PATCH`, `POST`, with a `DELETE` operation in the same changeset. For example, let's say you want to update data for Contact A by adding an additional email address and at the same time delete a mobile number that is no longer valid. So, you send a `PUT` operation on the `AdditionalId` entity with the new email address and a `DELETE` operation within the same changeset. One of these operations could cancel out the other and the resulting dataset will not be as intended.  
**Recommended Practice:** For such combined operations including a `DELETE` operation, we recommend that you always use the relevant function import, which allows deletion of specific entities, together with the appropriate OData operation `PUT`, `PATCH`, or `POST` within the same changeset.
- Batch requests are submitted as a single `HTTP POST` request to the `$batch` endpoint of a service as described in [[OData-URI](#)]. The batch request must contain a header parameter `content-type`, specifying the value `multipart/mixed` and `boundary=batch`.
- **Use of codes versus free text:** The properties listed in the left column of the table require code values. Incorrect codes will result in import errors, indicating that the corresponding code is not valid. If you are not thoroughly familiar with the internal codes available in SAP Marketing for these properties, you should use properties that allow a free text. For example, if you do not know that **DE** is the country code for **Germany**, you can use `Germany` as the free text.

Code in SAP Marketing	Free Text Property
Country	CountryName
Industry	IndustryName
Department	DepartmentName
Function	ContactFunctionName
GenderCode	GenderCodeName
Language	LanguageName
MaritalStatus	MaritalStatusName
AddressRegion	RegionName
FormOfAddress	FormOfAddressName

You must map your free text names to the available codes in the [Map Free Text](#) app. For more information, see [Map Free Texts](#).

- Do not mix different types of services for operations involving the same data source. For example, when importing contacts from a web shop, do not use the `CUAN_IMPORT` service for a `PUT` operation and then the `API_MKT_CONTACT` service to `PATCH` contacts. You can, however, migrate from `CUAN_IMPORT` to the `API*` services.

- The origin that you pass via the property *ContactOrigin* cannot be shareable. If the **main** origin is set to *Shareable*, this will trigger an error. For more information, see [Configuring Origins](#). You can view sample payloads and test the API at [https://api.sap.com/api/API\\_MKT\\_CONTACT\\_SRV\\_0004/resource](https://api.sap.com/api/API_MKT_CONTACT_SRV_0004/resource).

- **UTC Timestamp of Permissions:**

The UTC timestamp of permissions cannot lie in the future.

When you import permissions, they must not have a timestamp that lies in the future. The timestamp of imported permissions is always in UTC. The field name in the OData service is called *PermissionUTCDateTime*. If you want to use your local timestamp, you have to add the time zone information, that is, your local time zone together with the time zone offset or enter a timestamp that is converted to UTC.

#### Example

The date and time information is adapted by the standard time difference (offset) with +01:00 for Central European Time (CET) or -05:00 for Eastern Standard Time (EST). For example:  
2019-01-01T12:00:00+01:00

If you live east of UTC and enter your timestamp in your local time zone without time zone offset, this will result in a future timestamp. For example, you live in Germany and your local time is 8 a.m. on November, 28. If you enter this as the UTC timestamp without a time zone offset, the UTC permission timestamp will show as 8 a.m., November 28, while in the UTC time zone it is 7 a.m., November 28. You have created a UTC permission timestamp that lies in the future and is invalid.

## Error Messages

If the OData service is not accessible, for example due to missing authorization, or because the system is not available, a corresponding HTTP status code is returned.

If the OData service is accepted by the gateway component in the SAP Marketing Cloud system, the HTTP status code 201 or 204 is returned. Any processing errors are recorded in the SAP Marketing Cloud system in the *Import Monitor* app, where they can be monitored, restarted and discarded.

By default, data processing for contacts, interaction contacts, corporate accounts, or marketing permissions is asynchronous. In most cases an OK response, such as a receipt notification, is returned almost immediately. An exception to this would be data uploads that might contain severe errors, such as parse or format errors, and so would not return an OK response but an error message. The data you upload lands in a staging area, where it is then further processed. You can change the default setting to synchronous processing by setting the property *Sap-Cuan-ForceSynchronousProcessing* to True. In this case, any error messages are returned as soon as they are detected.

To view the processing status and to check for errors or success messages, you must launch the *Import Monitor* app. Messages for marketing permissions in this app are displayed under the API for Contact, API for Interaction Contacts, or API for Corporate Accounts depending on the API OData service you use. In the event of errors, you can restart or discard the import in the *Import Monitor*.

For more information, see [HTTP Response Status Codes \[page 408\]](#).

## Field Extensibility

You can add customer-specific fields using the *Custom Fields* app. For more information about how to do this, see [Custom Fields App and Custom Logic App](#).

Please enable the Data Source under UIs and Reports: `API_MKT_CONTACT_SRV 0004`

Parent topic: [Contacts \[page 412\]](#)

## Related Information

[Structure of OData Service API\\_MKT\\_CONTACTS \[page 418\]](#)

[Payload Examples \[page 445\]](#)

[Function Imports \[page 464\]](#)

[Best Practices and Recommended Package Sizes \[page 400\]](#)

## 5.2.1.2 Structure of OData Service API\_MKT\_CONTACTS

This document describes the structure of the Public OData API service `API_MKT_CONTACT`.

Make sure you read these topics before you start:

- [Best Practices and Recommended Package Sizes \[page 400\]](#)
- [Basic Concepts \[page 414\]](#)

## Request Header

The request header contains the additional header fields listed in the table. Remember to include at least the mandatory request header fields in each payload.

Property	Example	Description	Max.Length	Mandatory
<code>Sap-Cuan-RequestTimestamp</code>	'2017-09-28T12:13:14'	Timestamp of the import run in this format.		X

Property	Example	Description	Max.Length	Mandatory
Sap-Cuan-SequenceId	PatchAddress	This defines a set of fields that are to be updated, for example, address fields, which can be interpreted as a field group. The combination of the header fields Sap-Cuan-SequenceId and Sap-Cuan-RequestTimestamp is used to check the sequence of the data received. If the data that is received has a timestamp older than already imported data, it is ignored.		X (only mandatory for Patch Mode)
Sap-Cuan-SourceSystemType	EXT	Type of source system. This is a free text field.	20	
Sap-Cuan-SourceSystemId	HYBRIS	Identifier of source system. This is a free text field.	255	
Sap-Cuan-ForceSynchronousProcessing	X	<p>This flag is deselected by default, which means that uploaded data is processed asynchronously. On upload, a success message is output immediately, unless there are errors such as authorization issues or bad requests. Objects are uploaded to the staging area and processed successively from there. All status messages can be displayed in the <a href="#">Import Monitor</a> app.</p> <p>You can force imports to be processed <b>synchronously</b> by setting this flag. In this case, an error message will be returned as soon as an error is detected. Such error messages are output in the <a href="#">Import Monitor</a> app</p>		
Sap-Cuan-ReferenceId	345g67980907	External reference of the inbound message	32	

## Entity Sets

The Contact OData API provides the following entity sets:

Entity Set	Description	Path
Contacts	This entity contains all contact information from the contact's best record.	/Contacts
AccountTeamMembers	This entity contains information about the account team members.	/AccountTeamMembers
AdditionalIDs	This entity contains information about contacts' additional IDs.	/AdditionalIDs
ContactOriginData	This entity contains contact origin data. <b>i Note</b> The property <code>OriginDataLastChgUTCDateTime</code> is mandatory. It must be specified.	/ContactOriginData
ContactRelationData	This entity contains information about contacts' relationship data. <b>i Note</b> The property <code>RelationDataLastChgUTCDateTime</code> is mandatory. It must be specified.	/ContactRelationData
ContactRelationAdditionalIDs	This entity contains information about additional IDs of contact relationships.	/ContactRelationAdditionalIDs
MarketingAttributes	This entity contains information about marketing attributes.	/MarketingAttributes
MarketingAreas	This entity contains information about marketing areas.	/MarketingAreas
MarketingPermissions	This entity contains information about marketing permissions.	/MarketingPermissions
MarketingSubscriptions	This entity contains information about marketing subscriptions.	/MarketingSubscriptions



Entity Set	Description	Path
MarketingLocations	This entity contains information about marketing locations.	/MarketingLocations

## Contacts

**GET: Entity Path:** /Contacts

**Field Extensibility:** The following business contexts are relevant: *Marketing: Contact* and *Marketing: Contact and Corporate Account*

You can perform the following operations on the `Contacts` entity set:

HTTP Method	Description	Path
<a href="#">GET</a>	Get a list of contacts. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	/Contacts?\$top=1
	<div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>A maximum of 5000 contacts can be fetched in a single request</li> <li>Specification of TOP is mandatory.</li> </ul> </div>	
	Get the details of a specific contact using the <code>Contact</code> UUID.	/Contacts(guid'<Contact UUID>')

## AccountTeamMembers

You can perform the following operations on the `AccountTeamMember` entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/AccountTeamMembers`
- **PUT, PATCH, or DELETE in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/$batch`
- **PUT, PATCH, or DELETE in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/AccountTeamMembers(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',TeamMemberID='<TeamMemberID>',Role='<Role>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of account team members.	<p>/AccountTeamMembers?\$top=1</p> <p>This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 account team members can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>
	Get the details of a specific account team member.	This operation is not supported.
<i>POST (Batch)</i>	Update or create an account team member in batch mode.(Full Update)	<p>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$bat ch</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• The maximum number of requests in a changeset is 10000 (ten thousand).</li> </ul> </div>
	Delete an account team member in batch mode.	<p>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$bat ch</p>
	Add one new account team member.	<p>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$bat ch</p>
<i>PUT</i>	Update or create an account team member.	<p>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ AccountTeamMembers (ContactID='&lt;ContactID&gt;', Contac tOrigin='&lt;ContactOrigin&gt;', TeamM emberID='&lt;TeamMemberID&gt;', Role=' &lt;Role&gt;')</p>

HTTP Method	Description	Path
<i>PATCH</i>	Add one new account team member.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ AccountTeamMembers (ContactID='&lt;ContactID&gt;', Contac tOrigin='&lt;ContactOrigin&gt;', TeamM emberID='&lt;TeamMemberID&gt;', Role=' &lt;Role&gt;')</pre>
<i>DELETE</i>	Delete an account team member.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ AccountTeamMembers (ContactID='&lt;ContactID&gt;', Contac tOrigin='&lt;ContactOrigin&gt;', TeamM emberID='&lt;TeamMemberID&gt;', Role=' &lt;Role&gt;')</pre>

## AdditionalIDs

You can perform the following operations on the AdditionalIDs entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/AdditionalIDs`
- **PUT, PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/AdditionalIDs(ContactID='<ContactID>', ContactOrigin='<ContactOrigin>', ContactAdditionalOrigin='<ContactAdditionalOrigin>', ContactAdditionalID='<ContactAdditionalID>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of additional IDs by Contact ID and ID Origin.	<p>/AdditionalIDs?\$top=1</p> <p>This method supports standard OData parameters such as \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 additional IDs can be fetched in a single request.</li> <li>• Specification of TOP is mandatory.</li> <li>• \$filter is not supported for additional IDs.</li> </ul> </div>
	Get the details of a specific additional ID.	<p>/AdditionalIDs ('&lt;ContactID&gt;,&lt;ContactOrigin&gt;,&lt;ContactAdditionalOrigin&gt;,&lt;ContactAdditionalID&gt;')</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• The maximum number per changeset is 10000 (ten thousand) entities.</li> </ul> </div>
<i>POST (Batch)</i>	Update or create an additional ID in batch mode.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$batch</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• The maximum number of requests in a changeset is 10000 (ten thousand).</li> </ul> </div>
	Add one new additional ID.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$batch</p>

HTTP Method	Description	Path
<i>PUT</i>	Update or create an additional ID.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ AdditionalIDs (ContactID='&lt;Conta ctID&gt;',ContactOrigin='&lt;ContactO rigin&gt;',ContactAdditionalOrigin ='&lt;ContactAdditionalOrigin&gt;',Co ntactAdditionalID='&lt;ContactAddi tionalID&gt;')</pre>
<i>PATCH</i>	Add one new additional ID.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ AdditionalIDs (ContactID='&lt;Conta ctID&gt;',ContactOrigin='&lt;ContactO rigin&gt;',ContactAdditionalOrigin ='&lt;ContactAdditionalOrigin&gt;',Co ntactAdditionalID='&lt;ContactAddi tionalID&gt;')</pre>

## ContactOriginData

You can perform the following operations on the `ContactOriginData` entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/ContactOriginData`
- **PUT, PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/ContactOriginData (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>')`

**Field Extensibility:** The following business contexts are relevant: *Marketing: Contact* and *Marketing: Contact and Corporate Account*

HTTP Method	Description	Path
<i>GET</i>	Get a list of Contact Origin Data.	<p>/ContactOriginData?\$top=1</p> <p>This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 contact origin data entities can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>
	Get the details of specific contact origin data.	<p>/ContactOriginData ('&lt;ContactID&gt;, &lt;ContactOrigin&gt;')</p>
<i>POST (Batch)</i>	Update or create contact origin data in batch mode. This creates a contact if the contact not exist.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$batch</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>i Note</b></p> <p>The property OriginDataLastChgUTCDateTime is mandatory and must be specified.</p> </div>
	Delta Update PATCH attributes of the entity ContactOriginData.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$batch</p>
<i>PUT</i>	Update or create contact origin data. This creates a contact if the contact not exist.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/ContactOriginData (ContactID='&lt;ContactID&gt;', ContactOrigin='&lt;ContactOrigin&gt;')</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>i Note</b></p> <p>The property OriginDataLastChgUTCDateTime is mandatory and must be specified.</p> </div>

HTTP Method	Description	Path
<i>PATCH</i>	Delta Update PATCH attributes of the entity ContactOriginData.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ ContactOriginData(ContactID=' &lt;ContactID&gt;',ContactOrigin='&lt;Co ntactOrigin&gt;')</pre>

**i Note**

The property OriginDataLastChgUTCDateTime is mandatory and must be specified.

## ContactRelationData

You can perform the following operations on the ContactRelationData entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/ContactRelationData`
- **PUT, PATCH and DELETE in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/$batch`
- **PUT, PATCH and DELETE in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/ContactRelationData(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',RelationshipCategory='<RelationshipCategory>',ReltdIntactnContactID='<ReltdIntactnContactID>',ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of contact relationship data by Contact ID and ID Origin.	<pre>/ContactRelationData?\$top=1</pre> <p>This method supports standard OData parameters such as \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby</p>

**i Note**

- A maximum of 5000 entities can be fetched in a single request.
- Specification of TOP is mandatory.
- \$filter is not supported for additional IDs.

HTTP Method	Description	Path
	Get the details of a specific additional ID.	/ ContactRelationData (ContactID=' <ContactID>',ContactOrigin='<Co ntactOrigin>',RelationshipCateg ory='<RelationshipCategory>',Re ltdIntactnContactID='<ReltdInta ctnContactID>',ReltdIntactnCont actOrigin='<ReltdIntactnContact Origin>')
<i>POST (Batch)</i>	Update or create contact relationship data in batch mode.	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$bat ch <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>The property RelationDataLastChgUTC-DateTime is mandatory and must be specified.</p> </div>
	Add one new contact relationship data.	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$bat ch
<i>PUT</i>	Update or create contact relationship data.	/ ContactRelationData (ContactID=' <ContactID>',ContactOrigin='<Co ntactOrigin>',RelationshipCateg ory='<RelationshipCategory>',Re ltdIntactnContactID='<ReltdInta ctnContactID>',ReltdIntactnCont actOrigin='<ReltdIntactnContact Origin>')
		 <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>The property RelationDataLastChgUTC-DateTime is mandatory and must be specified.</p> </div>



HTTP Method	Description	Path
<i>PATCH</i>	Add one new contact relationship data.	/ ContactRelationData (ContactID='<ContactID>', ContactOrigin='<ContactOrigin>', RelationshipCategory='<RelationshipCategory>', ReltdIntactnContactID='<ReltdIntactnContactID>', ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>')
<i>DELETE</i>	Delete contact relationship data.	/ ContactRelationData (ContactID='<ContactID>', ContactOrigin='<ContactOrigin>', RelationshipCategory='<RelationshipCategory>', ReltdIntactnContactID='<ReltdIntactnContactID>', ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>')

## ContactRelationAdditionalIDs

You can perform the following operations on the ContactRelationAdditionalIDs entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/ContactRelationAdditionalIDs`
- **PUT, PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/ContactRelationAdditionalIDs (ContactID='<ContactID>', ContactOrigin='<ContactOrigin>', RelationshipCategory='<RelationshipCategory>', ReltdIntactnContactID='<ReltdIntactnContactID>', ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>', CntctRelationAdditionalID='<CntctRelationAdditionalID>', CntctRelationAdditionalOrigin='<CntctRelationAdditionalOrigin>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of additional IDs of a contact relationship.	<p>/ContactRelationAdditionalIDs?\$top=1</p> <p>This method supports standard OData parameters such as \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 additional IDs can be fetched in a single request.</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>
	Get the details of a specific additional ID of a contact relationship.	<p>/ContactRelationAdditionalIDs(ContactID='&lt;ContactID&gt;', ContactOrigin='&lt;ContactOrigin&gt;', RelationshipCategory='&lt;RelationshipCategory&gt;', ReltdIntactnContactID='&lt;ReltdIntactnContactID&gt;', ReltdIntactnContactOrigin='&lt;ReltdIntactnContactOrigin&gt;', CntctRelationAdditionalID='&lt;CntctRelationAdditionalID&gt;', CntctRelationAdditionalOrigin='&lt;CntctRelationAdditionalOrigin&gt;')</p>
<i>POST (Batch)</i>	Update or create an additional ID of a contact relationship in batch mode.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$batch</p>
	Add one new additional ID of a contact relationship.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$batch</p>

HTTP Method	Description	Path
<i>PUT</i>	Update or create an additional ID of a contact relationship.	<pre> https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ ContactRelationAdditionalIDs (Co ntactID='&lt;ContactID&gt;', ContactOr igin='&lt;ContactOrigin&gt;', Relation shipCategory='&lt;RelationshipCate gory&gt;', ReltdIntactnContactID='&lt; ReltdIntactnContactID&gt;', ReltdIn tactnContactOrigin='&lt;ReltdIntac tnContactOrigin&gt;', CntctRelation AdditionalID='&lt;CntctRelationAdd itionalID&gt;', CntctRelationAdditi onalOrigin='&lt;CntctRelationAddit ionalOrigin&gt;') </pre>
<i>PATCH</i>	Add one new additional ID of a contact relationship.	<pre> https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ ContactRelationAdditionalIDs (Co ntactID='&lt;ContactID&gt;', ContactOr igin='&lt;ContactOrigin&gt;', Relation shipCategory='&lt;RelationshipCate gory&gt;', ReltdIntactnContactID='&lt; ReltdIntactnContactID&gt;', ReltdIn tactnContactOrigin='&lt;ReltdIntac tnContactOrigin&gt;', CntctRelation AdditionalID='&lt;CntctRelationAdd itionalID&gt;', CntctRelationAdditi onalOrigin='&lt;CntctRelationAddit ionalOrigin&gt;') </pre>

## Projections

A projection is the technical term used for the automatically generated best record of a contact from the perspective of the specific relationship the contact has with an account, within a specific marketing area. For more information, see [Glossary](#).

You can perform the following operation on the `Projections` entity set:

**GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/Projections`

HTTP Method	Description	Path
<a href="#">GET</a>	Get a list of projections.	/Projections?\$top=2  This method supports standard OData parameters such as \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby

## MarketingAttributes

You can perform the following operations on the `MarketingAttributes` entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/MarketingAttributes`
- **PUT, PATCH, or DELETE in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/$batch`
- **PUT, PATCH, or DELETE in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/MarketingAttributes(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',MarketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValue>')`

**Field Extensibility:** The following business context is relevant: [Marketing: Marketing Attributes for Contacts](#)

HTTP Method	Description	Path
<a href="#">GET</a>	Get a list of marketing attributes by Contact ID and ID Origin.	/MarketingAttributes?\$top=1  This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby
	Get the details of a specific marketing attribute.	/MarketingAttributes('<ContactID>', '<ContactOrigin>', '<MarketingAttributeCategory>', '<MarketingAttributeValue>')

### Note

- A maximum of 5000 marketing attributes can be fetched in a single request
- Specification of TOP is mandatory.

HTTP Method	Description	Path
<i>POST (Batch)</i>	Update or create marketing attributes in batch mode.	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$bat ch
	Delete marketing attributes in batch mode.	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$bat ch
	Add one new marketing attribute.	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$bat ch
<i>PUT</i>	Update or create marketing attributes.	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ MarketingAttributes(ContactID=' <ContactID>',ContactOrigin=' <ContactOrigin>',MarketingAttri buteCategory='<MarketingAttribu teCategory>',MarketingAttribute Value='<MarketingAttributeValue >')
<i>PATCH</i>	Add one new marketing attribute.	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ MarketingAttributes(ContactID=' <ContactID>',ContactOrigin=' <ContactOrigin>',MarketingAttri buteCategory='<MarketingAttribu teCategory>',MarketingAttribute Value='<MarketingAttributeValue >')

HTTP Method	Description	Path
<i>DELETE</i>	Delete marketing attributes.	<pre> https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ MarketingAttributes(ContactID=' &lt;ContactID&gt;',ContactOrigin=' &lt;ContactOrigin&gt;',MarketingAttri buteCategory='&lt;MarketingAttribu teCategory&gt;',MarketingAttribute Value='&lt;MarketingAttributeValue &gt;') </pre>

## MarketingAreas

You can perform the following operations on the MarketingAreas entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004`
- **PUT, PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/MarketingAreas(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing areas by Contact ID and ID Origin.	<pre> /sap/opu/odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ Contacts? \$expand=MarketingAreas&amp;\$top=2 </pre> <p>This method supports standard OData parameters such as <code>\$filter</code>, <code>\$select</code>, <code>\$top</code>, <code>\$skip</code>, <code>\$count</code>, <code>\$inlinecount</code>, and <code>\$orderby</code></p>

### i Note

- A maximum of 5000 marketing areas can be fetched in a single request
- Specification of TOP is mandatory.

HTTP Method	Description	Path
	Get the details of a specific marketing area.	/ MarketingAreas ('<ContactID>,<ContactOrigin>,InteractionContactMktgArea')
<i>POST (Batch)</i>	Update or create marketing areas in batch mode.	https:// <Server>:<Port>/sap/opu/odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$batch/ MarketingAreas ('<ContactID>,<ContactOrigin>,InteractionContactMktgArea')
	Add one new marketing area.	https:// <Server>:<Port>/sap/opu/odata/SAP/ API_MKT_CONTACT_SRV;v=0004/\$batch
<i>PUT</i>	Update or create marketing areas.	https:// <Server>:<Port>/sap/opu/odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ MarketingAreas (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>')
<i>PATCH</i>	Add one new marketing area.	https:// <Server>:<Port>/sap/opu/odata/SAP/ API_MKT_CONTACT_SRV;v=0004/ MarketingAreas (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>')

## MarketingLocations

You can perform the following operations on the MarketingAreas entity set:

- **GET:** /MarketingAreas ('https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_CONTACT\_SRV;v=0004/MarketingLocations
- **PUT, PATCH in batch:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_CONTACT\_SRV;v=0004/\$batch
- **PUT, PATCH in a single operation:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_CONTACT\_SRV;v=0004/

MarketingLocations (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',  
MarketingLocationExternalID='< MarketingLocationExternalID>')

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing locations by Contact ID and ID Origin.	/MarketingLocations?\$stop=1  This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby
<i>POST (Batch)</i>	Update or create marketing areas in batch mode.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$batch
	Add one new marketing area	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/\$batch
<i>PUT</i>	Update or create marketing locations	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/MarketingLocations (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',MarketingLocationExternalID='<MarketingLocationExternalID>')
<i>PATCH</i>	Add one new marketing location	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/MarketingLocations (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',MarketingLocationExternalID='<MarketingLocationExternalID>')

## MarketingPermissions

**Entity Path:** /MarketingPermissions

**Field Extensibility:** The following business context is relevant: *Marketing: Marketing Permissions*. Custom fields for business object MKT\_PERMISSION (Marketing: Permission) are only supported if you use **version 2** or **version 3** of the API\_MKT\_CONTACT service.



### i Note

- For all HTTP operations both `$batch` requests and single requests can be used.
- Interactions are assigned when marketing permissions are created or updated to allow for analysis of contacts.

You can perform the following operations on the `MarketingPermissions` entity set:

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing permissions by Contact ID and ID Origin. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	<code>/MarketingPermissions?\$top=1</code>
	<b>i Note</b> <ul style="list-style-type: none"><li>• A maximum of 5000 marketing permissions can be fetched in a single request</li><li>• Specification of TOP is mandatory.</li></ul>	
<i>PATCH</i>	Update or create marketing permissions. This creates a marketing permission if the permission does not exist.  Delta Update of PATCH attributes of the entity <code>MarketingPermission</code> .	<code>/MarketingPermissions (ContactID='&lt;ContactID&gt;', ContactOrigin='&lt;ContactOrigin&gt;', ContactPermissionID='&lt;ContactPermissionID&gt;', ContactPermissionOrigin='&lt;ContactPermissionOrigin&gt;', MarketingArea='&lt;MarketingArea&gt;', CommunicationMedium='&lt;CommunicationMedium&gt;')</code>
<i>PUT</i>	Update or create marketing permissions. This creates a marketing permission if the permission does not exist.	<code>/MarketingPermissions (ContactID='&lt;ContactID&gt;', ContactOrigin='&lt;ContactOrigin&gt;', ContactPermissionID='&lt;ContactPermissionID&gt;', ContactPermissionOrigin='&lt;ContactPermissionOrigin&gt;', MarketingArea='&lt;MarketingArea&gt;', CommunicationMedium='&lt;CommunicationMedium&gt;')</code>

## Marketing Permission Property Descriptions

The table describes the properties for the `MarketingPermissions` entity.

## MarketingPermissions Property Names and Descriptions

Property Name	Property Description	Usage
ContactID	<p>The ContactID and ContactOrigin identify the contact uniquely.</p> <p><b>Example:</b> a business partner ID from the CRM system.</p>	
ContactOrigin	<p>The ContactID and ContactOrigin identify the contact uniquely.</p> <p>The ContactID will not be saved to the MarketingPermission but is only used to derive a unique ContactUUID. This data will not be returned in GET requests.</p> <p><b>Example:</b> SAP_CRM_BUPA</p>	
ContactPermissionID	<p>The ContactPermissionID and ContactPermissionOrigin store marketing permissions.</p> <p><b>Example:</b> first.lastname@mail.de</p>	Mandatory
ContactPermissionOrigin	<p>The ContactPermissionID and ContactPermissionOrigin store marketing permissions.</p> <p>ContactPermissionOrigin is the origin of a contact ID that stores marketing permissions. The origin indicates the source of an ID. By defining the origin, you determine that a contact with an ID associated to a source can be analyzed.</p> <p><b>Example:</b> EMAIL</p> <p>You can configure origins of contact IDs in the <a href="#">Configuring Origins</a> configuration app.</p>	Mandatory
ContactPermissionOriginName	<p>Description of property <b>ContactPermissionOrigin</b></p>	Read-Only
MarketingArea	<p>Identifies an area of responsibility or an organizational unit.</p> <p>You use a marketing area to restrict access to instances of an object, such as campaign, email message, email template, target group, or permission.</p>	<p>Mandatory</p> <p>The MarketingArea property field must be passed, but can be left empty.</p>

Property Name	Property Description	Usage
MarketingAreaName	Description of property <b>MarketingArea</b>	Read-Only
CommunicationMedium	<p>Represents the type of permission, for example, <b>EMAIL</b> or <b>PHONE</b>.</p> <p>You can configure communication media in the <a href="#">Managing Interaction Content</a> configuration app.</p>	Mandatory
CommunicationMediumName	Description of property <b>CommunicationMedium</b>	Read-Only
ContactUUID	Unique ID of a contact in SAP Marketing Cloud .	Read-Only The field value is returned internally.
PermissionGranted	The permission can be YES ( <b>Y</b> ) or NO ( <b>N</b> ).	Mandatory
PermissionUTCDateTime	<p>This is the timestamp for when the permission was given or removed.</p> <div data-bbox="603 1039 991 1189" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p><b>i Note</b></p> <p>The timestamp must not be <i>initial</i> or <i>null</i>.</p> </div>	Mandatory
PermissionUUID	Unique ID of a permission in SAP Marketing Cloud .	Read-Only The field value is returned internally.
PermissionSourceObject	<p>This field provides information on the source of the permission, that is, where it came from. For example, the ID of a landing page.</p> <p>This field can be filled with freetext.</p>	<p>If you enter a value for the PermissionSourceObject property, you must also specify a value for the PermissionSourceObjectType.</p> <p>Both fields must be filled or left empty.</p>
PermissionSourceObjectType	<p>This field provides information on the source of the permission and its type. For example, the business object name of a landing page.</p> <p>This field can be filled with freetext.</p>	
PermissionSourceSystem	<p>This is the system that stores the permission. For example, your local system ID.</p> <p>This field can be filled with freetext.</p>	<p>If you enter a value for the PermissionSourceSystem property, you must also specify a value for the PermissionSourceSystemType.</p> <p>Both fields must be filled or left empty.</p>

Property Name	Property Description	Usage
PermissionSourceSystemType	This is the type of system where the permission is stored. For example, SAP_CEI.  This field can be filled with freetext.	
PermissionSourceCommMedium	Indicates where the permission comes from, such as WEB, EMAIL, or PHONE. In case PermissionSourceCommMedium is not filled, this property is set to WEB.	Mandatory
PermissionSourceCommMediumName	Description of property <b>Permission-SourceCommMedium</b>	Read-Only
PermissionIsImplicit	If the system sets this field to TRUE, then it is an implicit permission, which is determined by country-specific regulation.  If the system sets this field to FALSE, the contact has given this permission explicitly.	Read-Only
IsConfirmationRequired	This is a boolean parameter. If the parameter is set to TRUE, the permission is stored using the double opt-in or opt-out process.  If the property is not specified in the payload or it is set to FALSE the permission is directly stored.	
LastChangedByUser	Name of the user who has changed the permissions last.	Read-Only
LastChangeDateTime	Date and time of the last permission change.	Read-Only
PermissionNoteText	A text to describe a permission change.	

## MarketingSubscriptions

**Entity Path:** /MarketingSubscriptions

**Field Extensibility:** The following business context is relevant: [Marketing: Marketing Permissions](#). Custom fields for business object MKT\_PERMISSION (Marketing: Permission) are only supported if you use **version 2** or **version 3** of the API\_MKT\_CONTACT service.

### i Note

- For all HTTP operations both \$batch requests and single requests can be used.
- Interactions are assigned when marketing permissions are created or updated to allow for analysis of contacts.

You can perform the following operations on the `MarketingSubscriptions` entity set:

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing subscriptions by Contact ID and ID Origin. This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby	<code>/MarketingSubscriptions?\$top=1</code>
	<div data-bbox="443 745 1080 920"><h3>i Note</h3><ul style="list-style-type: none"><li>• A maximum of 5000 marketing subscriptions can be fetched in a single request</li><li>• Specification of TOP is mandatory.</li></ul></div>	
<i>PATCH</i>	Update or create subscriptions. This creates a subscription if the subscription does not exist.	<code>/MarketingSubscriptions(ContactID='&lt;ContactID&gt;',ContactOrigin='&lt;ContactOrigin&gt;',ContactPermissionID='&lt;ContactPermissionID&gt;',ContactPermissionOrigin='&lt;ContactPermissionOrigin&gt;',CommunicationMedium='&lt;CommunicationMedium&gt;',SubscriptionTopic='&lt;SubscriptionTopic&gt;')</code>
<i>PUT</i>	Update or create subscriptions. This creates a subscription if the subscription does not exist.  Delta Update of PATCH attributes of the entity MarketingSubscriptions.	<code>/MarketingSubscriptions(ContactID='&lt;ContactID&gt;',ContactOrigin='&lt;ContactOrigin&gt;',ContactPermissionID='&lt;ContactPermissionID&gt;',ContactPermissionOrigin='&lt;ContactPermissionOrigin&gt;',CommunicationMedium='&lt;CommunicationMedium&gt;',SubscriptionTopic='&lt;SubscriptionTopic&gt;')</code>

## Marketing Subscription Property Descriptions

The table describes the properties for the `MarketingSubscription` entity.

## MarketingSubscription Property Names and Descriptions

Property Name	Property Description	Usage
ContactID	<p>The ContactID and ContactOrigin identify the contact uniquely.</p> <p><b>Example:</b> a business partner ID from the CRM system.</p>	
ContactOrigin	<p>The ContactID and ContactOrigin identify the contact uniquely.</p> <p>The ContactID will not be saved to the MarketingSubscription but is only used to derive a unique ContactUUID. This data will not be returned in GET requests.</p> <p><b>Example:</b> SAP_CRM_BUPA</p>	
ContactSubscriptionID	<p>The ContactPermissionID and ContactSubscriptionOrigin store marketing subscription.</p> <p><b>Example:</b></p>	Mandatory
ContactSubscriptionOrigin	<p>The ContactSubscriptionID and ContactSubscriptionOrigin store marketing subscriptions.</p> <p>ContactSubscriptionOrigin is the origin of a contact ID that stores marketing subscriptions. The origin indicates the source of an ID. By defining the origin, you determine that a contact with an ID associated to a source can be analyzed.</p> <p><b>Example:</b> EMAIL</p> <p>You can configure origins of contact IDs in the <a href="#">Configuring Origins</a> configuration app.</p>	Mandatory
ContactSubscriptionOriginName	<p>Description of property <b>ContactSubscriptionOrigin</b></p>	Read-Only
CommunicationMedium	<p>Represents the type of subscription, for example, <b>EMAIL</b> or <b>PHONE</b>.</p> <p>You can configure communication media in the <a href="#">Managing Interaction Content</a> configuration app.</p>	Mandatory

Property Name	Property Description	Usage
CommunicationMediumName	Description of property <b>CommunicationMedium</b>	Read-Only
ContactUUID	Unique ID of a contact in SAP Marketing Cloud .	Read-Only The field value is returned internally.
SubscriptionUUID	Unique ID of a subscription in SAP Marketing Cloud .	Read-Only The field value is returned internally.
SubscriptionUTCDateTime	This is the timestamp for when the subscription was given or removed.	Mandatory
<div style="background-color: #f0f0f0; padding: 10px; border-left: 2px solid #0070c0;"> <p><b>i Note</b></p> <p>The time stamp must not be initial or null.</p> </div>		
SubscriptionSignUpExists	The subscription can be YES (Y) or NO (N).	Mandatory
SubscriptionTopic	Represents a newsletter in SAP Marketing Cloud .	Mandatory The SubscriptionTopic property field must be passed, but can be left empty. If you want to create a newsletter subscription, you must specify the SubscriptionTopic.
SubscriptionTopicName	Name of the subscription topic.	
SubscriptionSourceObject	This field provides information on the source of the subscription, that is, where it came from. For example, the ID of a landing page.  This field can be filled with freetext.	
SubscriptionSourceObjectType	This field provides information on the source of the subscription and its type. For example, the business object name of a landing page.  This field can be filled with freetext.	
SubscriptionSourceSystem	This is the system that stores the subscription. For example, your local system ID.  This field can be filled with freetext.	

Property Name	Property Description	Usage
SubscriptionSourceSystemType	This is the type of system where the subscription is stored. For example, SAP_CEI.  This field can be filled with freetext.	
SubscriptionSourceCommMedium	Indicates where the subscription comes from, such as WEB, EMAIL, or PHONE. In case SubscriptionSourceCommMedium is not filled, this property is set to WEB.	Mandatory
SubscriptionSourceCommMedium-Name	Description of property <b>Subscription-SourceCommMedium</b>	Read-Only
IsConfirmationRequired	This is a boolean parameter. If the parameter is set to TRUE, the subscription is stored using the double opt-in or opt-out process.  If the property is not specified in the payload or it is set to FALSE the subscription is directly stored.	
LastChangedByUser	Name of the user who has changed the subscription last.	Read-Only
LastChangeDateTime	Date and time of the last permission change.	Read-Only
SubscriptionNoteText	A text to describe a subscription change.	

Parent topic: [Contacts \[page 412\]](#)

## Related Information

[Basic Concepts \[page 414\]](#)

[Payload Examples \[page 445\]](#)

[Function Imports \[page 464\]](#)



## 5.2.1.3 Payload Examples

Payload examples for API\_MKT\_CONTACT.

### i Note

- Before you start, please read the **Processing Info and Best Practices** section in [Basic Concepts \[page 414\]](#).
- Ensure that you include at least the mandatory request header fields in each payload and that you use the syntax as indicated in the examples for the different entries.

### Available Payload Examples

- [Contacts, Marketing Permissions, and Marketing Subscriptions \[page 445\]](#)
- [GET Requests \[page 453\]](#)
- [Account Team Members \[page 455\]](#)
- [Additional IDs \[page 456\]](#)
- [Contact Origin Data \[page 457\]](#)
- [Contact Relation Data \[page 459\]](#)
- [Contact Relation Additional IDs \[page 460\]](#)
- [Marketing Attributes \[page 461\]](#)
- [Marketing Areas \[page 463\]](#)

## Contacts, Marketing Permissions, and Marketing Subscriptions

### Create Contacts with Additional IDs

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT ContactOriginData(ContactID='4711',ContactOrigin='SAP_HYBRIS_CONSUMER')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "OriginDataLastChgUTCDateTime" : "2017-10-01T13:13:14Z",
  "CityName" : "Kiel",
  "Country" : "DE",
  "FirstName" : "Otto",
  "LastName" : "Normalverbraucher",
```

```

"FullName" : "Otto Normalverbraucher",
"BirthDate": "1961-10-28T00:00:00",
"GenderCode" : "1",
"AddressHouseNumber" : "1",
"IsConsumer" : false,
"IsContactPerson" : true,
"Language" : "DE",
"MaritalStatus" : "2",
"MaritalStatusName" : "Married",
"IsObsolete" : false,
"ContactPostalCode" : "24105",
"AddressRegion" : "01",
"StreetName" : "Hauptstrasse"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs (ContactID='4711',ContactOrigin='SAP_HYBRIS_CONSUMER',ContactAdditionalOrigin='EMAIL',ContactAdditionalID='otto.normalverbraucher@company.de')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs (ContactID='4711',ContactOrigin='SAP_HYBRIS_CONSUMER',ContactAdditionalOrigin='EMAIL',ContactAdditionalID='otto.normalverbraucher5@company.de')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes (ContactID='4711',ContactOrigin='SAP_HYBRIS_CONSUMER',MarketingAttributeCategory='HOBBY',MarketingAttributeValue='Soccer') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes (ContactID='4711',ContactOrigin='SAP_HYBRIS_CONSUMER',MarketingAttributeCategory='HOBBY',MarketingAttributeValue='Volleyball') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT

```

```

Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(ContactID='4711',ContactOrigin='SAP_HYBRIS_CONSUMER',MarketingAttributeCategory='Spoken_Language',MarketingAttributeValue='English')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(ContactID='4711',ContactOrigin='SAP_HYBRIS_CONSUMER',MarketingAttributeCategory='Spoken_Language',MarketingAttributeValue='Romanian')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT ContactOriginData(ContactID='4712',ContactOrigin='SAP_HYBRIS_CONSUMER')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-ForceSynchronousProcessing: X
Content-Type: application/json
{
  "OriginDataLastChgUTCDateTime" : "2017-10-01T13:13:14Z",
  "CityName" : "Walldorf",
  "Country" : "DE",
  "Department" : "",
  "FirstName" : "Erika",
  "LastName" : "Mustermann",
  "FullName" : "Erika Mustermann",
  "BirthDate": "1961-10-28T00:00:00",
  "Function" : " ",
  "GenderCode" : "2",
  "AddressHouseNumber" : "1",
  "Industry" : "",
  "IsConsumer" : true,
  "IsContactPerson" : false,
  "Language" : "DE",
  "MaritalStatus" : "1",
  "IsObsolete" : false,
  "ContactPostalCode" : "69190",
  "AddressRegion" : "08",
  "StreetName" : "Hauptstrasse"
}

```

```

--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs(ContactID='4712',ContactOrigin='SAP_HYBRIS_CONSUMER',ContactAdditionalOrigin='EMAIL',ContactAdditionalID='erika.mustermann4@privat.de')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs(ContactID='4712',ContactOrigin='SAP_HYBRIS_CONSUMER',ContactAdditionalOrigin='EMAIL',ContactAdditionalID='erika.mustermann5@privat.de')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Delete Contact

### i Note

A PUT request is executed to set the IsEndOfPurposeBlocked flag.

### Sample Code

```

--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT ContactOriginData(ContactID='AB20180612001-P',ContactOrigin='SAP_ERP_BUPA') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-07-23T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "IsEndOfPurposeBlocked": true,
  "OriginDataLastChgUTCDateTime": "2018-07-23T12:13:14"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Create Contacts with Marketing Permissions and Marketing Subscriptions

### i Note

The batch request is sent via http method POST containing PUT requests to create a new contact, marketing permission and marketing subscription. To update single attributes, you must use the PATCH request.

### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT ContactOriginData(ContactID='C98979992',ContactOrigin='SAP_C4C_BUPA')
HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-03-27T07:14:34'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "OriginDataLastChgUTCDateTime" : "2019-07-01T13:04:46.000",
  "CityName" : "Walldorf",
  "Country" : "DE",
  "FirstName" : "Max",
  "LastName" : "Mustermann",
  "FullName" : "Max Mustermann",
  "GenderCode" : "1",
  "AddressHouseNumber" : "99",
  "Language" : "DE",
  "MaritalStatus" : "2",
  "MaritalStatusName" : "Married",
  "ContactPostalCode" : "24105",
  "StreetName" : "Dietmar-Hopp-Allee"
}
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs(ContactID='C98979992',ContactOrigin='SAP_C4C_BUPA',ContactAdditionalOrigin='EMAIL',ContactAdditionalID='max.mustermann@mail.de') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.001'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
Content-Type: application/json
{
}
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingPermissions(ContactID='C98979992',ContactOrigin='SAP_C4C_BUPA',ContactPermissionID='max.mustermann@mail.de',ContactPermissionOrigin='EMAIL',MarketingArea='',CommunicationMedium='EMAIL') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.002'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
```

```

Content-Type: application/json
Content-Length: 1021
{
  "PermissionUTCDateTime" : "2019-07-01T13:04:46.002",
  "PermissionGranted" : "Y",
  "PermissionSourceCommMedium" : "WEB",
  "IsConfirmationRequired" : false,
  "PermissionNoteText" : "Sample Permission"
}
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingSubscriptions(ContactID='C98979992',ContactOrigin='SAP_C4C_BUPA',ContactSubscriptionID='max.mustermann@mail.de',ContactSubscriptionOrigin='EMAIL',CommunicationMedium='EMAIL',SubscriptionTopic='1') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.005",
  "SubscriptionSignUpExists" : "N",
  "SubscriptionSourceCommMedium" : "WEB",
  "IsConfirmationRequired" : false,
  "SubscriptionNoteText" : "Sample Subscription"
}
--changeset_01869434-0010-0001--
--batch--

```

## PATCH: Update Marketing Permissions and Marketing Subscriptions for a Contact

### Sample Code

```

--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingPermissions(ContactID='C98979992',ContactOrigin='SAP_C4C_BUPA',ContactPermissionID='max.mustermann@mail.de',ContactPermissionOrigin='EMAIL',MarketingArea='',CommunicationMedium='EMAIL') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.002'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "PermissionUTCDateTime" : "2019-07-01T13:04:46.003",
  "PermissionGranted" : "Y"
}
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingSubscriptions(ContactID='C98979992',ContactOrigin='SAP_C4C_BUPA',ContactSubscriptionID='max.mustermann@mail.de',ContactSubscriptionOrigin='EMAIL',CommunicationMedium='EMAIL',SubscriptionTopic='1') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'

```

```

Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
    {
        "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.006",
        "SubscriptionSignUpExists" : "N"
    }
--changeset_01869434-0010-0001--
--batch--

```

## PUT: Update or Create Marketing Permissions and Marketing Subscriptions for a Contact

### i Note

The sample code has a `PUT` request that updates marketing permissions and marketing subscriptions, or creates new marketing permissions and marketing subscriptions if they do not exist. To update single attributes, you must use the `PATCH` request. In addition, if the value of the property `IsConfirmationRequired` is set to true, a double opt-in is executed.

### Sample Code

```

--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingPermissions(ContactID='C98979992',ContactOrigin='SAP_C4C_BUPA',ContactPermissionID='max.mustermann@mail.de',ContactPermissionOrigin='EMAIL',MarketingArea='',CommunicationMedium='EMAIL') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.002'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
    {
        "PermissionUTCDateTime" : "2019-07-01T13:04:46.002",
        "PermissionGranted" : "Y",
        "PermissionSourceCommMedium" : "WEB",
        "IsConfirmationRequired" : false,
        "PermissionNoteText" : "Sample Permission"
    }
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingSubscriptions(ContactID='C98979992',ContactOrigin='SAP_C4C_BUPA',ContactSubscriptionID='max.mustermann@mail.de',ContactSubscriptionOrigin='EMAIL',CommunicationMedium='EMAIL',SubscriptionTopic='1') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
    {
        "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.005",
        "SubscriptionSignUpExists" : "N",
        "SubscriptionSourceCommMedium" : "WEB",

```

```

        "IsConfirmationRequired" : false,
        "SubscriptionNoteText" : "Sample Subscription"
    }
--changeset_01869434-0010-0001--
--batch--

```

## PUT: Update Additional IDs and their Permissions and Subscriptions Within One Changeset

Example Use Case:

1. A contact is created with opt-ins for email a@b.c and mobile +12345.
2. You want to change the email to d@e.f but retain the mobile number and the opt-ins for both.
3. To ensure that you do not lose the mobile opt-in, steps 4 and 5 must be in the same changeset.
4. To delete the email a@b.c, you use the Function Import.
5. You send all IDs, including the new email ID and the mobile ID.

### Sample Code

```

--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactOriginDeleteAdditionalIDs?
ContactID='98979992'&ContactOrigin='SAP_HYBRIS_CONSUMER' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs(ContactID='98979992',ContactOrigin='SAP_HYBRIS_CONSUMER',Contact
AdditionalOrigin='EMAIL',ContactAdditionalID='tobias.tester@company.de')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs(ContactID='98979992',ContactOrigin='SAP_HYBRIS_CONSUMER',Contact
AdditionalOrigin='EMAIL',ContactAdditionalID='peter.tester@company.de')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}

```



```

}
--changeset 761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingPermissions(ContactID='98979992',ContactOrigin='SAP_HYBRIS_CONSUMER',
ContactPermissionID='tobias.tester@company.de',ContactPermissionOrigin='EMAIL',
,MarketingArea='',CommunicationMedium='EMAIL') HTTP/1.1
Content-Length: 2035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "PermissionUTCDateTime" : "2018-11-02T09:19:12",
  "PermissionGranted" : "Y",
  "PermissionSourceCommMedium" : "WEB",
  "IsConfirmationRequired" : false
}
--changeset 761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingSubscriptions(ContactID='98979992',ContactOrigin='SAP_HYBRIS_CONSUMER',
,ContactSubscriptionID='peter.tester@company.de',ContactSubscriptionOrigin='E
MAIL',CommunicationMedium='EMAIL',SubscriptionTopic='') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "SubscriptionUTCDateTime" : "2018-11-02T09:19:12",
  "SubscriptionSignUpExists" : "Y",
  "SubscriptionSourceCommMedium" : "WEB"
}
--changeset 761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## GET Requests

### Get contact origin data for a specific contact from one origin

```

/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/
ContactOriginData(ContactOrigin='SAP_CRM_BUPA',ContactID='5320174712')

```

### Get additional IDs of a contact from a specific origin

#### i Note

\$filter is not supported for additional IDs.

```

/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/
ContactOriginData(ContactOrigin='SAP_CRM_BUPA',ContactID='5320174712')/
AdditionalIDs

```

### Get the first 500 contacts created on or after a given date

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/Contacts?$format=json&
$filter=CreationDateTime ge datetimeoffset'2018-10-01T00:00:00'&$top=500
```

### Get the first 500 contacts whose first name is Walter

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/Contacts?$format=json&
$filter=FirstName eq 'Walter'&$top=500
```

### Get the first five contacts related to a specific corporate account

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/Contacts?$format=json&
$filter=CorporateAccountUUID eq (guid'6c0b84b7-5523-1ed8-b1b8-34d75322d097')&$top=5
```

### Get all explicit marketing permissions for a specific ContactUUID

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/
ContactOriginData(ContactID='C98979992 ',ContactOrigin='SAP_C4C_BUPA')/
MarketingPermissions?$filter=PermissionIsImplicit eq false&$top=10
```

### Get all marketing permissions and marketing subscriptions for a contact with a certain ID and origin

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/
ContactOriginData(ContactID='C98979992 ',ContactOrigin='SAP_C4C_BUPA')?
$expand=MarketingPermissions,MarketingSubscriptions
```

### Get all marketing permissions and marketing subscriptions for a ContactUUID

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/
Contacts(ContactUUID=guid'6c0b84b7-5523-1ed9-a780-e4f6f36b1bfe')?
$expand=MarketingPermissions,MarketingSubscriptions
```

### Get contact data via ID and origin together with its marketing permissions and marketing subscriptions

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/
ContactOriginData(ContactID='C98979992 ',ContactOrigin='SAP_C4C_BUPA')/
MarketingPermissions
```

### Get all marketing subscriptions for a contact with a certain ID and origin

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/
ContactOriginData(ContactID='C98979992 ',ContactOrigin='SAP_C4C_BUPA')/
MarketingSubscriptions
```

### Get a contact via ContactUUID together with its marketing permissions and marketing subscriptions

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/MarketingSubscriptions?
$filter=ContactUUID eq guid'6c0b84b7-5523-1ed9-a780-e4f6f36b1bfe'&$top=20
```

### Get all marketing permissions for a specific email address of a contact

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/MarketingPermissions?
$filter=ContactPermissionID eq 'max.mustermann@mail.de' and ContactPermissionOrigin
eq 'EMAIL' &$top=20
```

## Get the first 500 contacts that subscribed to newsletter Fashion

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/MarketingSubscriptions?$top=500&
$filter=SubscriptionTopicName eq 'Fashion'
```

## Get the first 100 marketing permissions that are newer than a certain date and time

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0004/MarketingPermissions?$top=10&
$filter=PermissionUTCDateTime gt datetimeoffset'2019-01-01T00:00:00.001'
```

## Get the first 100 projections (relationship best record)

```
/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/Projections?$top=100
```

## Account Team Members

### PUT

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AccountTeamMembers(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',Tea
mMemberID='<TeamMemberID>',Role='<Role>') HTTP/1.1Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### DELETE

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
AccountTeamMembers(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',Tea
mMemberID='<TeamMemberID>',Role='<Role>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
```

```
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
AccountTeamMembers (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',TeamMemberID='<TeamMemberID>',Role='<Role>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "ContactID": "<ContactID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Additional IDs

## PUT

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',ContactAdditionalOrigin='<ContactAdditionalOrigin>',ContactAdditionalID='<ContactAdditionalID>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
AdditionalIDs (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',ContactAdditionalOrigin='<ContactAdditionalOrigin>',ContactAdditionalID='<ContactAdditionalID>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "ContactID": "<ContactID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Contact Origin Data

### PUT - Batch

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
ContactOriginData (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{"OriginDataLastChgUTCDateTime":"2017-10-01T13:13:14"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### PATCH - Batch

### Sample Code

```
--batch
```

```

Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ContactOriginData (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "OriginDataLastChgUTCDateTime": "2017-10-01T13:13:14",
  "AddressHouseNumber": "<AddressHouseNumber>",
  "ContactPostalCode": "<ContactPostalCode>",
  "StreetName": "<StreetName>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## PUT Single Entity (with ForceSync flag in Request Header)

### i Note

When you import single entities, the response body is empty. You can read the status of the import only in the **response header** in the attributes **Status** and **Sap-Message**.

### Sample Code

```

Request: PUT: /sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/
ContactOriginData (ContactID='C_20180828_00008',ContactOrigin='SAP_ERP_CONTACT'
)
{
  "OriginDataLastChgUTCDateTime" : "2017-10-01T13:13:14Z",
  "CityName" : "Kiel",
  "Country" : "DE",
  "FirstName" : "Otto",
  "LastName" : "Normalverbraucher",
  "FullName" : "Otto Normalverbraucher",
  "GenderCode" : "1",
  "AddressHouseNumber" : "1",
  "IsConsumer" : false,
  "IsContactPerson" : true,
  "Language" : "DE",
  "MaritalStatus" : "2",
  "MaritalStatusName" : "Married",
  "IsObsolete" : false,
  "ContactPostalCode" : "24105",
  "AddressRegion" : "01",
  "StreetName" : "Hauptstrasse",
}

```

## Contact Relation Data

### PATCH

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ContactRelationData(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',RelationshipCategory='<RelationshipCategory>',ReltdIntactnContactID='<ReltdIntactnContactID>',ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "ContactID": "<ContactID>"
  "RelationDataLastChgUTCDateTime": "2017-09-29T12:13:14"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### PUT

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
ContactRelationData(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',RelationshipCategory='<RelationshipCategory>',ReltdIntactnContactID='<ReltdIntactnContactID>',ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "RelationDataLastChgUTCDateTime": "2017-09-29T12:13:14"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## DELETE

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
ContactRelationData(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',Re
lationshipCategory='<RelationshipCategory>',ReltdIntactnContactID='<ReltdIntac
tnContactID>',ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Contact Relation Additional IDs

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ContactRelationAdditionalIDs(ContactID='<ContactID>',ContactOrigin='<ContactOr
igin>',RelationshipCategory='<RelationshipCategory>',ReltdIntactnContactID='<R
eltdIntactnContactID>',ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>'
,CntctRelationAdditionalID='<CntctRelationAdditionalID>',CntctRelationAddition
alOrigin='<CntctRelationAdditionalOrigin>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "ContactID": "<ContactID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```



## PUT

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
ContactRelationAdditionalIDs (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',RelationshipCategory='<RelationshipCategory>',ReltdIntactnContactID='<ReltdIntactnContactID>',ReltdIntactnContactOrigin='<ReltdIntactnContactOrigin>',CntctRelationAdditionalID='<CntctRelationAdditionalID>',CntctRelationAdditionalOrigin='<CntctRelationAdditionalOrigin>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## GET (Projections)

Get the first 100 projections

### Sample Code

```
/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0004/Projections?$top=100
```

## Marketing Attributes

## PUT

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',MarketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValue>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
```

```
}  
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--  
--batch--
```

## DELETE

### Sample Code

```
--batch  
Content-Type: multipart/mixed;  
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a  
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a  
content-type: application/http  
content-transfer-encoding: binary  
DELETE  
MarketingAttributes(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',Ma  
rketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValu  
e='<MarketingAttributeValue>') HTTP/1.1  
Content-Length: 1035  
Accept: application/json  
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'  
Sap-Cuan-SourceSystemType: EXT  
Sap-Cuan-SourceSystemId: HYBRIS  
Content-Type: application/json  
{  
}  
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--  
--batch--
```

## PATCH

### Sample Code

```
--batch  
Content-Type: multipart/mixed;  
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a  
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a  
content-type: application/http  
content-transfer-encoding: binary  
PATCH  
MarketingAttributes(ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',Ma  
rketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValu  
e='<MarketingAttributeValue>') HTTP/1.1  
Content-Length: 1035  
Accept: application/json  
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'  
Sap-Cuan-SourceSystemType: EXT  
Sap-Cuan-SourceSystemId: HYBRIS  
Sap-Cuan-SequenceId: UpdatePatch  
Content-Type: application/json  
{  
  "ContactID": "<ContactID>"  
}  
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--  
--batch--
```

## Marketing Areas

### PUT

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAreas (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### PATCH

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingAreas (ContactID='<ContactID>',ContactOrigin='<ContactOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "ContactID": "<ContactID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

Parent topic: [Contacts \[page 412\]](#)

## Related Information

[Basic Concepts \[page 414\]](#)

[Structure of OData Service API\\_MKT\\_CONTACTS \[page 418\]](#)

[Function Imports \[page 464\]](#)

[Payload Examples for Contact-to-Account Relationships](#)

### 5.2.1.4 Function Imports

Function imports are used to perform custom operations on an entity, which are typically not provided by OData operations. This section contains payload examples for the following function imports:

- [Set Main Contact \[page 464\]](#)
- [Delete Relationship Additional IDs \[page 465\]](#)
- [Delete Marketing Area \[page 465\]](#)
- [Delete All Marketing Areas from Origin \[page 466\]](#)
- [Delete Account Team Members \[page 467\]](#)
- [Delete Marketing Attribute \[page 467\]](#)
- [Delete Additional IDs \[page 468\]](#)
- [Delete Marketing Locations \[page 468\]](#)

### Set Main Contact

HTTP Method	Function Import
<i>POST</i>	ContactRelationDataSetMainContact  Flags (when <b>true</b> ) a contact relationship as the main contact for an account. Setting the value to <b>false</b> unflags a main contact.

### Payload Example

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactRelationDataSetMainContact?
ContactID='DEV_TEST'&ContactOrigin='SAP_ERP_CONTACT'&RelationshipCategory='BUR
001'&ReltdIntactnContactID='DEV_TEST'&ReltdIntactnContactOrigin='SAP_CRM_BUPA'
&IsMainContact=true HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
```

```
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Relationship Additional IDs

HTTP Method	Function Import
<i>POST</i>	ContactRelationDataDeleteAdditionalIDs  Deletes all additional IDs belonging to one contact relation data.

### Payload Example POST

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactRelationDataDeleteAdditionalIDs?
ContactID='DEV_TEST'&ContactOrigin='SAP_ERP_CONTACT'&RelationshipCategory='BUR
001'&ReltdIntactnContactID='DEV_TEST'&ReltdIntactnContactOrigin='SAP_CRM_BUPA'
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Marketing Area

HTTP Method	Function Import
<i>POST</i>	ContactDeleteMarketingArea  Deletes all occurrences of a marketing area from a contact.

## Payload Example

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactDeleteMarketingArea?
ContactID='DEV_TEST'&ContactOrigin='SAP_ERP_CONTACT'&InteractionContactMktgArea='GLOBAL' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete All Marketing Areas from Origin

HTTP Method	Function Import
<i>POST</i>	ContactOriginDeleteAllMktgAreas  Deletes all marketing areas from one origin.

## Payload Example

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactOriginDeleteAllMktgAreas?
ContactID='DEV_TEST'&ContactOrigin='SAP_ERP_CONTACT' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Account Team Members

HTTP Method	Function Import
<i>POST</i>	ContactDeleteAllAccountTeamMembers  Deletes all account team members for one contact.

### Payload Example

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactDeleteAllAccountTeamMembers?
ContactID='DEV_TEST'&ContactOrigin='SAP_ERP_CONTACT' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Marketing Attribute

HTTP Method	Function Import
<i>POST</i>	ContactOriginDeleteAllMktgAttributes  Deletes all marketing attributes from one origin.

### Payload Example

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactOriginDeleteAllMktgAttributes?
ContactID='DEV_TEST'&ContactOrigin='SAP_ERP_CONTACT' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
```

```
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Additional IDs

HTTP Method	Function Import
<i>POST</i>	ContactOriginDeleteAdditionalIDs Deletes all additional IDs from one origin except for IDs that come from the origin data.

### Payload Example

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactOriginDeleteAdditionalIDs?
ContactID='DEV_TEST'&ContactOrigin='SAP_ERP_CONTACT' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Marketing Locations

HTTP Method	Function Import
<i>POST</i>	ContactOriginDeleteAllMktgLocations Deletes all marketing locations from one origin.

### Payload Example

#### Sample Code

```
--batch
```



```

Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ContactOriginDeleteAllMktgLocations?
ContactID='DEV_TEST'&ContactOrigin='SAP_ERP_CONTACT' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Function Inset Parameters

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ContactID	ID of Contact	Edm. String	255	X	X
ContactOrigin	Origin of Contact	Edm. String	20	X	X

Parent topic: [Contacts \[page 412\]](#)

## Related Information

[Basic Concepts \[page 414\]](#)

[Structure of OData Service API\\_MKT\\_CONTACTS \[page 418\]](#)

[Payload Examples \[page 445\]](#)

## 5.2.2 Interaction Contacts

Public OData API (API\_MKT\_INTERACTION\_CONTACT\_SRV Version 3) for Interaction Contacts. Interaction Contact is a generic term to group all natural persons (contacts, consumers, or suspects), companies and “unknowns”, who interact with your company.

### i Note

We recommend that you use the current version 0003 of this service. If you want to use the previous version, you'll find the help links here:

- Version 0002: [Contact, Interaction Contact, Corporate Account API, Version 0002](#)

### i Note

This is a generic API Service. It should only be used in exceptional cases, as it has a limited subset of attributes, common to both natural persons and corporate contacts. Such an exceptional use case could be reading stored cookie IDs or reading "unknowns", that is, entities for whom it could not be determined whether they are natural persons or corporate accounts.

## Technical Data

### ⚠ Caution

The API services available in SAP Marketing Cloud must not be used for mass read (GET) operations. In other words, you cannot use them for extracting all available data, for example, to extract millions of contacts or interactions from your marketing system.

Name of the Service	API_MKT_INTERACTION_CONTACT
Authorizations	The following business catalog roles are required: <ul style="list-style-type: none"> <li>• For version 2: SAP_CEC_BC_MKT_API_IC2_PC</li> <li>• For version 3: SAP_CEC_BC_MKT_API_IC3_PC</li> </ul>
Communication Scenario ID	SAP_COM_0207
Component for Incidents	<ul style="list-style-type: none"> <li>• CEC-MKT-DM-IC (Interaction Contacts)</li> <li>• CEC-MKT-DM-PER (Permissions and Subscriptions)</li> </ul> <div data-bbox="821 1357 914 1397" data-label="Section-Header"> <h3>i Note</h3> </div> <div data-bbox="821 1413 1356 1482" data-label="Text"> <p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p> </div>
OData Version	2.0
Root URI	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003
Service Metadata URI:	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/\$metadata

Field Extensibility Supported

Yes



For more information, see the *Field Extensibility* section for marketing permissions in [Structure of API\\_MKT\\_INTERACTION\\_CONTACT](#) [page 475].

### i Note

You need to open the collapsible sections of the document first.

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Comment
<code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SR V;v=0003/\$metadata?sap- documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Interaction Contacts Details Page</a> 	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>1. On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li><li>2. Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Marketing - Interaction Contacts API</a> 	General access link takes you directly to the <i>Contacts</i> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

[Basic Concepts](#) [page 472]

Public OData API (`API_MKT_INTERACTION_CONTACT_SRV`) for Interaction Contacts. Interaction Contact is a generic term to group all natural persons (contacts, consumers, or suspects), companies and “unknowns”, who interact with your company.

[Structure of API\\_MKT\\_INTERACTION\\_CONTACT \[page 475\]](#)

This document describes the structure of the **Public OData API** `API_MKT_INTERACTION_CONTACT`.

[Payload Examples for Interaction Contacts \[page 495\]](#)

Payload examples for `API_MKT_INTERACTION_CONTACT`.

[Function Imports \[page 508\]](#)

Function imports are used to perform custom operations on an entity, which are typically not provided by OData operations. This section contains payload examples for the following function imports:

## 5.2.2.1 Basic Concepts

Public OData API (`API_MKT_INTERACTION_CONTACT_SRV`) for Interaction Contacts. Interaction Contact is a generic term to group all natural persons (contacts, consumers, or suspects), companies and “unknowns”, who interact with your company.

### Overview

The public API for Interaction Contact supports operations on the `Interaction Contact Business Object` and the `Marketing Permissions Business Object`.

#### i Note


There is no separate public OData API for **marketing permissions**. The corresponding entity is part of this service since marketing permissions are always stored for a certain interaction contact.

### Processing Info and Best Practices

#### i Note

For generally applicable recommendations and best practices, make sure you refer to the section [Best Practices and Recommended Package Sizes \[page 400\]](#).

- The minimum data required when importing interaction contacts is an ID, an ID Origin, a timestamp, and **at least one other attribute**.
- **When to use `PUT` and `PATCH`:**
  - `PUT` requests are most suitable for an initial data import, for example, when you want to create a new contact. A `PUT` request requires that you always send all properties. Any properties that you omit are overwritten by blank entries. That is, any existing entries are deleted. If no record is found, a new record is created. In other words, the `PUT` request functions as a full **upsert**

- We recommend that you use `PATCH` requests for all other imports. A `PATCH` request updates only the properties provided in the request body and leaves everything untouched that was not provided. So, you can omit all properties that are not to be changed. Like the `PUT` request, if no record is found, a new record is created with the available properties.. In other words, the `PATCH` request functions as a delta **upsert**.
- An additional advantage of using `PATCH` is that you specify your own sequence ID. For this reason, it is more flexible than a `PUT` operation, where the sequence ID is set by default and cannot be changed.
- Basically, since you can use `PATCH` with the same payload as you would use for `PUT`, the `PATCH` operation is more universal and you can work with it exclusively.
- We recommend that you don't mix `PUT` and `PATCH` operations. Doing so can lead to unwanted results since a `PUT` operation is processed before a `PATCH`.
- **Do not combine a DELETE operation with other OData operations in one changeset.**  
We recommend that you do not combine the OData operations `PUT`, `PATCH`, `POST`, with a `DELETE` operation in the same changeset. For example, let's say you want to update data for Contact A by adding an additional email address and at the same time delete a mobile number that is no longer valid. So, you send a `PUT` operation on the `AdditionalId` entity with the new email address and a `DELETE` operation within the same changeset. One of these operations could cancel out the other and the resulting dataset will not be as intended.  
**Recommended Practice:** For such combined operations including a `DELETE` operation, we recommend that you always use the relevant function import, which allows deletion of specific entities, together with the appropriate OData operation `PUT`, `PATCH`, or `POST` within the same changeset.
- Batch requests are submitted as a single `HTTP POST` request to the `$batch` endpoint of a service as described in [OData-URI ]. The batch request must contain a header parameter `content-type`, specifying the value `multipart/mixed` and `boundary=batch`.
- Do not mix different types of services for operations involving the same data source. For example, when importing contacts from a web shop, do not use the `CUAN_IMPORT` service for a `PUT` operation and then the `API_MKT_INTERACTION_CONTACT` service to `PATCH` contacts. You can, however, migrate from `CUAN_IMPORT` to the `API*` services.
- The origin that you pass via the property `ContactOrigin` cannot be shareable. If the **main** origin is set to `Shareable`, this will trigger an error. For more information, see [Configuring Origins](#). You can view sample payloads and test the API at [!\[\]\(94dfacbf937cdd7da4837a6fcd8fc785\_img.jpg\)](https://api.sap.com/api/API_MKT_INTERACTION_CONTACT_SRV_0003/resource).
- **SAP internal codes:** If you are not thoroughly familiar with the internal codes used by SAP for the following entities, you should use a free text version of these instead to avoid errors during import: Ad Network, Country, Customer Industry Code, Department, Device Type, Function, Gender, Language, Marital Status, Region, and Title. You should then map your free text name to the SAP internal code in the [Map Free Text](#) app. For more information, see [Map Free Texts](#).

### Note

The UTC timestamp of permissions can't lie in the future.

When you import permissions, they must not have a timestamp that lies in the future. The timestamp of imported permissions is always in UTC. The field name in the OData service is called `PermissionUTCDateTime`.

If you want to use your local timestamp, you have to add the time zone information, that is, your local time zone together with the time zone offset or enter a timestamp that is converted to UTC.

The date and time information is adapted by the standard time difference (offset) with +01:00 for Central European Time (CET) or -05:00 for Eastern Standard Time (EST). For example:  
2019-01-01T12:00:00+01:00

If you live east of UTC and enter your timestamp in your local time zone without time zone offset, this will result in a future timestamp. For example, you live in Germany and your local time is 8 a.m on November, 28. If you enter this as the UTC timestamp without a time zone offset, the UTC permission timestamp will show as 8 a.m., November 28, while in the UTC time zone it's 7 a.m., November 28. You've created a UTC permission timestamp that lies in the future and is invalid.

## Error Messages

If the OData service is not accessible, for example due to missing authorization, or because the system is not available, a corresponding HTTP status code is returned.

If the OData service is accepted by the gateway component in the SAP Marketing Cloud system, the HTTP status code 201 or 204 is returned. Any processing errors are recorded in the SAP Marketing Cloud system in the *Import Monitor* app, where they can be monitored, restarted and discarded.

By default, data processing for contacts, interaction contacts, corporate accounts, or marketing permissions is asynchronous. In most cases an OK response, such as a receipt notification, is returned almost immediately. An exception to this would be data uploads that might contain severe errors, such as parse or format errors, and so would not return an OK response but an error message. The data you upload lands in a staging area, where it is then further processed. You can change the default setting to synchronous processing by setting the property Sap-Cuan-ForceSynchronousProcessing to True. In this case, any error messages are returned as soon as they are detected.

To view the processing status and to check for errors or success messages, you must launch the *Import Monitor* app. Messages for marketing permissions in this app are displayed under the API for Contact, API for Interaction Contacts, or API for Corporate Accounts depending on the API OData service you use. In the event of errors, you can restart or discard the import in the *Import Monitor*.

**Parent topic:** [Interaction Contacts \[page 469\]](#)

## Related Information

[Structure of API\\_MKT\\_INTERACTION\\_CONTACT \[page 475\]](#)

[Payload Examples for Interaction Contacts \[page 495\]](#)

[Function Imports \[page 508\]](#)

## 5.2.2.2 Structure of API\_MKT\_INTERACTION\_CONTACT

This document describes the structure of the **Public OData API** `API_MKT_INTERACTION_CONTACT`.

Make sure you read these topics before you start:

- [Best Practices and Recommended Package Sizes \[page 400\]](#)
- [Basic Concepts \[page 472\]](#)

### Request Header

The request header contains the additional header fields listed in the table. Remember to include at least the mandatory request header fields in each payload.

Property	Example	Description	Max. Length	Mandatory
Sap-Cuan-RequestTimestamp	'2017-09-28T12:13:14'	Timestamp of the import run in this format.		X
Sap-Cuan-SequenceId	PatchAddress	This defines a set of fields that are to be updated, for example, address fields, which can be interpreted as a field group. The combination of the header fields Sap-Cuan-SequenceId and Sap-Cuan-RequestTimestamp is used to check the sequence of the data received. If the data that is received has a timestamp older than already imported data, it is ignored.		X (only mandatory for Patch Mode)
Sap-Cuan-SourceSystemType	EXT	Type of source system. This is a free text field.	20	
Sap-Cuan-SourceSystemId	HYBRIS	Identifier of source system. This is a free text field.	255	

Property	Example	Description	Max. Length	Mandatory
Sap-Cuan-ForceSynchronousProcessing	X	<p>This flag is deselected by default, which means that uploaded data is processed asynchronously. On upload, a success message is output immediately, unless there are errors such as authorization issues or bad requests. Objects are uploaded to the staging area and processed successively from there. All status messages can be displayed in the <a href="#">Import Monitor</a> app.</p> <p>You can force imports to be processed <b>synchronously</b> by setting this flag. In this case, an error message will be returned as soon as an error is detected. Such error messages are output in the <a href="#">Import Monitor</a> app</p>		
Sap-Cuan-ReferenceId	345g67980907	External reference of the inbound message	32	

## Entity Sets

The Interaction Contact OData API provides the following entity sets:

Entity Set	Description	Path
InteractionContacts	This entity contains all interaction contact information from the root.	/InteractionContacts
AccountTeamMembers	This entity contains information about the account team members.	/AccountTeamMembers
AdditionalIDs	This entity contains information about additional IDs.	/AdditionalIDs



Entity Set	Description	Path
InteractionContactOriginData	This entity contains interaction contact origin data. <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <p>The property <code>OriginDataLastChgUTCDateTime</code> is mandatory and must be specified.</p> </div>	/InteractionContactOriginData
MarketingAttributes	This entity contains information about marketing attributes.	/MarketingAttributes
MarketingAreas	This entity contains information about marketing areas.	/MarketingAreas
MarketingPermissions	This entity contains information about marketing permissions.	/MarketingPermissions
MarketingSubscriptions	This entity contains information about marketing subscriptions.	/MarketingSubscriptions

## InteractionContacts

**GET: Entity Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/InteractionContacts`

You can perform the following operations on the `InteractionContacts` entity set:

HTTP Method	Description	Path
<a href="#">GET</a>	Get a list of interaction contacts. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> . <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>A maximum of 5000 interaction contacts can be fetched in a single request</li> <li>Specification of TOP is mandatory.</li> </ul> </div>	/InteractionContacts?\$top=1
	Get the details of a specific contact using the <code>InteractionContactUUID</code> .	/InteractionContacts(guid '<InteractionContactUUID>')

## AccountTeamMembers

You can perform the following operations on the `AccountTeamMember` entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/AccountTeamMembers`
- **PUT, PATCH, or DELETE in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/$batch`
- **PUT, PATCH, or DELETE in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/ AccountTeamMembers (InteractionContactID='<InteractionContactID>', InteractionContactOrigin='<InteractionContactOrigin>', TeamMemberID='<TeamMemberID>', Role='<Role>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of account team members.	<code>/AccountTeamMembers?\$top=1</code>  This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> .
		<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 account team members can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>
	Get the details of a specific account team member.	This operation is not supported.
<i>POST (Batch)</i>	Update or create an account team member in batch mode	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/\$batch</code>
	Delete account team member in batch mode	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/\$batch</code>
	Append one new account team member	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/\$batch</code>

HTTP Method	Description	Path
<i>PUT</i>	Update or create an account team member.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/ AccountTeamMembers (InteractionContactID='&lt;Interac tionContactID&gt;', InteractionCont actOrigin='&lt;InteractionContactO rigin&gt;', TeamMemberID='&lt;TeamMemb erID&gt;', Role='&lt;Role&gt;')</pre>
<i>PATCH</i>	Add one new account team member.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/ AccountTeamMembers (InteractionContactID='&lt;Interac tionContactID&gt;', InteractionCont actOrigin='&lt;InteractionContactO rigin&gt;', TeamMemberID='&lt;TeamMemb erID&gt;', Role='&lt;Role&gt;')</pre>
<i>DELETE</i>	Delete an account team member.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/ AccountTeamMembers (InteractionContactID='&lt;Interac tionContactID&gt;', InteractionCont actOrigin='&lt;InteractionContactO rigin&gt;', TeamMemberID='&lt;TeamMemb erID&gt;', Role='&lt;Role&gt;')</pre>

## AdditionalIDs

You can perform the following operations on the AdditionalIDs entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/AdditionalIDs`
- **PUT, PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/AdditionalIDs(InteractionContactID='<InteractionContactID>', InteractionContactOrigin='<InteractionContactOrigin>', InteractionContactAdditionalOrigin='<InteractionContactAdditionalOrigin>', InteractionContactAdditionalExternalID='<InteractionContactAdditionalExternalID>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of additional IDs by Interaction Contact ID and ID Origin.	<p><code>/AdditionalIDs?\$top=1</code></p> <p>This method supports standard OData parameters such as <code>\$select</code>, <code>\$top</code>, <code>\$skip</code>, <code>\$count</code>, <code>\$inlinecount</code>, and <code>\$orderby</code></p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 additional IDs can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> <li>• <code>\$filter</code> is not supported for additional IDs.</li> </ul> </div>
	Get the details of a specific additional ID.	<p><code>/AdditionalIDs('&lt;InteractionContactID&gt;,&lt;InteractionContactOrigin&gt;,&lt;InteractionContactAdditionalOrigin&gt;,&lt;InteractionContactAdditionalExternalID&gt;')</code></p>
<i>POST (Batch)</i>	Update or create an additional ID in batch mode	
	Add one new Additional ID	<p><code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/\$batch</code></p>
<i>PUT</i>	Update or create an additional ID.	<p><code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/AdditionalIDs(InteractionContactID='&lt;InteractionContactID&gt;', InteractionContactOrigin='&lt;InteractionContactOrigin&gt;', InteractionContactAdditionalOrigin='&lt;InteractionContactAdditionalOrigin&gt;', InteractionContactAdditionalExternalID='&lt;InteractionContactAdditionalExternalID&gt;')</code></p>

HTTP Method	Description	Path
<i>PATCH</i>	Add one new additional ID.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/ AdditionalIDs (InteractionContac tID='&lt;InteractionContactID&gt;', In teractionContactOrigin='&lt;Intera ctionContactOrigin&gt;', Interactio nContactAdditionalOrigin='&lt;Inte ractionContactAdditionalOrigin&gt; ', InteractionContactAdditionalE xternalID='&lt;InteractionContactA dditionalExternalID&gt;')</pre>

## InteractionContactOriginData

You can perform the following operations on the InteractionContactOriginData entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/InteractionContactOriginData`
- **PUT, PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/InteractionContactOriginData (InteractionContactID='<InteractionContactID>', InteractionContactOrigin='<InteractionContactOrigin>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of Interaction Contact Origin Data.	<pre>/InteractionContactOriginData? \$top=1</pre> <p>This method supports standard OData parameters such as <code>\$filter</code>, <code>\$select</code>, <code>\$top</code>, <code>\$skip</code>, <code>\$count</code>, <code>\$inlinecount</code>, and <code>\$orderby</code></p>

### i Note

- A maximum of 5000 interaction contact origin data entities can be fetched in a single request
- Specification of TOP is mandatory.

HTTP Method	Description	Path
	Get the details of specific interaction contact origin data.	/ InteractionContactOriginData ('<InteractionContactID>,<InteractionContactOrigin>')
<i>POST (Batch)</i>	<p><b>i Note</b></p> <p>The property <code>OriginDataLastChgUTCDateTime</code> is mandatory and must be specified.</p>	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/\$batch
	Delta update: PATCH attributes of the entity <code>InteractionContactOriginDataUpdate</code> or create interaction contact origin data in batch mode; (creates a contact if the contact not exist)	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/\$batch
<i>PUT</i>	Update or create interaction contact origin data. This creates an interaction contact if the contact not exist.	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/ InteractionContactOriginData (InteractionContactID='<InteractionContactID>', InteractionContactOrigin='<InteractionContactOrigin>')
	<p><b>i Note</b></p> <p>The property <code>OriginDataLastChgUTCDateTime</code> is mandatory and must be specified.</p>	
<i>PATCH</i>	Delta Update: PATCH attributes of the entity <code>InteractionContactOriginData</code> .	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/ InteractionContactOriginData (InteractionContactID='<InteractionContactID>', InteractionContactOrigin='<InteractionContactOrigin>')

## MarketingAttributes

You can perform the following operations on the `MarketingAttributes` entity set:

- **GET:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_INTERACTION\_CONTACT\_SRV;v=0003/MarketingAttributes
- **PUT, PATCH, or DELETE in batch:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_INTERACTION\_CONTACT\_SRV;v=0003/\$batch

- **PUT, PATCH, or DELETE in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/MarketingAttributes(InteractionContactID='<InteractionContactID>',InteractionContactOrigin='<InteractionContactOrigin>',MarketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValue>')`

HTTP Method	Description	Path
<i>GET</i>	Update or create interaction contact origin data in batch mode;Get a list of marketing attributes by Interaction Contact ID and Origin.	/MarketingAttributes?\$stop=1  This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby
	Get the details of a specific marketing attribute.	/ MarketingAttributes ('<InteractionContactID>,<InteractionContactOrigin>,<MarketingAttributeCategory>,<MarketingAttributeValue>')
<i>POST (Batch)</i>	Update or create marketing attributes in batch mode	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/\$batch</code>
	Append one new marketing attribute	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/\$batch</code>
<i>PUT</i>	Update or create marketing attributes.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/MarketingAttributes(InteractionContactID='&lt;ContactID&gt;',ContactOrigin='&lt;ContactOrigin&gt;',MarketingAttributeCategory='&lt;MarketingAttributeCategory&gt;',MarketingAttributeValue='&lt;MarketingAttributeValue&gt;')</code>

**i Note**

- A maximum of 5000 marketing attributes can be fetched in a single request
- Specification of TOP is mandatory.

HTTP Method	Description	Path
<i>PATCH</i>	Add one new marketing attribute.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/MarketingAttributes(InteractionContactID=' <InteractionContactID>',InteractionContactOrigin=' <InteractionContactOrigin>',MarketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValue>')
<i>DELETE</i>	Delete marketing attributes.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/MarketingAttributes(InteractionContactID=' <InteractionContactID>',InteractionContactOrigin=' <InteractionContactOrigin>',MarketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValue>')

## Marketing Areas

You can perform the following operations on the `MarketingAreas` entity set:

- **GET:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_INTERACTION\_CONTACT\_SRV;v=0003
- **PUT, PATCH in batch:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_INTERACTION\_CONTACT\_SRV;v=0003/\$batch
- **PUT, PATCH in a single operation:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_INTERACTION\_CONTACT\_SRV;v=0003/MarketingAreas(InteractionContactID='<InteractionContactID>',InteractionContactOrigin='<InteractionContactOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>')



HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing areas by Interaction Contact ID and Origin.	<pre>/sap/opu/odata/SAP/ API_MKT_CONTACT_SRV;v=0003/ Contacts? \$expand=MarketingAreas&amp;\$top=2</pre> <p>This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 marketing areas can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>
	Get the details of a specific marketing attribute.	<pre>/ MarketingAreas ('&lt;InteractionCon tactID&gt;,&lt;InteractionContactOrig in&gt;,&lt;InteractionContactMktgArea &gt;,' )</pre>
<i>POST (Batch)</i>	Update or create marketing areas in batch mode	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/\$batch</pre>
	Append one new marketing area	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/\$batch</pre>
<i>PUT</i>	Update or create marketing areas.	<pre>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/API_MKT_INTERAC- TION_CONTACT_SRV;v=0003/MarketingAr- eas(InteractionContactID='&lt;InteractionCon- tactID&gt;',InteractionContactOrigin='&lt;Interac- tionContactOrigin&gt;',InteractionContactMktg- Area='&lt;InteractionContactMktgArea&gt;')</pre>

HTTP Method	Description	Path
<i>PATCH</i>	Add one new marketing area.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_INTERACTION_CONTACT_SRV ;v=0003/ MarketingAreas(InteractionConta ctID='&lt;InteractionContactID&gt;', I nteractionContactOrigin='&lt;Inter actionContactOrigin&gt;', Interacti onContactMktgArea='&lt;Interaction ContactMktgArea&gt;')</pre>

## MarketingPermissions

**Entity Path:** /MarketingPermissions

**Field Extensibility:** The following business context is relevant: *Marketing: Marketing Permissions*. Custom fields for business object MKT\_PERMISSION (Marketing: Permission) are only supported if you use **version 3** of the API\_MKT\_INTERACTION\_CONTACT service.

### i Note

Please note the following:

- For all HTTP operations both \$batch requests and single requests can be used.
- Interactions are assigned when marketing permissions are created or updated to allow for analysis of interaction contacts.

You can perform the following operations on the MarketingPermissions entity set:

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing permissions by Interaction Contact ID and ID Origin. This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby	/ MarketingPermissions?\$top=1

### i Note

- A maximum of 5000 marketing permissions can be fetched in a single request
- Specification of TOP is mandatory.

HTTP Method	Description	Path
<i>PATCH</i>	Update or create marketing permissions. This creates a marketing permission if the permission does not exist.	<code>/MarketingPermissions(InteractionContactID='&lt;InteractionContactID&gt;', InteractionContactOrigin='&lt;InteractionContactOrigin&gt;', InteractionContactPermissionID='&lt;InteractionContactPermissionID&gt;', InteractionContactPermissionOrigin='&lt;InteractionContactPermissionOrigin&gt;', MarketingArea='&lt;MarketingArea&gt;', CommunicationMedium='&lt;CommunicationMedium&gt;')</code>
<i>PUT</i>	Update or create marketing permissions. This creates a marketing permission if the permission does not exist.  Delta Update of PATCH attributes of the entity MarketingPermission.	<code>/MarketingPermissions(InteractionContactID='&lt;InteractionContactID&gt;', InteractionContactOrigin='&lt;InteractionContactOrigin&gt;', InteractionContactPermissionID='&lt;InteractionContactPermissionID&gt;', InteractionContactPermissionOrigin='&lt;InteractionContactPermissionOrigin&gt;', MarketingArea='&lt;MarketingArea&gt;', CommunicationMedium='&lt;CommunicationMedium&gt;')</code>

The table below describes the properties for the entity `MarketingPermissions`.

## MarketingPermissions Property Names and Descriptions

Property Name	Property Description	Usage
InteractionContactID	<p>The InteractionContactID and InteractionContactOrigin identify the contact uniquely.</p> <p><b>Example:</b> a business partner ID from the CRM system</p>	Mandatory
InteractionContactOrigin	<p>The InteractionContactID and InteractionContactOrigin identify the contact uniquely.</p> <p><b>Example:</b> SAP_CRM_BUPA</p>	Mandatory
InteractionContactPermissionID	<p>The InteractionContactPermissionID and IntactnCntctPermissionOrigin store marketing permissions.</p> <p><b>Example:</b> first.lastname@mail.de</p>	Mandatory
IntactnCntctPermissionOrigin	<p>The InteractionContactPermissionID and IntactnCntctPermissionOrigin store marketing permissions.</p> <p>IntactnCntctPermissionOrigin is the origin of the interaction contact ID that stores marketing permissions.</p> <p>The origin indicates the source of an ID. By defining the origin, you determine that an interaction contact with an ID associated to a source is eligible to be analyzed.</p> <p>You can configure origins of contacts IDs in the <a href="#">Configuring Origins</a> configuration app.</p> <p><b>Example:</b> EMAIL</p>	Mandatory
IntactnCntctPrmssnOriginName	<p>Description of property <b>IntactnCntctPermissionOrigin</b></p>	Read-Only
MarketingArea	<p>Identifies an area of responsibility or an organizational unit.</p> <p>You use a marketing area to restrict access to instances of an object, such as campaign, email message, email template, target group, or permission.</p>	<p>Mandatory</p> <p>The MarketingArea property field must be passed, but can be left empty.</p>
MarketingAreaName	<p>Description of property <b>MarketingArea</b>.</p>	

Property Name	Property Description	Usage
CommunicationMedium	Represents the type of permission, for example, <b>EMAIL</b> or <b>PHONE</b> .  You can configure communication media in the <a href="#">Managing Interaction Content</a> configuration app.	Mandatory
CommunicationMediumName	Description of property <b>CommunicationMedium</b>	
InteractionContactUUID	Unique ID of an interaction contact in SAP Marketing Cloud .	Read-Only
PermissionUTCDateTime	This is the timestamp for when the permission was given or removed.  <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <p>The time stamp must not be initial or null.</p> </div>	Mandatory
PermissionUUID	Unique ID of a permission in SAP Marketing Cloud .	
PermissionGranted	The permission can be YES ( <b>Y</b> ) or NO ( <b>N</b> ).	Mandatory
PermissionSourceObject	This field provides information on the source of the permission, that is, where it came from. For example, the ID of a landing page.  This field can be filled with freetext.	If you enter a value for the PermissionSourceObject property, you must also specify a value for the PermissionSourceObjectType.  Both fields must be filled or left empty.
PermissionSourceObjectType	This field provides information on the source of the permission and its type. For example, the business object name of a landing page.  This field can be filled with freetext.	
PermissionSourceSystem	This is the system that stores the permission. For example, ABD client 100.  This field can be filled with freetext.	If you enter a value for the PermissionSourceSystem property, you must also specify a value for the PermissionSourceSystemType.

Property Name	Property Description	Usage
PermissionSourceSystemType	This is the type of system where the permission is stored. For example, SAP_CEI.  This field can be filled with freetext.	
PermissionSourceCommMedium	Indicates where the permission comes from, such as WEB, EMAIL, or PHONE. In case PermissionSourceCommMedium is not filled, this property is set to WEB.	
PermissionSourceCommMediumName	Description of property <b>PermissionSourceCommMedium</b>	Read-Only
PermissionIsImplicit	If the system sets this field to TRUE, then it is an implicit permission, which is determined by country-specific regulation.  If the system sets this field to FALSE, the contact has given this permission explicitly.	Read-Only
PermissionNoteText	A text to describe a permission change.	
IsConfirmationRequired	This is a boolean parameter. If the parameter is set to TRUE, the permission is stored using the double opt-in or opt-out process.  If the property is not specified in the payload or it is set to FALSE the permission is directly stored.	
LastChangedByUser	Name of the user who has changed the permissions last.	Read-Only
LastChangeDateTime	Date and time of the last permission change.	Read-Only

## MarketingSubscriptions

Entity Path: /MarketingSubscriptions

### i Note

- For all HTTP operations both \$batch requests and single requests can be used.

- Interactions are assigned when marketing permissions are created or updated to allow for analysis of contacts.

You can perform the following operations on the `MarketingSubscriptions` entity set:

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing subscriptions by Contact ID and ID Origin. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> .	<code>/MarketingSubscriptions?\$top=1</code>
	<p><b>Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 marketing subscriptions can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul>	
<i>PATCH</i>	Update or create subscriptions. This creates a subscription if the subscription does not exist.	<code>/MarketingSubscriptions(ContactID='&lt;ContactID&gt;',ContactOrigin='&lt;ContactOrigin&gt;',ContactPermissionID='&lt;ContactPermissionID&gt;',ContactPermissionOrigin='&lt;ContactPermissionOrigin&gt;',CommunicationMedium='&lt;CommunicationMedium&gt;',SubscriptionTopic='&lt;SubscriptionTopic&gt;')</code>
<i>PUT</i>	Update or create subscriptions. This creates a subscription if the subscription does not exist.  Delta Update of PATCH attributes of the entity <code>MarketingSubscriptions</code> .	<code>/MarketingSubscriptions(ContactID='&lt;ContactID&gt;',ContactOrigin='&lt;ContactOrigin&gt;',ContactPermissionID='&lt;ContactPermissionID&gt;',ContactPermissionOrigin='&lt;ContactPermissionOrigin&gt;',CommunicationMedium='&lt;CommunicationMedium&gt;',SubscriptionTopic='&lt;SubscriptionTopic&gt;')</code>

## Marketing Subscription Property Descriptions

The table describes the properties for the `MarketingSubscription` entity.

## MarketingSubscription Property Names and Descriptions

Property Name	Property Description	Usage
InteractionContactID	<p>The InteractionContactID and InteractionContactOrigin identify the contact uniquely.</p> <p><b>Example:</b> a business partner ID from the CRM system.</p>	
InteractionContactOrigin	<p>The InteractionContactID and InteractionContactOrigin identify the contact uniquely.</p> <p>The InteractionContactID will not be saved to the MarketingSubscription but is only used to derive a unique InteractionContactUUID. This data will not be returned in GET requests.</p> <p><b>Example:</b> SAP_CRM_BUPA</p>	
IntactnCntctSubscriptionID	<p>The InteractionContactPermissionID and InteractionContactSubscriptionOrigin store marketing subscription.</p>	Mandatory
IntactnCntctSubscriptionOrigin	<p>The InteractionContactSubscriptionID and InteractionContactSubscriptionOrigin store marketing subscriptions.</p> <p>InteractionContactSubscriptionOrigin is the origin of an interaction contact ID that stores marketing subscriptions. The origin indicates the source of an ID. By defining the origin, you determine that an interaction contact with an ID associated to a source can be analyzed.</p> <p><b>Example:</b> EMAIL</p> <p>You can configure origins of contact IDs in the <a href="#">Configuring Origins</a> configuration app.</p>	Mandatory
InteractionContactSubscriptionOrigin-Name	<p>Description of property <b>InteractionContactSubscriptionOrigin</b></p>	Read-Only
CommunicationMedium	<p>Represents the type of subscription, for example, <b>EMAIL</b> or <b>PHONE</b>.</p> <p>You can configure communication media in the <a href="#">Managing Interaction Content</a> configuration app.</p>	Mandatory



Property Name	Property Description	Usage
CommunicationMediumName	Description of property <b>CommunicationMedium</b>	Read-Only
InteractionContactUUID	Unique ID of an interaction contact in SAP Marketing Cloud .	Read-Only The field value is returned internally.
SubscriptionUUID	Unique ID of a subscription in SAP Marketing Cloud .	
SubscriptionUTCDateTime	This is the timestamp for when the subscription was given or removed.	Mandatory
	<div style="border-left: 2px solid #0070C0; padding-left: 10px; background-color: #F0F0F0;"> <p><b>i Note</b></p> <p>The time stamp must not be initial or null.</p> </div>	
SubscriptionSignUpExists	The subscription can be YES (Y) or NO (N).	Mandatory
SubscriptionTopic	Represents a newsletter in SAP Marketing Cloud .	Mandatory The SubscriptionTopic property field must be passed, but can be left empty. If you want to create a newsletter subscription, you must specify the SubscriptionTopic.
SubscriptionTopicName	Name of the subscription topic.	
SubscriptionSourceObject	This field provides information on the source of the subscription, that is, where it came from. For example, the ID of a landing page.  This field can be filled with freetext.	
SubscriptionSourceObjectType	This field provides information on the source of the subscription and its type. For example, the business object name of a landing page.  This field can be filled with freetext.	
SubscriptionSourceSystem	This is the system that stores the subscription. For example, your local system ID.  This field can be filled with freetext.	

Property Name	Property Description	Usage
SubscriptionSourceType	This is the type of system where the subscription is stored. For example, SAP_CEI.  This field can be filled with freetext.	
SubscriptionSourceCommMedium	Indicates where the subscription comes from, such as WEB, EMAIL, or PHONE. In case SubscriptionSourceCommMedium is not filled, this property is set to WEB.	
SubscriptionSourceCommMedium-Name	Description of property <b>Subscription-SourceCommMedium</b>	
IsConfirmationRequired	This is a boolean parameter. If the parameter is set to TRUE, the subscription is stored using the double opt-in or opt-out process.  If the property is not specified in the payload or it is set to FALSE the subscription is directly stored.	
LastChangedByUser	Name of the user who has changed the subscription last.	Read-Only
LastChangeDateTime	Date and time of the last permission change.	Read-Only
SubscriptionNoteText	A text to describe a subscription change.	

Parent topic: [Interaction Contacts \[page 469\]](#)

## Related Information

[Basic Concepts \[page 472\]](#)

[Payload Examples for Interaction Contacts \[page 495\]](#)

[Function Imports \[page 508\]](#)

## 5.2.2.3 Payload Examples for Interaction Contacts

Payload examples for API\_MKT\_INTERACTION\_CONTACT.

### i Note

- Before you start, please read the **Processing Info and Best Practices** section in [Basic Concepts \[page 472\]](#).
- Remember to include at least the mandatory request header fields in each payload.

### Available Payload Examples

- [Interaction Contacts, Marketing Permissions, and Marketing Subscriptions \[page 495\]](#)
- [GET Requests \[page 501\]](#)
- [Account Team Members \[page 502\]](#)
- [Additional IDs \[page 503\]](#)
- [Interaction Contact Origin Data \[page 504\]](#)
- [Marketing Attributes \[page 506\]](#)
- [Marketing Areas \[page 507\]](#)

## Interaction Contacts, Marketing Permissions, and Marketing Subscriptions

### Create Interaction Contacts with Additional IDs

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
InteractionContactOriginData(InteractionContactID='4711',InteractionContactOrigin='SAP_HYBRIS_CONSUMER') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "OriginDataLastChgUTCDateTime": "2017-10-01T13:13:14Z",
  "CityName": "Kiel",
  "Country": "DE",
  "EmailAddress": "otto.normalverbraucher@company.de",
  "FullName": "Normalverbraucher",
  "AddressHouseNumber": "1",
  "MobileNumber": "+49119201412191",
  "PhoneNumber": "+49115",
  "ContactPostalCode": "24105",
```

```

"AddressRegion": "01",
"StreetName": "Hauptstrasse"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs(InteractionContactID='4711',InteractionContactOrigin='SAP_HYBRIS
_CONSUMER',InteractionContactAdditionalOrigin='EMAIL',InteractionContactAdditi
onalExternalID='otto.normalverbraucher@company.de') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs(InteractionContactID='4711',InteractionContactOrigin='SAP_HYBRIS
_CONSUMER',InteractionContactAdditionalOrigin='EMAIL',InteractionContactAdditi
onalExternalID='otto.normalverbraucher5@company.de') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(InteractionContactID='4711',InteractionContactOrigin='SAP
_HYBRIS_CONSUMER',MarketingAttributeCategory='HOBBY',MarketingAttributeValue='S
occer') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(InteractionContactID='4711',InteractionContactOrigin='SAP
_HYBRIS_CONSUMER',MarketingAttributeCategory='HOBBY',MarketingAttributeValue='V
olleyball') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(InteractionContactID='4711',InteractionContactOrigin='SAP_

```

```

HYBRIS_CONSUMER',MarketingAttributeCategory='Spoken_Language',MarketingAttributeValue='English') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(InteractionContactID='4711',InteractionContactOrigin='SAP_HYBRIS_CONSUMER',MarketingAttributeCategory='Spoken_Language',MarketingAttributeValue='Romanian') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Delete Interaction Contact

### i Note

A PUT request is executed to set the IsEndOfPurposeBlocked flag.

### Sample Code

```

--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT InteractionContactOriginData(InteractionContactID='AB20180612001-P',InteractionContactOrigin='SAP_ERP_BUPA') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-07-23T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "IsEndOfPurposeBlocked": true,
  "OriginDataLastChgUTCDateTime": "2018-07-23T12:13:14"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Create Interaction Contacts with Marketing Permissions and Marketing Subscriptions

### i Note

The batch request is sent via http method `POST` containing `PUT` requests to create a new interaction contact, marketing permission and marketing subscription. To update single attributes, you must use the `PATCH` request.

### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
InteractionContactOriginData(InteractionContactID='IC98979992',InteractionCont
actOrigin='SAP_C4C_BUPA') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-03-27T07:14:34'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "OriginDataLastChgUTCDateTime" : "2019-07-01T13:04:46.000",
  "CityName" : "Walldorf",
  "Country" : "DE",
  "EmailAddress" : "max.mustermann@mail.de",
  "PhoneNumber" : "+619022580475611",
  "MobileNumber" : "+622485500519911",
  "FullName" : "Max Mustermann",
  "AddressHouseNumber" : "99",
  "Language" : "DE",
  "ContactPostalCode" : "24105",
  "StreetName" : "Dietmar-Hopp-Allee"
}
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingPermissions(InteractionContactID='IC98979992',InteractionContactOri
gin='SAP_C4C_BUPA',InteractionContactPermissionID='max.mustermann@mail.de',Intac
tnCntctPermissionOrigin='EMAIL',MarketingArea='',CommunicationMedium='EMAIL')
HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.002'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "PermissionUTCDateTime" : "2019-07-01T13:04:46.002",
  "PermissionGranted" : "Y",
  "PermissionSourceCommMedium" : "WEB",
  "IsConfirmationRequired" : false,
  "PermissionNoteText" : "Sample Permission"
}
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingSubscriptions(InteractionContactID='IC98979992',InteractionContactOri
```

```

gin='SAP_C4C_BUPA',IntactnCtctSubscriptionID='max.mustermann@mail.de',Intactn
CtctSubscriptionOrigin='EMAIL',CommunicationMedium='EMAIL',SubscriptionTopic=
'1') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.005",
  "SubscriptionSignUpExists" : "N",
  "SubscriptionSourceCommMedium" : "WEB",
  "IsConfirmationRequired" : false,
  "SubscriptionNoteText" : "Sample Subscription"
}
--changeset_01869434-0010-0001--
--batch--

```

## PATCH: Update Marketing Permissions and Marketing Subscriptions for an Interaction Contact

### Sample Code

```

--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingPermissions(InteractionContactID='IC98979992',InteractionContactOri
gin='SAP_C4C_BUPA',InteractionContactPermissionID='max.mustermann@mail.de',Intac
tnCtctPermissionOrigin='EMAIL',MarketingArea='',CommunicationMedium='EMAIL')
HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "PermissionUTCDateTime" : "2019-07-01T13:04:46.005",
  "PermissionGranted" : "N"
}
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingSubscriptions(InteractionContactID='IC98979992',InteractionContactOri
gin='SAP_C4C_BUPA',IntactnCtctSubscriptionID='max.mustermann@mail.de',Intactn
CtctSubscriptionOrigin='EMAIL',CommunicationMedium='EMAIL',SubscriptionTopic=
'1') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.007",
  "SubscriptionSignUpExists" : "Y"
}
--changeset_01869434-0010-0001--

```

```
--batch--
```

## PUT: Update or Create Marketing Permissions and Marketing Subscriptions for an Interaction Contact

### Note

The sample code has a `PUT` request that updates marketing permissions and marketing subscriptions, or creates new marketing permissions and marketing subscriptions if they do not exist. To update single attributes, you must use the `PATCH` request. In addition, if the value of the property `IsConfirmationRequired` is set to `true`, a double opt-in is executed.

### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingPermissions(InteractionContactID='IC98979992',InteractionContactOrigin='SAP_C4C_BUPA',InteractionContactPermissionID='max.mustermann@mail.de',IntactnCntctPermissionOrigin='EMAIL',MarketingArea='',CommunicationMedium='EMAIL') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.002'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "PermissionUTCDateTime" : "2019-07-01T13:04:46.002",
  "PermissionGranted" : "Y",
  "PermissionSourceCommMedium" : "WEB",
  "IsConfirmationRequired" : false,
  "PermissionNoteText" : "Sample Permission"
}
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingSubscriptions(InteractionContactID='IC98979992',InteractionContactOrigin='SAP_C4C_BUPA',IntactnCntctSubscriptionID='max.mustermann@mail.de',IntactnCntctSubscriptionOrigin='EMAIL',CommunicationMedium='EMAIL',SubscriptionTopic='1') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.005",
  "SubscriptionSignUpExists" : "N",
  "SubscriptionSourceCommMedium" : "WEB",
  "IsConfirmationRequired" : false,
  "SubscriptionNoteText" : "Sample Subscription"
}
--changeset_01869434-0010-0001--
--batch--
```



## GET Requests

### Get all explicit marketing permissions for a specific InteractionContactUUID

```
/sap/opu/odata/sap/API_MKT_INTERACTION_CONTACT_SRV;v=0003/MarketingPermissions?
$filter=InteractionContactUUID eq guid'6c0b84b7-5523-1ed9-a792-18a320d91baf' and
PermissionIsImplicit eq false&$top=10
```

### Get all marketing permissions and marketing subscriptions for an interaction contact with a certain ID and origin

```
/sap/opu/odata/sap/API_MKT_INTERACTION_CONTACT_SRV;v=0003/
InteractionContactOriginData (InteractionContactID='IC98979992
',InteractionContactOrigin='SAP_C4C_BUPA')?
$expand=MarketingPermissions,MarketingSubscriptions
```

### Get all marketing permissions and marketing subscriptions for an InteractionContactUUID

```
/sap/opu/odata/sap/API_MKT_INTERACTION_CONTACT_SRV;v=0003/
InteractionContacts (InteractionContactUUID=guid'6c0b84b7-5523-1ed9-
a792-18a320d91baf')?$expand=MarketingPermissions,MarketingSubscriptions
```

### Get all marketing permissions for an interaction contact with a certain ID and origin

```
/sap/opu/odata/sap/API_MKT_INTERACTION_CONTACT_SRV;v=0003/
InteractionContactOriginData (InteractionContactID='IC98979992
',InteractionContactOrigin='SAP_C4C_BUPA')/MarketingPermissions
```

### Get interaction contact data via ID and origin together with its marketing permissions and marketing subscriptions

```
/sap/opu/odata/sap/API_MKT_INTERACTION_CONTACT_SRV;v=0003/
InteractionContactOriginData (InteractionContactID='IC98979992
',InteractionContactOrigin='SAP_C4C_BUPA')/MarketingSubscriptions
```

### Get an interaction contact via InteractionContactUUIDID together with its marketing permissions and marketing subscriptions

```
/sap/opu/odata/sap/API_MKT_INTERACTION_CONTACT_SRV;v=0003/MarketingSubscriptions?
$filter=InteractionContactUUID eq guid'6c0b84b7-5523-1ed9-a792-18a320d91baf' &
$top=20
```

### Get all marketing permissions for a specific email address of an interaction contact

```
/sap/opu/odata/sap/API_MKT_INTERACTION_CONTACT_SRV;v=0003/MarketingPermissions?
$filter=InteractionContactPermissionID eq 'max.mustermann@mail.de' and
IntactnCntctPermissionOrigin eq 'EMAIL' &$top=20
```

### Get the first 500 interaction contacts that subscribed to newsletter Fashion

```
/sap/opu/odata/sap/API_MKT_INTERACTION_CONTACT_SRV;v=0003/MarketingSubscriptions?
$top=500&$filter=SubscriptionTopicName eq 'Fashion'
```

## Get the first 100 marketing permissions that are newer than a certain date and time

```
/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0003/MarketingPermissions?$top=10&
$filter=PermissionUTCDateTime gt datetimeoffset'2019-01-01T00:00:00.001'
```

## Account Team Members

### PUT

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AccountTeamMembers(InteractionContactID='<InteractionContactID>', InteractionCo
ntactOrigin='<InteractionContactOrigin>', TeamMemberID='<TeamMemberID>', Role='<
Role>') HTTP/1.1Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### DELETE

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
AccountTeamMembers(InteractionContactID='<InteractionContactID>', InteractionCo
ntactOrigin='<InteractionContactOrigin>', TeamMemberID='<TeamMemberID>', Role='<
Role>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
AccountTeamMembers(InteractionContactID='<InteractionContactID>', InteractionCo
ntactOrigin='<InteractionContactOrigin>', TeamMemberID='<TeamMemberID>', Role='<
Role>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "InteractionContactID": "<InteractionContactID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Additional IDs

## PUT

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs(InteractionContactID='<InteractionContactID>', InteractionContact
Origin='<InteractionContactOrigin>', InteractionContactAdditionalOrigin='<Inter
actionContactAdditionalOrigin>', InteractionContactAdditionalExternalID='<Inter
actionContactAdditionalExternalID>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
AdditionalIDs(InteractionContactID='<InteractionContactID>',InteractionContact
Origin='<InteractionContactOrigin>',InteractionContactAdditionalOrigin='<Inter
actionContactAdditionalOrigin>',InteractionContactAdditionalExternalID='<Inter
actionContactAdditionalExternalID>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "InteractionContactID": "<InteractionContactID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Interaction Contact Origin Data

## PUT

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
InteractionContactOriginData(InteractionContactID='<InteractionContactID>',Int
eractionContactOrigin='<InteractionContactOrigin>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
InteractionContactOriginData(InteractionContactID='<InteractionContactID>', InteractionContactOrigin='<InteractionContactOrigin>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "OriginDataLastChgUTCDateTime": "2017-10-01T13:13:14",
  "AddressHouseNumber": "<AddressHouseNumber>",
  "ContactPostalCode": "<ContactPostalCode>",
  "StreetName": "<StreetName>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PUT Single Entity (with ForceSync flag in Request Header)

### Note

When you import single entities, the response body is empty. You can read the status of the import only in the **response header** in the attributes **Status** and **Sap-Message**.

### Sample Code

```
Request: PUT: /sap/opu/odata/SAP/API_MKT_INTERACTION_CONTACT_SRV;v=0003/
InteractionContactOriginData(ContactID='C_20180828_00008', InteractionContactOrigin='SAP_ERP_CONTACT')
{
  "OriginDataLastChgUTCDateTime" : "2017-10-01T13:13:14Z",
  "CityName" : "Kiel",
  "Country" : "DE",
  "EmailAddress" : "otto.normalverbraucher@company.de",
  "FirstName" : "Otto",
  "LastName" : "Normalverbraucher",
  "FullName" : "Otto Normalverbraucher",
  "GenderCode" : "1",
  "AddressHouseNumber" : "1",
  "IsConsumer" : false,
  "IsContactPerson" : true,
  "Language" : "DE",
  "MaritalStatus" : "2",
  "MaritalStatusName" : "Married",
  "MobileNumber" : "+49119201412191",
  "IsObsolete" : false,
  "PhoneNumber" : "+49115",
  "ContactPostalCode" : "24105",
  "AddressRegion" : "01",
  "StreetName" : "Hauptstrasse",
}
```

## Marketing Attributes

### PUT

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(InteractionContactID='<InteractionContactID>', InteractionC
ontactOrigin='<InteractionContactOrigin>', MarketingAttributeCategory='<Marketi
ngAttributeCategory>', MarketingAttributeValue='<MarketingAttributeValue>')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### DELETE

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
MarketingAttributes(InteractionContactID='<InteractionContactID>', InteractionC
ontactOrigin='<InteractionContactOrigin>', MarketingAttributeCategory='<Marketi
ngAttributeCategory>', MarketingAttributeValue='<MarketingAttributeValue>')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingAttributes(InteractionContactID='<InteractionContactID>', InteractionC
ontactOrigin='<InteractionContactOrigin>', MarketingAttributeCategory='<Marketi
ngAttributeCategory>', MarketingAttributeValue='<MarketingAttributeValue>')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "InteractionContactID": "<InteractionContactID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Marketing Areas

## PUT

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAreas(InteractionContactID='<InteractionContactID>', InteractionContac
tOrigin='<InteractionContactOrigin>', InteractionContactMktgArea='<InteractionC
ontactMktgArea>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingAreas(InteractionContactID='<InteractionContactID>', InteractionContactOrigin='<InteractionContactOrigin>', InteractionContactMktgArea='<InteractionContactMktgArea>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "InteractionContactID": "<InteractionContactID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

Parent topic: [Interaction Contacts \[page 469\]](#)

## Related Information

[Basic Concepts \[page 472\]](#)

[Structure of API\\_MKT\\_INTERACTION\\_CONTACT \[page 475\]](#)

[Function Imports \[page 508\]](#)

### 5.2.2.4 Function Imports

Function imports are used to perform custom operations on an entity, which are typically not provided by OData operations. This section contains payload examples for the following function imports:

- [Delete Marketing Areas \[page 509\]](#)
- [Delete All Marketing Areas from Origin \[page 509\]](#)
- [Delete Account Team Members \[page 510\]](#)
- [Delete Marketing Attributes \[page 510\]](#)
- [Delete Additional IDs \[page 511\]](#)



## Delete Marketing Areas

HTTP Method	Function Import
<i>POST</i>	InteractionContactDeleteMarketingArea  Deletes all occurrences of a marketing area from an interaction contact.

### Payload Example

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InteractionContactDeleteMarketingArea?
InteractionContactID='DEV_TEST'&InteractionContactOrigin='SAP_ERP_CONTACT'&Int
eractionContactMktgArea='GLOBAL' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete All Marketing Areas from Origin

HTTP Method	Function Import
<i>POST</i>	IntactnCntctOriginDeleteAllMktgAreas  Deletes all marketing areas from one origin.

### Payload Example

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InteractionContactOriginDeleteAllMktgAreas?
InteractionContactID='DEV_TEST'&InteractionContactOrigin='SAP_ERP_CONTACT'
HTTP/1.1
Content-Length: 1035
Accept: application/json
```

```
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Account Team Members

HTTP Method	Function Import
<i>POST</i>	IntactnCntctDeleteAllAccountTeamMembers Deletes all account team members for one interaction contact.

### Payload Example

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InteractionContactDeleteAllAccountTeamMembers?
InteractionContactID='DEV_TEST'&InteractionContactOrigin='SAP_ERP_CONTACT'
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Marketing Attributes

HTTP Method	Function Import
<i>POST</i>	IntactnCntctOrignDeleteAllMktgAttributes Deletes all marketing attributes from one origin.

## Payload Example

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST IntactnCntctOriginDeleteAllMktgAttributes?
InteractionContactID='DEV_TEST'&InteractionContactOrigin='SAP_ERP_CONTACT'
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Additional IDs

HTTP Method	Function Import
<i>POST</i>	IntactnCntctOriginDeleteAdditionalIDs  Deletes all additional IDs from one origin except for IDs that come from the origin data.

## Payload Example

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST IntactnCntctOriginDeleteAdditionalIDs?
InteractionContactID='DEV_TEST'&InteractionContactOrigin='SAP_ERP_CONTACT'
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Function Import Parameters

Property	Description	Edm Core Type	Max Length	Mandatory	Key
InteractionContactID	ID of Interaction Contact	Edm. String	255	X	X
InteractionContactOrigin	Origin of Interaction Contact	Edm. String	20	X	X
InteractionContactMarketingArea	Marketing Area	Edm. String		X	X

Parent topic: [Interaction Contacts \[page 469\]](#)

## Related Information

[Basic Concepts \[page 472\]](#)

[Structure of API\\_MKT\\_INTERACTION\\_CONTACT \[page 475\]](#)

[Payload Examples for Interaction Contacts \[page 495\]](#)

## 5.2.3 Corporate Accounts

Public OData API (`API_MKT_CORPORATE_ACCOUNT_SRV` Version 3) for reading and writing master data about Corporate Accounts only. Corporate accounts are companies or organizations that interact with your company. The public API for Corporate Account supports operations on the `Corporate Account Business Object` and the `Marketing Permissions Business Object`.

### i Note

We recommend that you use the current version 0003 of this service. If you want to use the previous version, you'll find the help links here:

- Version 0002: [Contact](#), [Interaction Contact](#), [Corporate Account API, Version 0002](#)

## Technical Data



### ⚠ Caution

The API services available in SAP Marketing Cloud must not be used for mass read (GET) operations. In other words, you cannot use them for extracting all available data, for example, to extract millions of contacts or interactions from your marketing system.

Name of the Service	API_MKT_CORPORATE_ACCOUNT
Authorizations	The following business catalog roles are required: <ul style="list-style-type: none"><li>• For version 2: SAP_CEC_BC_MKT_API_IC2_PC</li><li>• For version 3: SAP_CEC_BC_MKT_API_IC3_PC</li></ul>
Communication Scenario ID	SAP_COM_0207
Component for Incidents	<ul style="list-style-type: none"><li>• CEC-MKT-DM-IC (Interaction Contacts)</li><li>• CEC-MKT-DM-PER (Permissions and Subscriptions)</li></ul> <div data-bbox="820 974 912 1008" data-label="Section-Header"><h4>i Note</h4></div> <div data-bbox="820 1028 1361 1095" data-label="Text"><p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p></div>
OData Version	2.0
Root URI	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003
Service Metadata URI:	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$metadata
Field Extensibility Supported	Yes. For more information, search for <b>extensibility</b> in <a href="#">Structure of API_MKT_CORPORATE_ACCOUNT [page 517]</a> . <div data-bbox="820 1590 912 1624" data-label="Section-Header"><h4>i Note</h4></div> <div data-bbox="820 1644 1361 1711" data-label="Text"><p>You need to open the collapsible sections of the document first.</p></div>

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Comment
<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CORPORATE_ACCOUNT_SRV; v=0003/\$metadata?sap- documentation=all</pre>	Only for internal access. You need to provide the server and port names.
<a href="#">Corporate Account Details Page</a> 	<p>General access to the <a href="#">Details</a> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>1. On the <a href="#">Details</a> page, click <a href="#">Download Specification</a> and download as EDMX.</li> <li>2. Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Marketing - Corporate Accounts API</a> 	General access link takes you directly to the <a href="#">Contacts</a> metadata file. One-time registration or logon is required.

### **i** Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

#### [Basic Concepts \[page 515\]](#)

Public OData API (`API_MKT_CORPORATE_ACCOUNT_SRV`) for reading and writing master data about Corporate Accounts only. Corporate accounts are companies or organizations that interact with your company.

#### [Structure of API\\_MKT\\_CORPORATE\\_ACCOUNT \[page 517\]](#)

This document describes the structure of the **Public OData API** `API_MKT_CORPORATE_ACCOUNT`.

#### [Payload Examples for Corporate Accounts \[page 538\]](#)

Payload examples for `API_MKT_CORPORATE_ACCOUNT`.

#### [Function Imports \[page 551\]](#)

Function imports are used to perform custom operations on an entity, which are typically not provided by OData operations. This section contains payload examples for the following function imports:

## 5.2.3.1 Basic Concepts

Public OData API (`API_MKT_CORPORATE_ACCOUNT_SRV`) for reading and writing master data about Corporate Accounts only. Corporate accounts are companies or organizations that interact with your company.


### Processing Info and Best Practices

#### i Note

For generally applicable recommendations and best practices, make sure you refer to the section [Best Practices and Recommended Package Sizes \[page 400\]](#).

- The minimum data required when importing corporate accounts is an ID, an ID Origin, a timestamp, and **at least one other attribute**.
- **When to use `PUT` and `PATCH`:**
  - `PUT` requests are most suitable for an initial data import, for example, when you want to create a new corporate account. A `PUT` request requires that you always send all properties. Any properties that you omit are overwritten by blank entries. That is, any existing entries are deleted. If no record is found, a new record is created. In other words, the `PUT` request functions as a full **upsert**.
  - We recommend that you use `PATCH` requests for all other imports. A `PATCH` request updates only the properties provided in the request body and leaves everything untouched that was not provided. So, you can omit all properties that are not to be changed. Like the `PUT` request, if no record is found, a new record is created with the available properties.. In other words, the `PATCH` request functions as a delta **upsert**.
  - An additional advantage of using `PATCH` is that you specify your own sequence ID. For this reason, it is more flexible than a `PUT` operation, where the sequence ID is set by default and cannot be changed.
  - Basically, since you can use `PATCH` with the same payload as you would use for `PUT`, the `PATCH` operation is more universal and you can work with it exclusively.
  - We recommend that you don't mix `PUT` and `PATCH` operations. Doing so can lead to unwanted results since a `PUT` operation is processed before a `PATCH`.
- **Do not combine a `DELETE` operation with other OData operations in one changeset.**

We recommend that you do not combine the OData operations `PUT`, `PATCH`, `POST`, with a `DELETE` operation in the same changeset. For example, let's say you want to update data for corporate account A by adding an additional email address and at the same time delete a mobile number that is no longer valid. So, you send a `PUT` operation on the `AdditionalId` entity with the new email address and a `DELETE` operation within the same changeset. One of these operations could cancel out the other and the resulting dataset will not be as intended.

**Recommended Practice:** For such combined operations including a `DELETE` operation, we recommend that you always use the relevant function import, which allows deletion of specific entities, together with the appropriate OData operation `PUT`, `PATCH`, or `POST` within the same changeset.
- Batch requests are submitted as a single `HTTP POST` request to the `$batch` endpoint of a service as described in [[OData-URI](#) ]. The batch request must contain a header parameter `content-type`, specifying the value `multipart/mixed` and `boundary=batch`.
- **Use of codes versus free text:** The properties listed in the left column of the table require code values. Incorrect codes will result in import errors, indicating that the corresponding code is not valid. If you are

not thoroughly familiar with the internal codes available in SAP Marketing for these properties, you should use properties that allow a free text. For example, if you do not know that **DE** is the country code for **Germany**, you can use **Germany** as the free text.

Code in SAP Marketing	Free Text Property
Country	CountryName
Industry	IndustryName
Department	DepartmentName
Function	FunctionName
GenderCode	GenderCodeName
Language	LanguageName
MaritalStatus	MaritalStatusName
AddressRegion	RegionName
FormOfAddress	FormOfAddressName

You must map your free text names to the available codes in the [Map Free Text](#) app. For more information, see [Map Free Texts](#).

- Do not mix different types of services for operations involving the same data source. For example, when importing contacts from a web shop, do not use the CUAN\_IMPORT service for a PUT operation and then the API\_MKT\_CORPORATE\_ACCOUNTS service to PATCH contacts. You can, however, migrate from CUAN\_IMPORT to the API\* services.
- One contact can be assigned to a maximum of one corporate account, while one corporate account can have more than one contact.
- The origin that you pass via the property *ContactOrigin* cannot be shareable. If the **main** origin is set to *Shareable*, this will trigger an error. For more information, see [Configuring Origins](#). You can view sample payloads and test the API at [https://api.sap.com/api/API\\_MKT\\_CORPORATE\\_ACCOUNT\\_SRV\\_0003/resource](https://api.sap.com/api/API_MKT_CORPORATE_ACCOUNT_SRV_0003/resource).

## i Note

The UTC timestamp of permissions can't lie in the future.

When you import permissions, they must not have a timestamp that lies in the future. The timestamp of imported permissions is always in UTC. The field name in the OData service is called `PermissionUTCDateTime`.

If you want to use your local timestamp, you have to add the time zone information, that is, your local time zone together with the time zone offset or enter a timestamp that is converted to UTC.

The date and time information is adapted by the standard time difference (offset) with +01:00 for Central European Time (CET) or -05:00 for Eastern Standard Time (EST). For example:

2019-01-01T12:00:00+01:00

If you live east of UTC and enter your timestamp in your local time zone without time zone offset, this will result in a future timestamp. For example, you live in Germany and your local time is 8 a.m. on November, 28. If you enter this as the UTC timestamp without a time zone offset, the UTC permission timestamp will show as 8 a.m., November 28, while in the UTC time zone it's 7 a.m., November 28. You've created a UTC permission timestamp that lies in the future and is invalid.



## Error Messages

If the OData service is not accessible, for example due to missing authorization, or because the system is not available, a corresponding HTTP status code is returned.

If the OData service is accepted by the gateway component in the SAP Marketing Cloud system, the HTTP status code 201 or 204 is returned. Any processing errors are recorded in the SAP Marketing Cloud system in the *Import Monitor* app, where they can be monitored, restarted and discarded.

By default, data processing for contacts, interaction contacts, corporate accounts, or marketing permissions is asynchronous. In most cases an OK response, such as a receipt notification, is returned almost immediately. An exception to this would be data uploads that might contain severe errors, such as parse or format errors, and so would not return an OK response but an error message. The data you upload lands in a staging area, where it is then further processed. You can change the default setting to synchronous processing by setting the property `Sap-Cuan-ForceSynchronousProcessing` to True. In this case, any error messages are returned as soon as they are detected.

To view the processing status and to check for errors or success messages, you must launch the *Import Monitor* app. Messages for marketing permissions in this app are displayed under the API for Contact, API for Interaction Contacts, or API for Corporate Accounts depending on the API OData service you use. In the event of errors, you can restart or discard the import in the *Import Monitor*.

## Field Extensibility

In addition to the pre-delivered attributes, you can add customer-specific fields using the *Custom Fields* app. For more information about how to do this, see [Custom Fields](#).

Please enable the Data Source under UIs and Reports: `API_MKT_CORPORATE_ACCOUNT_SRV 0002`

**Parent topic:** [Corporate Accounts \[page 512\]](#)

## Related Information

[Structure of API\\_MKT\\_CORPORATE\\_ACCOUNT \[page 517\]](#)

[Payload Examples for Corporate Accounts \[page 538\]](#)

[Function Imports \[page 551\]](#)

### 5.2.3.2 Structure of API\_MKT\_CORPORATE\_ACCOUNT

This document describes the structure of the **Public OData API** `API_MKT_CORPORATE_ACCOUNT`.

Make sure you read these topics before you start:

- [Best Practices and Recommended Package Sizes \[page 400\]](#)

- [Basic Concepts \[page 515\]](#)

## Request Header

The request header contains the additional header fields listed in the table. Remember to include at least the mandatory request header fields in each payload.

Property	Example	Description	Max. Length	Mandatory
Sap-Cuan-RequestTimestamp	'2017-09-28T12:13:14'	Timestamp of the import run in this format.		X
Sap-Cuan-SequenceId	PatchAddress	This defines a set of fields that are to be updated, for example, address fields, which can be interpreted as a field group. The combination of the header fields Sap-Cuan-SequenceId and Sap-Cuan-RequestTimestamp is used to check the sequence of the data received. If the data that is received has a timestamp older than already imported data, it is ignored.		X (only mandatory for Patch Mode)
Sap-Cuan-SourceSystemType	EXT	Type of source system. This is a free text field.	20	
Sap-Cuan-SourceSystemId	HYBRIS	Identifier of source system. This is a free text field.	255	

Property	Example	Description	Max. Length	Mandatory
Sap-Cuan-ForceSynchronousProcessing	X	<p>This flag is deselected by default, which means that uploaded data is processed asynchronously. On upload, a success message is output immediately, unless there are errors such as authorization issues or bad requests. Objects are uploaded to the staging area and processed successively from there. All status messages can be displayed in the <a href="#">Import Monitor</a> app.</p> <p>You can force <b>synchronous processing</b> of imports by setting this flag. In this case, an error message will be returned as soon as an error is detected. Such error messages are output in the <a href="#">Import Monitor</a> app</p>		
Sap-Cuan-ReferenceId	345g67980907	External reference of the inbound message	32	

## Entity Sets

The Corporate OData API provides the following entity sets:

Entity Set	Description	Path
CorporateAccount	This entity contains all contact information from the root.	/CorporateAccount
AccountTeamMembers	This entity contains information about the account team members.	/AccountTeamMembers
<div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>As <code>TeamMemberID</code>, you must enter the employee ID.</p> </div>		
AdditionalIDs	This entity contains information about additional IDs.	/AdditionalIDs

Entity Set	Description	Path
CorporateAccountOriginData	This entity contains contact origin data. <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <p>The property <code>OriginDataLastChgUTCDateTime</code> is mandatory and must be specified.</p> </div>	/CorporateAccountOriginData
MarketingAttributes	This entity contains information about marketing attributes.	/MarketingAttributes
MarketingAreas	This entity contains information about marketing areas.	/MarketingAreas

## Corporate Accounts

**GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/CorporateAccount`

**Field Extensibility:** The following business contexts are relevant: [Marketing: Corporate Account](#) and [Marketing: Contact and Corporate Account](#)

You can perform the following operations on the `Corporate Account` entity set:

HTTP Method	Description	Path
<i>GET</i>	Get a list of corporate accounts. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> . <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>A maximum of 5000 corporate accounts can be fetched in a single request</li> <li>Specification of TOP is mandatory.</li> </ul> </div>	/CorporateAccount?\$top=1
	Get the details of a specific corporate accounts using the <code>Corporate Account</code> UUID.	/CorporateAccounts(guid'<CorporateAccount UUID>')

## AccountTeamMembers

You can perform the following operations on the `AccountTeamMember` entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/AccountTeamMembers`
- **PUT, PATCH, or DELETE in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/$batch`

- **PUT, PATCH, or DELETE in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/AccountTeamMembers(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',TeamMemberID='<TeamMemberID>',Role='<Role>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of account team members.	/AccountTeamMembers?\$top=1  This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby
		<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 account team members can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>
	Get the details of a specific account team member.	This operation is not supported.
<i>POST (Batch)</i>	Update or create an account team member in batch mode.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</code>
	Delete an account team member in batch mode.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</code>
	Add one new account team member	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</code>
<i>PUT</i>	Update or create an account team member.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/AccountTeamMembers(CorporateAccountID='&lt;CorporateAccountID&gt;',CorporateAccountOrigin='&lt;CorporateAccountOrigin&gt;',TeamMemberID='&lt;TeamMemberID&gt;',Role='&lt;Role&gt;')</code>

HTTP Method	Description	Path
<i>PATCH</i>	Add one new account team member.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CORPORATE_ACCOUNT_SRV;v =0003/ AccountTeamMembers (CorporateAccountID='&lt;Corporate AccountID&gt;',CorporateAccountOri gin='&lt;CorporateAccountOrigin&gt;', TeamMemberID='&lt;TeamMemberID&gt;',R ole='&lt;Role&gt;')</pre>
<i>DELETE</i>	Delete an account team member.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CORPORATE_ACCOUNT_SRV;v =0003/ AccountTeamMembers (CorporateAccountID='&lt;Corporate AccountID&gt;',CorporateAccountOri gin='&lt;CorporateAccountOrigin&gt;', TeamMemberID='&lt;TeamMemberID&gt;',R ole='&lt;Role&gt;')</pre>

## AdditionalIDs

You can perform the following operations on the AdditionalIDs entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/AdditionalIDs`
- **PUT, PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/AdditionalIDs(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',InteractionContactAdditionalOrigin='<InteractionContactAdditionalOrigin>',InteractionContactAdditionalExternalID='<InteractionContactAdditionalExternalID>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of additional IDs by Account ID and ID Origin.	<p>/AdditionalIDs?\$top=1</p> <p>This method supports standard OData parameters such as \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 additional IDs can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> <li>• \$filter is not supported for additional IDs.</li> </ul> </div>
	Get the details of a specific additional ID.	<p>/</p> <p>AdditionalIDs ('&lt;CorporateAccountID&gt;,&lt;CorporateAccountOrigin&gt;,&lt;InteractionContactAdditionalOrigin&gt;,&lt;InteractionContactAdditionalExternalID&gt;')</p>
<i>POST (Batch)</i>	Update or create an additional ID in batch mode.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</p>
	Append one new additional ID	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</p>
<i>PUT</i>	Update or create an additional ID.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/AdditionalIDs (CorporateAccountID='&lt;CorporateAccountID&gt;', CorporateAccountOrigin='&lt;CorporateAccountOrigin&gt;', InteractionContactAdditionalOrigin='&lt;InteractionContactAdditionalOrigin&gt;', InteractionContactAdditionalExternalID='&lt;InteractionContactAdditionalExternalID&gt;')</p>

HTTP Method	Description	Path
<i>PATCH</i>	Add one new additional ID.	<pre> https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CORPORATE_ACCOUNT_SRV;v =0003/ AdditionalIDs (CorporateAccountI D='&lt;CorporateAccountID&gt;', Corpor ateAccountOrigin='&lt;CorporateAcc ountOrigin&gt;', InteractionContact AdditionalOrigin='&lt;InteractionC ontactAdditionalOrigin&gt;', Intera ctionContactAdditionalExternalI D='&lt;InteractionContactAdditional ExternalID&gt;')</pre>

## CorporateAccountOriginData

You can perform the following operations on the CorporateAccountOriginData entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/CorporateAccountOriginData`
- **PUT, PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/CorporateAccountOriginData (CorporateAccountID='<CorporateAccountID>', CorporateAccountOrigin='<CorporateAccountOrigin>')`

**Field Extensibility:** The following business contexts are relevant: *Marketing: Corporate Account* and *Marketing: Contact and Corporate Account*



HTTP Method	Description	Path
<i>GET</i>	Get a list of Corporate Account Origin Data.	<p>/CorporateAccountOriginData?\$top=1</p> <p>This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 corporate account origin data entities can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>
	Get the details of specific corporate account origin data.	<p>/CorporateAccountOriginData ('&lt;CorporateAccountID&gt;', &lt;CorporateAccountOrigin&gt;')</p>
<i>POST (Batch)</i>	Update or create contact origin data in batch mode. Creates a CorporateAccount if the CorporateAccount not exist.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <p>The property OriginDataLastChgUTCDateTime is mandatory and must be specified.</p> </div>
	Delta Update PATCH attributes of the entity CorporateAccountOriginData	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</p>
<i>PUT</i>	Update or create corporate account origin data. This creates a contact if the corporate account not exist.	<p>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/CorporateAccountOriginData (CorporateAccountID='&lt;CorporateAccountID&gt;', CorporateAccountOrigin='&lt;CorporateAccountOrigin&gt;')</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <p>The property OriginDataLastChgUTCDateTime is mandatory and must be specified.</p> </div>

HTTP Method	Description	Path
<i>PATCH</i>	Delta Update PATCH attributes of the entity CorporateAccountOriginData.	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CORPORATE_ACCOUNT_SRV;v =0003/ CorporateAccountOriginData (Corp orateAccountID=' &lt;CorporateAccountID&gt;', Corporate AccountOrigin='&lt;CorporateAccoun tOrigin&gt;')</pre>

## MarketingAttributes

You can perform the following operations on the MarketingAttributes entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingAttributes`
- **PUT, PATCH, or DELETE in BATCH:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/$batch`
- **PUT, PATCH, or DELETE in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingAttributes (CorporateAccountID='<CorporateAccountID>', CorporateAccountOrigin='<CorporateAccountOrigin>', MarketingAttributeCategory='<MarketingAttributeCategory>', MarketingAttributeValue='<MarketingAttributeValue>')`

**Field Extensibility:** The following business contexts are relevant: *Marketing: Marketing Attributes for Contacts*.

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing attributes by Contact ID and ID Origin.	<pre>/MarketingAttributes?\$top=1</pre> <p>This method supports standard OData parameters such as <code>\$filter</code>, <code>\$select</code>, <code>\$top</code>, <code>\$skip</code>, <code>\$count</code>, <code>\$inlinecount</code>, and <code>\$orderby</code></p> <div style="border: 1px solid #0070c0; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 marketing attributes can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>

HTTP Method	Description	Path
	Get the details of a specific marketing attribute.	/ MarketingAttributes ('<CorporateAccountID>,<CorporateAccountOrigin>,<MarketingAttributeCategory>,<MarketingAttributeValue>')
<i>POST</i>	Update or create marketing attributes in batch mode.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch
<i>POST</i>	Delete marketing attributes in batch mode.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch
<i>POST</i>	Append one new Marketing Attribute	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch
<i>PUT</i>	Update or create marketing attributes.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingAttributes(ContactID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',MarketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValue>')
<i>PATCH</i>	Add one new marketing attribute.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingAttributes(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',MarketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValue>')
<i>DELETE</i>	Delete marketing attributes.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingAttributes(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',MarketingAttributeCategory='<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValue>')

## MarketingAreas

You can perform the following operations on the MarketingAreas entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003`
- **PUT, PATCH in BATCH:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/$batch`
- **PUT, PATCH in a single operation:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingAreas(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>')`

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing areas by Contact ID and ID Origin.	<code>/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0003/Contacts?\$expand=MarketingAreas&amp;\$top=2</code>  This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>
		<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5000 marketing areas can be fetched in a single request</li> <li>• Specification of TOP is mandatory.</li> </ul> </div>
	Get the details of a specific marketing area.	<code>/MarketingAreas('&lt;CorporateAccountID&gt;,&lt;CorporateAccountOrigin&gt;,&lt;InteractionCorporateAccountMktgArea&gt;')</code>
<i>POST</i>	Update or create marketing areas in batch mode.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</code>
<i>POST</i>	Append one new Marketing Area	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/\$batch</code>

HTTP Method	Description	Path
<i>PUT</i>	Update or create marketing areas.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingAreas(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>')
<i>PATCH</i>	Add one new marketing area.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingAreas(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',InteractionContactMktgArea='<InteractionContactMktgArea>')

## MarketingPermissions

**Entity Path:** /MarketingPermissions

**Field Extensibility:** The following business context is relevant: [Marketing: Marketing Permissions](#). Custom fields for business object MKT\_PERMISSION (Marketing: Permission) are only supported if you use **version 2** of the API\_MKT\_CORPORATE\_ACCOUNT service.

### i Note

- For all HTTP operations both \$batch requests and single requests can be used.
- Interactions are assigned when marketing permissions are created or updated to allow for analysis of corporate accounts.

You can perform the following operations on the MarketingPermissions entity set:

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing permissions by Corporate Account. This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby	/MarketingPermissions?\$top=1

**i Note**

- A maximum of 5000 marketing permissions can be fetched in a single request
- Specification of TOP is mandatory.

HTTP Method	Description	Path
<i>PATCH</i>	Update or create marketing permissions. This creates a marketing permission if the permission does not exist.	/ MarketingPermissions(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',CorporateAccountPermissionID='<CorporateAccountPermissionID>',CorporateAccountOrigin='<CorporateAccountPermissionOrigin>',MarketingArea='<MarketingArea>',CommunicationMedium='<CommunicationMedium>')
<i>PUT</i>	Update or create marketing permissions. This creates a marketing permission if the permission does not exist.  Delta Update of PATCH attributes of the entity MarketingPermission.	/ MarketingPermissions(CorporateAccountID='<CorporateAccountID>',CorporateAccountOrigin='<CorporateAccountOrigin>',CorporateAccountPermissionID='<CorporateAccountPermissionID>',CorporateAccountOrigin='<CorporateAccountPermissionOrigin>',MarketingArea='<MarketingArea>',CommunicationMedium='<CommunicationMedium>')

The table below describes the properties for the entity `MarketingPermissions`.

## MarketingPermissions Property Names and Descriptions

Property Name	Property Description	Usage
CorporateAccountID	<p>The CorporateAccountID and CorporateAccountOrigin identify the corporate account uniquely.</p> <p><b>Example:</b> a business partner ID from the CRM system.</p>	Mandatory
CorporateAccountUUID	<p>Unique ID of a corporate account in SAP Marketing Cloud .</p>	Read-Only
CorporateAccountOrigin	<p>The CorporateAccountID and CorporateAccountOrigin identify the contact uniquely.</p> <p><b>Example:</b> SAP_CRM_BUPA</p>	Mandatory
CorporateAccountPermissionID	<p>The CorporateAccountPermissionID and CorporateAccountPermissionOrigin store marketing permissions.</p> <p><b>Example:</b> first.lastname@company.de</p>	Mandatory
CorpAcctPermissionOrigin	<p>The CorporateAccountPermissionID and CorporateAccountPermissionOrigin store marketing permissions.</p> <p>CorpAcctPermissionOrigin is the origin of the corporate account ID that stores marketing permissions.</p> <p>By defining the origin, you determine that a corporate account with an ID associated to a source is eligible to be analyzed.</p> <p>You can configure origins of contacts IDs in the <a href="#">Configuring Origins</a> configuration app.</p> <p><b>Example:</b> EMAIL</p>	Mandatory
CorpAcctPermissionOriginName	<p>Description of property <b>CorpAcctPermissionOrigin</b></p>	Read-Only
MarketingArea	<p>Identifies an area of responsibility or an organizational unit.</p> <p>You use a marketing area to restrict access to instances of an object, such as campaign, email message, email template, target group, or permission.</p>	<p>Mandatory</p> <p>The MarketingArea property field must be passed, but can be left empty.</p>

Property Name	Property Description	Usage
MarketingAreaName	Description of property <b>MarketingArea</b>	Read-Only
CommunicationMedium	<p>Represents the type of permission, for example, <b>EMAIL</b> or <b>PHONE</b>.</p> <p>You can configure communication media in the <a href="#">Managing Interaction Content</a> configuration app.</p>	Mandatory
CommunicationMediumName	Description of property <b>CommunicationMedium</b>	
PermissionUUID	Unique ID of a permission in SAP Marketing Cloud .	
PermissionUTCDateTime	<p>This is the timestamp for when the permission was given or removed.</p> <div data-bbox="603 927 991 1081" style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px;"> <p><b>i Note</b></p> <p>The time stamp must not be initial or null.</p> </div>	Mandatory
PermissionSourceObject	<p>This field provides information on the source of the permission, that is, where it came from. For example, the ID of a landing page.</p> <p>This field can be filled with freetext.</p>	<p>If you enter a value for the PermissionSourceObject property, you must also specify a value for the PermissionSourceObjectType.</p> <p>Both fields must be filled or left empty.</p>
PermissionSourceObjectType	<p>This field provides information on the source of the permission and its type. For example, the business object name of a landing page.</p> <p>This field can be filled with freetext.</p>	
PermissionSourceSystem	<p>This is the system that stores the permission. For example, ABD client 100.</p> <p>This field can be filled with freetext.</p>	If you enter a value for the PermissionSourceSystem property, you must also specify a value for the PermissionSourceSystemType.
PermissionSourceSystemType	<p>This is the type of system where the permission is stored. For example, SAP_CEI.</p> <p>This field can be filled with freetext.</p>	



Property Name	Property Description	Usage
PermissionSourceCommMedium	Indicates where the permission comes from, such as WEB, EMAIL, or PHONE. In case PermissionSourceCommMedium is not filled, this property is set to WEB.	
PermissionSourceCommMediumName	Description of property <b>PermissionSourceCommMedium</b>	
PermissionIsImplicit	<p>If the system sets this field to TRUE, then it is an implicit permission, which is determined by country-specific regulation.</p> <p>If the system sets this field to FALSE, the contact has given this permission explicitly.</p>	Read-Only
PermissionNoteText	A text to describe a permission change.	
IsConfirmationRequired	<p>This is a boolean parameter. If the parameter is set to TRUE, the permission is stored using the double opt-in or opt-out process.</p> <p>If the property is not specified in the payload or it is set to FALSE the permission is directly stored.</p>	
LastChangedByUser	Name of the user who has changed the permissions last.	Read-Only
LastChangeDateTime	Date and time of the last permission change.	Read-Only

## MarketingSubscriptions

**Entity Path:** /MarketingSubscriptions

**Field Extensibility:** The following business context is relevant: [Marketing: Marketing Permissions](#). Custom fields for business object MKT\_PERMISSION (Marketing: Permission) are only supported if you use **version 2** or **version 3** of the API\_MKT\_CONTACT service.

### i Note

- For all HTTP operations both \$batch requests and single requests can be used.
- Interactions are assigned when marketing permissions are created or updated to allow for analysis of contacts.

You can perform the following operations on the `MarketingSubscriptions` entity set:

HTTP Method	Description	Path
<i>GET</i>	Get a list of marketing subscriptions by Contact ID and ID Origin. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	<code>/MarketingSubscriptions?\$top=1</code>
	<p><b>Note</b></p> <ul style="list-style-type: none"> <li>A maximum of 5000 marketing subscriptions can be fetched in a single request</li> <li>Specification of TOP is mandatory.</li> </ul>	
<i>PATCH</i>	Update or create subscriptions. This creates a subscription if the subscription does not exit.	<code>/MarketingSubscriptions(CorporateAccountID='&lt;CorporateAccountID&gt;',CorporateAccountOrigin='&lt;ContactOriginCorporateAccount&gt;'CorporateAccountPermissionID='&lt;CorporateAccountPermissionID&gt;',CorporateAccountPermissionOrigin='&lt;CorporateAccountPermissionOrigin&gt;',CommunicationMedium='&lt;CommunicationMedium&gt;',SubscriptionTopic='&lt;SubscriptionTopic&gt;')</code>
<i>PUT</i>	Update or create subscriptions. This creates a subscription if the subscription does not exit.  Delta Update of PATCH attributes of the entity <code>MarketingSubscriptions</code> .	<code>/MarketingSubscriptions(CorporateAccountID='&lt;CorporateAccountID&gt;',CorporateAccountOrigin='&lt;ContactOriginCorporateAccount&gt;'CorporateAccountPermissionID='&lt;CorporateAccountPermissionID&gt;',CorporateAccountPermissionOrigin='&lt;CorporateAccountPermissionOrigin&gt;',CommunicationMedium='&lt;CommunicationMedium&gt;',SubscriptionTopic='&lt;SubscriptionTopic&gt;')</code>

## Marketing Subscription Property Descriptions

The table describes the properties for the `MarketingSubscription` entity.

MarketingSubscription Property Names and Descriptions

Property Name	Property Description	Usage
CorporateAccountID	<p>The CorporateAccountID and CorporateAccountOrigin identify the contact uniquely.</p> <p><b>Example:</b> a business partner ID from the CRM system.</p>	
CorporateAccountOrigin	<p>The CorporateAccountID and CorporateAccountOrigin identify the contact uniquely.</p> <p>The CorporateAccountID will not be saved to the MarketingSubscription but is only used to derive a uniqueCorporateAccountUUID. This data will not be returned in GET requests.</p> <p><b>Example:</b> SAP_CRM_BUPA</p>	
CorporateAccountSubscriptionID	<p>The InteractionContactPermissionID and InteractionContactSubscriptionOrigin store marketing subscription.</p>	Mandatory
CorpAcctSubscriptionOrigin	<p>The CorporateAccountSubscriptionID and CorpAcctSubscriptionOrigin store marketing subscriptions.</p> <p>CorpAcctSubscriptionOrigin is the origin of a corporate account ID that stores marketing subscriptions. The origin indicates the source of an ID. By defining the origin, you determine that a corporate account with an ID associated to a source can be analyzed.</p> <p><b>Example:</b> EMAIL</p> <p>You can configure origins of contact IDs in the <a href="#">Configuring Origins</a> configuration app.</p>	Mandatory
CorpAcctSubscriptionOriginName	<p>Description of property <b>CorpAcctSubscriptionOriginName</b></p>	Read-Only

Property Name	Property Description	Usage
CommunicationMedium	Represents the type of subscription, for example, <b>EMAIL</b> or <b>PHONE</b> .  You can configure communication media in the <a href="#">Managing Interaction Content</a> configuration app.	Mandatory
CommunicationMediumName	Description of property <b>CommunicationMedium</b>	Read-Only
CorporateAccountUUID	Unique ID of a corporate account in SAP Marketing Cloud .	Read-Only  The field value is returned internally.
SubscriptionUUID	Unique ID of a subscription in SAP Marketing Cloud .	
SubscriptionUTCDateTime	This is the timestamp for when the subscription was given or removed.  <div data-bbox="603 981 991 1133" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>i Note</b> The time stamp must not be initial or null.</p> </div>	Mandatory
SubscriptionSignUpExists	The subscription can be YES (Y) or NO (N).	Mandatory
SubscriptionTopic	Represents a newsletter in SAP Marketing Cloud .	Mandatory  The SubscriptionTopic property field must be passed, but can be left empty.  If you want to create a newsletter subscription, you must specify the SubscriptionTopic.
SubscriptionTopicName	Name of the subscription topic.	
SubscriptionSourceObject	This field provides information on the source of the subscription, that is, where it came from. For example, the ID of a landing page.  This field can be filled with freetext.	

Property Name	Property Description	Usage
SubscriptionSourceObjectType	This field provides information on the source of the subscription and its type. For example, the business object name of a landing page.  This field can be filled with freetext.	
SubscriptionSourceSystem	This is the system that stores the subscription. For example, your local system ID.  This field can be filled with freetext.	
SubscriptionSourceSystemType	This is the type of system where the subscription is stored. For example, SAP_CEI.  This field can be filled with freetext.	
SubscriptionSourceCommMedium	Indicates where the subscription comes from, such as WEB, EMAIL, or PHONE. In case SubscriptionSourceCommMedium is not filled, this property is set to WEB.	
SubscriptionSourceCommMedium-Name	Description of property <b>Subscription-SourceCommMedium</b>	
IsConfirmationRequired	This is a boolean parameter. If the parameter is set to TRUE, the subscription is stored using the double opt-in or opt-out process.  If the property is not specified in the payload or it is set to FALSE the subscription is directly stored.	
LastChangedByUser	Name of the user who has changed the subscription last.	Read-Only
LastChangeDateTime	Date and time of the last permission change.	Read-Only
SubscriptionNoteText	A text to describe a subscription change.	

Parent topic: [Corporate Accounts \[page 512\]](#)

## Related Information

[Basic Concepts \[page 515\]](#)

[Payload Examples for Corporate Accounts \[page 538\]](#)

[Function Imports \[page 551\]](#)

### 5.2.3.3 Payload Examples for Corporate Accounts

Payload examples for API\_MKT\_CORPORATE\_ACCOUNT.

#### Note

- Before you start, please read the **Processing Info and Best Practices** section in [Basic Concepts \[page 515\]](#).
- Remember to include at least the mandatory request header fields in each payload.

#### Available Payload Examples

- [Corporate Accounts, Marketing Permissions, and Marketing Subscriptions \[page 538\]](#)
- [GET Requests \[page 544\]](#)
- [Account Team Members \[page 545\]](#)
- [Additional IDs \[page 546\]](#)
- [Corporate Account Origin Data \[page 547\]](#)
- [Marketing Attributes \[page 548\]](#)
- [Marketing Areas \[page 550\]](#)

## Corporate Accounts, Marketing Permissions, and Marketing Subscriptions

### Create Corporate Accounts with Additional IDs

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
CorporateAccountOriginData (CorporateAccountID='47110815',CorporateAccountOrigin='SAP_ERP_CUSTOMER') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
```

```

Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "OriginDataLastChgUTCDateTime" : "2017-10-01T13:13:14Z",
  "CityName" : "Walldorf",
  "Country" : "DE",
  "EmailAddress" : "info.germany@sap.de",
  "FullName" : "SAP SE",
  "AddressHouseNumber" : "16",
  "Language" : "EN",
  "PhoneNumber" : "+496227747474",
  "ContactPostalCode" : "69190",
  "StreetName" : "Dietmar-Hopp-Allee"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs (CorporateAccountID='47110815',CorporateAccountOrigin='SAP_ERP_CU
STOMER',InteractionContactAdditionalOrigin='EMAIL',InteractionContactAdditiona
lExternalID='info2.germany@sap.de') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
AdditionalIDs (CorporateAccountID='47110815',CorporateAccountOrigin='SAP_ERP_CU
STOMER',InteractionContactAdditionalOrigin='SAP_CRM_BUPA',InteractionContactAd
ditionalExternalID='47110815') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes (CorporateAccountID='47110815',CorporateAccountOrigin='SAP_
ERP_CUSTOMER',MarketingAttributeCategory='Company_Size',MarketingAttributeValu
e='Big') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes (CorporateAccountID='47110815',CorporateAccountOrigin='SAP_
ERP_CUSTOMER',MarketingAttributeCategory='Spoken_Language',MarketingAttributeV
alue='English') HTTP/1.1
Content-Length: 1035
Accept: application/json

```

```

Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(CorporateAccountID='47110815',CorporateAccountOrigin='SAP_
ERP_CUSTOMER',MarketingAttributeCategory='Spoken_Language',MarketingAttributeV
alue='German') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingAttributes(CorporateAccountID='47110815',CorporateAccountOrigin='SAP_
ERP_CUSTOMER',MarketingAttributeCategory='Spoken_Language',MarketingAttributeV
alue='French') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Delete Corporate Account

### i Note

A PUT request is executed to set the `IsEndOfPurposeBlocked` flag.

### Sample Code

```

--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT CorporateAccountOriginData(CorporateAccountID='AB20180612001-
P',CorporateAccountOrigin='SAP_ERP_BUPA') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-07-23T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
  "IsEndOfPurposeBlocked": true,
  "OriginDataLastChgUTCDateTime": "2018-07-23T12:13:14"
}

```



```
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--  
--batch--
```

## Create Corporate Accounts with Marketing Permissions and Marketing Subscriptions

### i Note

The batch request is sent via http method `POST` containing `PUT` requests to create a new corporate account, marketing permission and marketing subscription. To update single attributes, you must use the `PATCH` request.

### Sample Code

```
--batch  
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001  
--changeset_01869434-0010-0001  
content-type: application/http  
content-transfer-encoding: binary  
PUT  
CorporateAccountOriginData (CorporateAccountID='A98979992',CorporateAccountOrigin='SAP_C4C_BUPA') HTTP/1.1  
Accept: application/json  
Sap-Cuan-RequestTimestamp: '2019-03-27T07:14:34'  
Sap-Cuan-SourceSystemType: EXT  
Sap-Cuan-SourceSystemId: C4C  
sap-cuan-referenceid: REQ1  
Content-Type: application/json  
Content-Length: 1021  
{  
  "OriginDataLastChgUTCDateTime" : "2019-07-01T13:04:46.000",  
  "CityName" : "Walldorf",  
  "Country" : "DE",  
  "EmailAddress" : "info@company.de",  
  "PhoneNumber" : "+619022580475611",  
  "MobileNumber" : "+622485500519911",  
  "FullName" : "Company GmbH",  
  "AddressHouseNumber" : "99",  
  "Language" : "DE",  
  "ContactPostalCode" : "24105",  
  "StreetName" : "Dietmar-Hopp-Allee"  
}  
--changeset_01869434-0010-0001  
content-type: application/http  
content-transfer-encoding: binary  
PUT  
MarketingPermissions (CorporateAccountID='A98979992',CorporateAccountOrigin='SAP_C4C_BUPA',CorporateAccountPermissionID='info@company.de',CorpAcctPermissionOrigin='EMAIL',MarketingArea='',CommunicationMedium='EMAIL') HTTP/1.1  
Accept: application/json  
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.002'  
Sap-Cuan-SourceSystemType: EXT  
Sap-Cuan-SourceSystemId: C4C  
sap-cuan-referenceid: REQ1  
Content-Type: application/json  
Content-Length: 1021  
{  
  "CorporateAccountID" : "A98979992",  
  "CorporateAccountOrigin" : "SAP_C4C_BUPA",  
  "CorporateAccountPermissionID" : "info@company.de",  
  "CorpAcctPermissionOrigin" : "EMAIL",  
  "PermissionUTCDateTime" : "2019-07-01T13:04:46.001",  
  "PermissionGranted" : "Y",  
  "PermissionSourceCommMedium" : "WEB",  
  "IsConfirmationRequired" : false,
```

```

    "PermissionNoteText" : "Sample Permission"
  }
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingSubscriptions (CorporateAccountID='A98979992',CorporateAccountOrigin='SAP_C4C_BUPA',CorporateAccountSubscriptionID='info@company.de',CorpAcctSubscriptionOrigin='EMAIL',CommunicationMedium='EMAIL',SubscriptionTopic='1')
HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
{
  "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.001",
  "SubscriptionSignUpExists" : "N",
  "SubscriptionSourceCommMedium" : "WEB",
  "IsConfirmationRequired" : false,
  "SubscriptionNoteText" : "Sample Subscription"
}
--changeset_01869434-0010-0001--
--batch--

```

## PATCH: Update Marketing Permissions and Marketing Subscriptions for a Corporate Account

### ☰ Sample Code

```

--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingPermissions (CorporateAccountID='A98979992',CorporateAccountOrigin='SAP_C4C_BUPA',CorporateAccountPermissionID='info@company.de',CorpAcctPermissionOrigin='EMAIL',MarketingArea='',CommunicationMedium='EMAIL') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.002'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
  {
    "PermissionUTCDateTime" : "2019-07-01T13:04:46.001",
    "PermissionGranted" : "Y"
  }
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PATCH
MarketingSubscriptions (CorporateAccountID='A98979992',CorporateAccountOrigin='SAP_C4C_BUPA',CorporateAccountSubscriptionID='info@company.de',CorpAcctSubscriptionOrigin='EMAIL',CommunicationMedium='EMAIL',SubscriptionTopic='1')
HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json

```

```

Content-Length: 1021
  {
    "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.001",
    "SubscriptionSignUpExists" : "N"
  }
--changeset_01869434-0010-0001--
--batch--

```

## PUT: Update or Create Marketing Permissions and Marketing Subscriptions for a Corporate Account

### i Note

The sample code has a `PUT` request that updates marketing permissions and marketing subscriptions, or creates new marketing permissions and marketing subscriptions if they do not exist. To update single attributes, you must use the `PATCH` request. In addition, if the value of the property `IsConfirmationRequired` is set to true, a double opt-in is executed.

### Sample Code

```

--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0010-0001
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingPermissions (CorporateAccountID='A98979992', CorporateAccountOrigin=
'SAP_C4C_BUPA', CorporateAccountPermissionID='info@company.de', CorpAcctPermi
ssionOrigin='EMAIL', MarketingArea='', CommunicationMedium='EMAIL') HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.002'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
  {
    "PermissionUTCDateTime" : "2019-07-01T13:04:46.201",
    "PermissionGranted" : "Y",
    "PermissionSourceCommMedium" : "WEB",
    "IsConfirmationRequired" : false,
    "PermissionNoteText" : "Sample Permission"
  }
--changeset_01869434-0010-0001
content-type: application/http
content-transfer-encoding: binary
PUT
MarketingSubscriptions (CorporateAccountID='A98979992', CorporateAccountOrigin=
'SAP_C4C_BUPA', CorporateAccountSubscriptionID='info@company.de', CorpAcctS
ubscriptionOrigin='EMAIL', CommunicationMedium='EMAIL', SubscriptionTopic='1'
) HTTP/1.1
Accept: application/json
Sap-Cuan-RequestTimestamp: '2019-07-01T13:04:46.005'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: C4C
sap-cuan-referenceid: REQ1
Content-Type: application/json
Content-Length: 1021
  {
    "SubscriptionUTCDateTime" : "2019-07-01T13:04:46.201",
    "SubscriptionSignUpExists" : "N",
    "SubscriptionSourceCommMedium" : "WEB",
    "IsConfirmationRequired" : false,
    "SubscriptionNoteText" : "Sample Subscription"
  }

```

```
}  
--changeset_01869434-0010-0001--  
--batch--
```

## GET Requests

### Get all explicit marketing permissions for a specific CorporateAccountUUID

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingPermissions?  
$filter=CorporateAccountUUID eq guid'6c0b84b7-5523-1ed9-a791-f00f93927b51' and  
PermissionIsImplicit eq false&$top=10
```

### Get all marketing permissions and marketing subscriptions for a corporate account with a certain ID and origin

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/  
CorporateAccountOriginData(CorporateAccountID='A98979992'  
,CorporateAccountOrigin='SAP_C4C_BUPA')?  
$expand=MarketingPermissions,MarketingSubscriptions
```

### Get all marketing permissions and marketing subscriptions for a CorporateAccountUUID

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/  
CorporateAccounts(CorporateAccountUUID=guid'6c0b84b7-5523-1ed9-a791-f00f93927b51')?  
$expand=MarketingPermissions,MarketingSubscriptions
```

### Get all marketing permissions for a corporate account with a certain ID and origin

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/  
CorporateAccountOriginData(CorporateAccountID='A98979992'  
,CorporateAccountOrigin='SAP_C4C_BUPA')/MarketingPermissions
```

### Get corporate account data via ID and origin together with its marketing permissions and marketing subscriptions

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/  
CorporateAccountOriginData(CorporateAccountID='A98979992'  
,CorporateAccountOrigin='SAP_C4C_BUPA')/MarketingSubscriptions
```

### Get a corporate account via CorporateAccountUUID together with its marketing permissions and marketing subscriptions

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingSubscriptions?  
$filter=CorporateAccountUUID eq guid'6c0b84b7-5523-1ed9-a791-f00f93927b51'&$top=20
```

### Get all marketing permissions for a specific email address of a corporate account

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingPermissions?  
$filter=CorporateAccountPermissionID eq 'info@company.de' and  
CorpAcctPermissionOrigin eq 'EMAIL' &$top=20
```

## Get the first 500 a corporate accounts that subscribed to newsletter Fashion

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingSubscriptions?$top=500&$filter=SubscriptionTopicName eq 'Fashion'
```

## Get the first 100 marketing permissions that are newer than a certain date and time

```
/sap/opu/odata/sap/API_MKT_CORPORATE_ACCOUNT_SRV;v=0003/MarketingPermissions?$top=100&$filter=PermissionUTCDateTime gt datetimeoffset'2019-01-01T00:00:00.001'
```

## Account Team Members

### PUT

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT AccountTeamMembers(CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>',TeamMemberID='<TeamMemberID>'
,Role='<Role>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### DELETE

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE AccountTeamMembers(CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>',TeamMemberID='<TeamMemberID>'
,Role='<Role>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH AccountTeamMembers(CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>',TeamMemberID='<TeamMemberID>
',Role='<Role>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-02T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
"CorporateAccountID": "<CorporateAccountID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Additional IDs

## PUT

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT AdditionalIDs(CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>',InteractionContactAdditional
Origin='<InteractionContactAdditionalOrigin>',InteractionContactAdditionalExte
rnalID='<InteractionContactAdditionalExternalID>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## PATCH

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH AdditionalIDs (CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>', InteractionContactAdditional
Origin='<InteractionContactAdditionalOrigin>', InteractionContactAdditionalExte
rnalID='<InteractionContactAdditionalExternalID>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "CorporateAccountID": "<CorporateAccountID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Corporate Account Origin Data

### PUT - Batch

### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT CorporateAccountOriginData (CorporateAccountID='<AccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### PATCH - Batch

### Sample Code

```
--batch
```

```

Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH CorporateAccountOriginData(CorporateAccountID='<AccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-28T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "OriginDataLastChgUTCDateTime": "2017-10-01T13:13:14",
  "StreetName": "<StreetName>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## PUT Single Entity (use the same request header attributes as for batch)

### i Note

When you import single entities, the response body is empty. You can read the status of the import only in the **response header** in the attributes **Status** and **Sap-Message**.

### ≡ Sample Code

```

Request: PUT: /sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0003/
CorporateAccountOriginData(CorporateAccountID='C_20180828_00008',CorporateAccountOrigin='SAP_ERP_CUSTOMER')
{
  "OriginDataLastChgUTCDateTime" : "2017-10-01T13:13:14Z",
  "CityName" : "Walldorf",
  "Country" : "DE",
  "EmailAddress" : "info.germany@sap.de",
  "FullName" : "SAP SE",
  "AddressHouseNumber" : "16",
  "Language" : "EN",
  "PhoneNumber" : "+496227747474",
  "ContactPostalCode" : "69190",
  "StreetName" : "Dietmar-Hopp-Allee"
}

```

## Marketing Attributes

### PUT

### ≡ Sample Code

```

--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http

```



```

content-transfer-encoding: binary
PUT MarketingAttributes (CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>',MarketingAttributeCategory='
<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValu
e>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## DELETE

### Sample Code

```

--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE MarketingAttributes (CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>',MarketingAttributeCategory='
<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValu
e>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## PATCH

### Sample Code

```

--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH MarketingAttributes (CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>',MarketingAttributeCategory='
<MarketingAttributeCategory>',MarketingAttributeValue='<MarketingAttributeValu
e>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-09-29T12:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
}

```

```
"CorporateAccountID": "<CorporateAccountID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Marketing Areas

### PUT

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PUT MarketingAreas (CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>', InteractionContactMktgArea='
<InteractionContactMktgArea>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### PATCH

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH MarketingAreas (CorporateAccountID='<CorporateAccountID>',
CorporateAccountOrigin='<CorporateAccountOrigin>', InteractionContactMktgArea='
<InteractionContactMktgArea>') HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2017-10-01T13:13:14'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Sap-Cuan-SequenceId: UpdatePatch
Content-Type: application/json
{
  "CorporateAccountID": "<CorporateAccountID>"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

Parent topic: [Corporate Accounts \[page 512\]](#)

## Related Information

[Basic Concepts \[page 515\]](#)

[Structure of API\\_MKT\\_CORPORATE\\_ACCOUNT \[page 517\]](#)

[Function Imports \[page 551\]](#)

### 5.2.3.4 Function Imports

Function imports are used to perform custom operations on an entity, which are typically not provided by OData operations. This section contains payload examples for the following function imports:

- [Delete Marketing Area \[page 551\]](#)
- [Delete All Marketing Areas from Origin \[page 552\]](#)
- [Delete Account Team Members \[page 552\]](#)
- [Delete Marketing Attribute \[page 553\]](#)
- [Delete Additional IDs \[page 554\]](#)

#### Delete Marketing Area

HTTP Method	Function Import
<i>POST</i>	<code>CorporateAccountDeleteMarketingArea</code>  Deletes all occurrences of a marketing area from a corporate account.

#### Payload Example

##### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST CorporateAccountDeleteMarketingArea?
CorporateAccountID='DEV_TEST'&CorporateAccountOrigin='SAP_ERP_CONTACT'&Interac
tionContactMktgArea='GLOBAL' HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
```

```
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--  
--batch--
```

## Delete All Marketing Areas from Origin

HTTP Method	Function Import
<i>POST</i>	CorpAcctOriginDeleteAllMktgAreas  Deletes all marketing areas from one origin.

### Payload Example

#### Sample Code

```
--batch  
Content-Type: multipart/mixed;  
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a  
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a  
content-type: application/http  
content-transfer-encoding: binary  
POST CorporateAccountOriginDeleteAllMktgAreas?  
CorporateAccountID='DEV_TEST'&CorporateAccountOrigin='SAP_ERP_CONTACT'  
HTTP/1.1  
Content-Length: 1035  
Accept: application/json  
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'  
Sap-Cuan-SourceSystemType: EXT  
Sap-Cuan-SourceSystemId: HYBRIS  
Content-Type: application/json  
{  
}  
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--  
--batch--
```

## Delete Account Team Members

HTTP Method	Function Import
<i>POST</i>	CorpAcctDeleteAllAccountTeamMembers  Deletes all account team members from one corporate account.

### Payload Example

#### Sample Code

```
--batch  
Content-Type: multipart/mixed;  
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a  
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
```

```

content-type: application/http
content-transfer-encoding: binary
POST CorporateAccountDeleteAllAccountTeamMembers?
CorporateAccountID='DEV_TEST'&CorporateAccountOrigin='SAP_ERP_CONTACT'
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Delete Marketing Attribute

HTTP Method	Function Import
<i>POST</i>	CorpAcctOriginDeleteAllMktgAttributes Deletes all marketing attributes for one origin.

### Payload Example

#### Sample Code

```

--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST CorpAcctOriginDeleteAllMktgAttributes?
CorporateAccountID='DEV_TEST'&CorporateAccountOrigin='SAP_ERP_CONTACT'
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Delete Additional IDs

HTTP Method	Function Import
POST	CorpAcctOriginDeleteAdditionalIDs  Deletes all additional IDs from one origin except the IDs that come from the origin data.

### Payload Example

#### Sample Code

```
--batch
Content-Type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST CorpAcctOriginDeleteAdditionalIDs?
CorporateAccountID='DEV_TEST'&CorporateAccountOrigin='SAP_ERP_CONTACT'
HTTP/1.1
Content-Length: 1035
Accept: application/json
Sap-Cuan-RequestTimestamp: '2018-11-02T09:19:12'
Sap-Cuan-SourceSystemType: EXT
Sap-Cuan-SourceSystemId: HYBRIS
Content-Type: application/json
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### Function Import Parameters

Property	Description	Edm Core Type	Max Length	Mandatory	Key
CorporateAccountID	ID of Corporate Account	Edm. String	255	X	X
CorporateAccountOrigin	Origin of Corporate Account	Edm. String	20	X	X
CorporateAccountMktgArea	Marketing Area	Edm. String		X	X

Parent topic: [Corporate Accounts \[page 512\]](#)

## Related Information

[Basic Concepts \[page 515\]](#)

[Structure of API\\_MKT\\_CORPORATE\\_ACCOUNT \[page 517\]](#)

[Payload Examples for Corporate Accounts \[page 538\]](#)

## 5.2.4 Business Partners from SAP Cloud for Customer

Import business partners from SAP Cloud for Customer via CUAN\_BUSINESS\_PARTNER\_IMP\_SRV to marketing.

[Overview \[page 555\]](#)

[Technical Prerequisites \[page 556\]](#)

[Basic Concepts \[page 556\]](#)

[Structure of OData Service CUAN\\_BUSINESS\\_PARTNER\\_IMP\\_SRV \[page 558\]](#)

[Payload Examples for CUAN\\_BUSINESS\\_PARTNER\\_IMP\\_SRV \[page 566\]](#)

This section contains payload examples for replicating business partners from SAP Cloud for Customer to marketing with the OData service CUAN\_BUSINESS\_PARTNER\_IMP\_SRV.

### 5.2.4.1 Overview

OData service CUAN\_BUSINESS\_PARTNER\_IMP\_SRV is used for standard SAP Marketing Cloud integration with SAP Cloud for Customer. It is used to replicate SAP Cloud for Customer business partners to SAP Marketing Cloud interaction contacts. For details of standard SAP Marketing Cloud integration with SAP Cloud for Customer see [Technical Prerequisites \[page 556\]](#).

**Parent topic:** [Business Partners from SAP Cloud for Customer \[page 555\]](#)

## Related Information

[Technical Prerequisites \[page 556\]](#)

[Basic Concepts \[page 556\]](#)

[Structure of OData Service CUAN\\_BUSINESS\\_PARTNER\\_IMP\\_SRV \[page 558\]](#)

[Payload Examples for CUAN\\_BUSINESS\\_PARTNER\\_IMP\\_SRV \[page 566\]](#)

## 5.2.4.2 Technical Prerequisites

OData service `CUAN_BUSINESS_PARTNER_IMP_SRV` is available as part of standard integration with SAP Cloud for Customer.

For the setup you have the following two options:

1. You are guided by the Cloud Integration Automation Service (see the SAP Help Portal at [Introduction to Cloud Integration Automation Service](#)) by making use of the [Maintenance Planner](#).
2. You use the integration setup of SAP Cloud for Customer with SAP Marketing Cloud (see [SAP Cloud for Customer Integration with SAP Marketing](#), or [Setting Up SAP Marketing Cloud Integration with SAP Cloud for Customer \(1J9\)](#)).

**Parent topic:** [Business Partners from SAP Cloud for Customer \[page 555\]](#)

### Related Information

[Overview \[page 555\]](#)

[Basic Concepts \[page 556\]](#)

[Structure of OData Service `CUAN\_BUSINESS\_PARTNER\_IMP\_SRV` \[page 558\]](#)

[Payload Examples for `CUAN\_BUSINESS\_PARTNER\_IMP\_SRV` \[page 566\]](#)

## 5.2.4.3 Basic Concepts

### Calling the OData Service

To import persons, companies, or relationships, a deep insert on entity `Import_Headers` with HTTP method `POST` has to be performed; other methods like create, update, or delete on any other entity are not supported. You find code snippets under [Structure of OData Service `CUAN\_BUSINESS\_PARTNER\_IMP\_SRV` \[page 558\]](#) and the subordinated chapters.

#### i Note

We recommend that you do not use batch processing (`$batch`) because error handling is more complex with batch processing. A batch request can return an OK code and still have errors that have to be checked in the response body. It is still possible to send multiple entities in one POST request without using batch processing.



## Id and IdOrigin

The `Id` and the `IdOrigin` define the external key of an interaction contact in the source system. The `IdOrigin` indicates the source origin of the ID. It is either defined and delivered by SAP or can be maintained in the Self-Service Configuration app [Define Origins of Contact ID](#). The attribute `InternalId` defines the ID of the company or person in SAP Marketing Cloud. The `InternalId` corresponds to a GUID of `IdOrigin` `SAP_HYBRIS_MKT_IC`.

## Update Behavior

The source system must always provide a whole snapshot of the object as the system of SAP Marketing Cloud always updates a complete person, company or relationship; partial update of an entity is not supported. This means that it is not possible to update only two attributes of a person, for example, as the empty attributes of the update would overwrite the existing attribute entries.

If a facet or marketing area is not provided anymore in the update case the system sets this facet or marketing area to obsolete.

## Sequence Handling with Attribute LastChangeDate

Attribute `LastChangeDate` of the entities `Company`, `Person`, or `Relationship` defines the last change in the source system. The timestamp is used to ensure that different requests are processed in the correct sequence. An incoming request is always validated against the last saved timestamp for the respective entity and checked, whether the provided `LastChangeDate` is newer than the saved one. Requests which contain an outdated `LastChangeDate` are discarded as a more recent snapshot of the entity has already been saved.

## Code Values

Attributes based on codes can only process valid code values. If a request provides an invalid code value the request will result in an error. The error can be checked in the [Import Monitor](#) app. For more information, see [Import Monitor \[page 404\]](#).

## Error Handling

Technical errors are returned to the sender with the corresponding HTTP error code.

Application inbound errors are recorded in the system of SAP Marketing Cloud and can be monitored, restarted, and discarded in the [Import Monitor](#) app. For more information, see [Import Monitor \[page 404\]](#).

## i Note

If you encounter issues with the OData service `CUAN_BUSINESS_PARTNER_IMP_SRV`, create a support ticket under component `CEC-MKT-DM-IC` (Interaction Contacts).

The component is not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

## Extensibility

All custom fields registered with the *Custom Fields* app will automatically appear in the OData service. It is not necessary to enable the usage for a specific field for that OData service.

The entity `Company` has the following business contexts assigned: `Marketing: Corporate Account` and `Marketing: Contact` and `Corporate Account`.

The entity `Person` has the following business context assigned: `Context Marketing: Contact` and `Marketing: Contact` and `Corporate Account`.

The entity `Relationship` is not extensible.

**Parent topic:** [Business Partners from SAP Cloud for Customer \[page 555\]](#)

## Related Information

[Overview \[page 555\]](#)

[Technical Prerequisites \[page 556\]](#)

[Structure of OData Service `CUAN\_BUSINESS\_PARTNER\_IMP\_SRV` \[page 558\]](#)

[Payload Examples for `CUAN\_BUSINESS\_PARTNER\_IMP\_SRV` \[page 566\]](#)

### 5.2.4.4 Structure of OData Service `CUAN_BUSINESS_PARTNER_IMP_SRV`

The `CUAN_BUSINESS_PARTNER_IMP_SRV` OData service consists of the following entity sets and entity types:

Entity Set	Entity Type	Entity Description
ImportHeaders	ImportHeader	Technical Import Message Header
Companies	Company	Company

Entity Set	Entity Type	Entity Description
Persons	Person	Person
Relationships	Relationship	Relationship (from contact person to corporate account)
Facets	Facet	Facet
MarketingAreas	MarketingArea	Marketing Area

The metadata structure of the service is read by means of the OData call:

- **Request URI:** `https://<server>:<port> /sap/opu/odata/sap/CUAN_BUSINESS_PARTNER_IMP_SRV/$metadata`
- **HTTP Method:** GET

Parent topic: [Business Partners from SAP Cloud for Customer \[page 555\]](#)

## Related Information

[Overview \[page 555\]](#)

[Technical Prerequisites \[page 556\]](#)

[Basic Concepts \[page 556\]](#)

[Payload Examples for CUAN\\_BUSINESS\\_PARTNER\\_IMP\\_SRV \[page 566\]](#)

### 5.2.4.4.1 ImportHeader

The entity type `ImportHeader` describes the technical header of an import of multiple business partners. The properties `Id` and `Timestamp` are used for logging the external data request. If an error occurs during the posting of the business partner, in addition to the import header data the error message and the failed record are saved. This data can be checked with the *Import Monitor* app. For more information, see [Import Monitor \[page 404\]](#).

For every service request, a new, unique ID is required. If no ID value is provided it is defaulted internally. In the *Import Monitor* app, the ID is used as search field and the status of the import is shown for the request. If no timestamps is provided it is defaulted to the time of import processing. If the timestamp is provided it is stored at the import header and displayed in the *Import Monitor* as date/time of the import notification.

The `SourceSystemId` and `SourceSystemType` property allows you to distinguish between different source systems. The `SourceSystemId` and `SourceSystemType` are mandatory attributes.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
Id	Unique technical identifier of import run.	Edm.String	32	X	X
Timestamp	Timestamp of the run in the format: number of milliseconds since midnight Jan 1, 1970. For example: /Date(1406014140922)/	Edm.DateTime	0		
SourceSystem-Type	Type of the source system, such as C4C	Edm.String	20	X	
SourceSystemId	Identifier of the source system	Edm.String	20	X	

## 5.2.4.4.2 Company

The entity type `Company` contains all attributes that are required to create a corporate account with its main origin data. The ID of the company has to be provided by the external source system to perform later updates.

This entity is used in standard SAP Marketing Cloud integration with SAP Cloud for Customer for replicating accounts.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
Id	Id of the Company in the external system	Edm.String	100	X	X
IdOrigin	Origin or source of ID of companies from external systems	Edm.String	20	X	X

Property	Description	Edm Core Type	Max Length	Mandatory	Key
InternalId	ID of the company on SAP Marketing Cloud integration with SAP Cloud for Customer	Edm.String	100	X	
LastChangeDate	Timestamp  Format: number of milliseconds since midnight Jan 1, 1970. For example: /Date(1406014140922)/	Edm.DateTime	0	X	
CompanyName	Company Name	Edm.String	80		
CountryCode	Country Code	Edm.String	3		
RegionCode	Region Code	Edm.String	3		
CityName	City Name as part of the postal address	Edm.String	40		
PostalCode	Postal Code	Edm.String	10		
Street	Street as part of postal address	Edm.String	60		
HouseNumber	House number as part of postal address	Edm.String	10		
EmailAddress	E-mail Address	Edm.String	241		
PhoneNumber	Phone number for mobile  Format: +country code region code + phone number, such as +49151123456	Edm.String	30		

Property	Description	Edm Core Type	Max Length	Mandatory	Key
FaxNumber	Fax number  Format: +country code region code + fax number suuch as +49 6227 123456	Edm.String	30		
WebSite	Web URI	Edm.String	1.000		
IndustryCode	Industry Code	Edm.String	4		
LanguageCode	Preferred commu- nication Language Code	Edm.String	2		
MobileNumber	Mobile Number	Edm.String	30		
Latitude	Latitude	Edm.Decimal (prec. 20; scale)	10		
Longitude	Longitude	Edm.Decimal (prec. 20; scale)	10		
SpatialReference- System	Spatial Reference System	Edm.String	30		

### 5.2.4.4.3 Person

This entity is used in standard SAP Marketing Cloud integration with SAP Cloud for Customer for replicating contacts and individual customers.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ID	ID of the person in the external sys- tem	Edm.String	100	X	X
IdOrigin	Origin or source of ID of person from external systems	Edm.String	20	X	X

Property	Description	Edm Core Type	Max Length	Mandatory	Key
InternalID	ID of the person in SAP Marketing Cloud integration with SAP Cloud for Customer	Edm.String	100	X	
LastChangeDate	Obsolete must be filled for compatibility reasons	Edm.DateTime	0	X	
FirstName	First Name	Edm.String	40		
LastName	Last Name	Edm.String	40		
FullName	Full Name	Edm.String	80		
DateOfBirth	Date of Birth	Edm.DateTime	0		
GenderCode	Gender Code	Edm.String	1		
MaritalStatusCode	Marital Status Code	Edm.String	1		
TitleCode	Title Code	Edm.String	4		
WebSite	Web URI	Edm.String	1.000		
IsConsumer	Person is Consumer	Edm.Boolean	0		
IsContact	Person is Contact	Edm.Boolean	0		
LanguageCode	Preferred Language	Edm.String	2		
PostalCode	Postal Code as part of the address; only relevant for consumer	Edm.String	10		
Street	Street as part of the address; only relevant for consumer	Edm.String	60		
HouseNumber	House Number as part of the address; only relevant for consumer	Edm.String	10		

Property	Description	Edm Core Type	Max Length	Mandatory	Key
EmailAddress	Email Address; only relevant for consumer	Edm.String	241		
PhoneNumber	Phone Number; only relevant for consumer	Edm.String	30		
MobilePhoneNum- ber	Mobile Phone Number; only rele- vant for consumer	Edm.String	30		
FaxNumber	Fax Number; only relevant for con- sumer	Edm.String	30		
MobileNumber	Mobile Number	Edm.String	30		
Latitude	Latitude	Edm.Decimal (prec. 20; scale)	10		
Longitude	Longitude	Edm.Decimal (prec. 20; scale)	10		
SpatialReference- System	Spatial Reference System	Edm.String	30		

#### 5.2.4.4.4 Relationship

This entity is used in standard SAP Marketing Cloud integration with SAP Cloud for Customer for replicating `IsContactPersonFor` relationships. The relationship type has to be provided. The relationship is directed from a contact person to a corporate account.

It is possible to create or to delete a relationship by providing the corresponding `ActionCode`.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
IdFrom	External ID of con- tact person	Edm.String	100	X	X
IdTo	External ID of cor- porate account	Edm.String	100	X	X



Property	Description	Edm Core Type	Max Length	Mandatory	Key
IdOrigin	Origin or source of ID of From and To Business Partner	Edm.String	20	X	X
InternalIdFrom	ID of the contact person in SAP Marketing Cloud	Edm.String	100		
InternalIdTo	ID of the corporate account in SAP Marketing Cloud	Edm.String	100		
RelationType	Type of the relationship	Edm.String	2	X	
ActionCode	The action code controls how the relationship is to be processed in the backend:  1. Create/Change 2. Delete	Edm.String	1	X	
LastChangeDate	Timestamp of the external object. Timestamp is used to process messages in the right sequence	Edm.DateTime	0		
EmailAddress	Business Email Address of contact person	Edm.String	241		
PhoneNumber	Business Phone Number	Edm.String	30		
MobilePhoneNumber	Business Mobile Phone Number	Edm.String	30		
FaxNumber	Business Fax Number	Edm.String	30		
DepartmentCode	Department	Edm.String	4		
FunctionCode	Function	Edm.String	4		

Property	Description	Edm Core Type	Max Length	Mandatory	Key
IsMain	Is main Contact	Edm.Boolean			

### 5.2.4.4.5 Facet

Facets can be used to import additional external IDs of a company or a person.

If the facets are used to import those IDs the additional IDs have to be provided in the entity `Relationship` for consistency reasons. For more information, see section *Update Behavior* in [Basic Concepts \[page 556\]](#).

Property	Description	Edm Core Type	Max Length	Mandatory	Key
Id	ID from the external system	Edm.String	10	X	X
Origin	Origin of ID from external system	Edm.String	30	X	X

### 5.2.4.4.6 Marketing Area

Entity `Marketing Area` can be used to import marketing area assignments for a person or a company.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
MarketingAreald	ID of the Marketing Area	Edm.String	40	X	X

## 5.2.4.5 Payload Examples for CUAN\_BUSINESS\_PARTNER\_IMP\_SRV

This section contains payload examples for replicating business partners from SAP Cloud for Customer to marketing with the OData service `CUAN_BUSINESS_PARTNER_IMP_SRV`.

**Parent topic:** [Business Partners from SAP Cloud for Customer \[page 555\]](#)

## Related Information

[Overview \[page 555\]](#)

[Technical Prerequisites \[page 556\]](#)

[Basic Concepts \[page 556\]](#)

[Structure of OData Service CUAN\\_BUSINESS\\_PARTNER\\_IMP\\_SRV \[page 558\]](#)

### 5.2.4.5.1 Importing Company Data

All attributes of entity `Company` are relevant for updating a company including all extension fields of business context `Marketing: Corporate Account` and `Marketing: Contact and Corporate Account`.

The following code snippet shows an example in JSON format of how to import two corporate accounts, one of them with an additional ERP ID:

- Request URI:  
`/sap/opu/odata/sap/CUAN_BUSINESS_PARTNER_IMP_SRV/ImportHeaders`
- HTTP Method: POST

#### Sample Code

```
{
  "Id": "",
  "Timestamp": "2016-05-04T14:07:21.6779610",
  "SourceSystemType": "C4C",
  "SourceSystemId": "CLOUDFORCUSTOMER",
  "Companies": [
    {
      "Id": "123456789",
      "IdOrigin": "SAP_C4C_BUPA",
      "LastChangeDate": "2016-05-04T14:07:21.6779610",
      "CompanyName": "SAP Deutschland SE & Co. KG",
      "CountryCode": "DE",
      "RegionCode": "BW",
      "CityName": "Walldorf",
      "PostalCode": "69190",
      "Street": "Hasso-Plattner-Ring",
      "HouseNumber": "7",
      "EmailAddress": "info.germany@sap.com",
      "PhoneNumber": "+496227747474",
      "FaxNumber": "+496227757575",
      "WebSite": "www.sap.com/germany",
      "IndustryCode": "63",
      "LanguageCode": "DE",
      "Facets": [
        {
          "Id": "123456789",
          "IdOrigin": "SAP_ERP_BUPA"
        }
      ]
    },
    {
      "Id": "923456789",
      "IdOrigin": "SAP_C4C_BUPA",
      "LastChangeDate": "2016-05-04T14:07:21.6779610",
      "CompanyName": "OtherCompany",
```

```
        "CountryCode": "DE",
        "RegionCode": "BW"
    }
]
}
```

During the import of companies, the system determines whether there are relationships to that corporate account. If there are contact persons with an `Is Contact Person` relation to that account the corporate account's postal address (street, house number, postal code, city name, region code, country) is copied to all active contact persons. After the import of the corporate account, all contact persons of that account have the same postal address.

## 5.2.4.5.2 Importing Person Data

During the import of persons via the OData service `CUAN_BUSINESS_PARTNER_IMP_SRV`, a person can be either a consumer (B2C process) or a contact person (B2B process).

- A consumer is a natural person who generates sales revenue (`IS_CONSUMER = X`).
- A contact person is a natural person a company interacts in a B2B process (`IS_CONTACT = X`) with, and which usually has an `Is Contact Person` relationship to a corporate account. A contact person has workplace-related information, such as function, department or workplace communication data (phone, email, fax etc). A contact person's postal address is derived from the related company's postal address.

Dependent on the attribute `IsConsumer` or `IsContact`, a different set of attributes is used for updating the interaction contact. A contact person may only have the related company's postal address. That is why importing a postal address for a person marked as `IsContact` is not possible, and if attributes are provided they are ignored during import.

The following table provides you with attributes that are relevant for updating a consumer or contact person.

Person (consumer)

- Id
  - IdOrigin
  - InternalId
  - LastChangeDate
  - FirstName
  - LastName
  - FullName
  - DateOfBirth
  - GenderCode
  - MaritalStatusCode
  - TitleCode
  - CountryCode
  - RegionCode
  - CityName
  - PostalCode
  - Street
  - HouseNumber
  - EmailAddress
  - PhoneNumber
  - MobilePhoneNumber
  - FaxNumber
  - WebSite
  - IsConsumer
  - LanguageCode
  - Customer extension fields for Business Context  
Marketing: Contact and Marketing: Contact  
and Corporate Account
-

Person (Contact Person)

- Id
- IdOrigin
- InternalId
- LastChangeDate
- FirstName
- LastName
- FullName
- DateOfBirth
- GenderCode
- MartialStatusCode
- TitleCode
- WebSite
- IsContact
- LanguageCode
- Customer extension fields for Business Context  
Marketing: Contact and Marketing: Contact  
and Corporate Account

---

## 5.2.4.5.2.1 Consumer

The following code snippet shows an example in JSON format of how to import a consumer with an additional ERP ID:

- Request URI:  
`/sap/opu/odata/sap/CUAN_BUSINESS_PARTNER_IMP_SRV/ImportHeaders`
- HTTP Method: POST

### Sample Code

```
{
  "Id": "",
  "Timestamp": "2016-05-04T14:07:21.6779610",
  "SourceSystemType": "C4C",
  "SourceSystemId": "CLOUDFORCUSTOMER",
  "Persons": [
    {
      "Id": "223456789",
      "IdOrigin": "SAP_C4C_BUPA",
      "LastChangeDate": "2016-05-04T14:07:21.6779610",
      "FirstName": "Erika",
      "LastName": "Mustermann",
      "FullName": "Erika Mustermann",
      "GenderCode": "2",
      "MaritalStatusCode": "1",
      "CountryCode": "DE",
      "RegionCode": "BW",
      "CityName": "Walldorf",
      "PostalCode": "69190",
      "Street": "Bahnhofstraße",
      "HouseNumber": "1",
      "EmailAddress": "erika.mustermann@privat.de",
```

```

        "LanguageCode": "DE",
        "MobilePhoneNumber": "49119201412192",
        "PhoneNumber": "49119201412191",
        "TitleCode": "0001",
        "IsConsumer": true,
        "Facets": [{
            "Id": "223456789",
            "IdOrigin": "SAP_ERP_BUPA"
        }]
    }
]
}

```

As a result of this sample request, a consumer will be created with the provided attributes.

## 5.2.4.5.2.2 Contact Person

The following code snippet shows an example in JSON format of how to import a contact person with additional ERP ID:

- Request URI:  
/sap/opu/odata/sap/CUAN\_BUSINESS\_PARTNER\_IMP\_SRV/ImportHeaders
- HTTP Method: POST

### Sample Code

```

{
  "Id": "",
  "Timestamp": "2016-05-04T14:07:21.6779610",
  "SourceSystemType": "C4C",
  "SourceSystemId": "CLOUDFORCUSTOMER",
  "Persons": [
    {
      "Id": "323456789",
      "IdOrigin": "SAP_C4C_BUPA",
      "LastChangeDate": "2016-05-04T14:07:21.6779610",
      "FirstName": "Heinz",
      "LastName": "Müller",
      "FullName": "Heinz Müller",
      "GenderCode": "1",
      "MaritalStatusCode": "2",
      "TitleCode": "0002",
      "DateOfBirth": "1978-05-12T00:00:00.0000000",
      "IsContact": true,
      "LanguageCode": "DE",
      "Facets": [{
        "Id": "323456789",
        "IdOrigin": "SAP_ERP_BUPA"
      }]
    }
  ]
}

```

The result of this sample request depends on the data already in the system.

If the contact person is created with this sample request or if the contact person of the request does not exist in the system, the contact person will be created and the contact person will only contain the attributes listed in the request.

If the contact person already has a relationship to an account the contact person will additionally have the following attributes:

- The corporate account's postal address
- All attributes provided with the relationship request

## 5.2.4.5.3 Importing Relationship Data

### 5.2.4.5.3.1 Create or Change Relationship

The following code snippet shows an example in JSON format of how to import a relationship between a company and a contact with an additional ERP ID. The contact person's additional IDs are transferred via the entity `FromFacets`:

- Request URI:  
`/sap/opu/odata/sap/CUAN_BUSINESS_PARTNER_IMP_SRV/ImportHeaders`
- HTTP Method: POST

#### Sample Code

```
{
  "Id": "",
  "Timestamp": "2016-05-04T14:07:21.6779610",
  "SourceSystemType": "C4C",
  "SourceSystemId": "CLOUDFORCUSTOMER",
  "Relationships": [
    {
      "IdOrigin": "SAP_C4C_BUPA",
      "IdFrom": "323456789",
      "IdTo": "123456789",
      "RelationType": "01",
      "ActionCode": "1",
      "LastChangeDate": "2016-05-04T14:07:21.6779610",
      "FunctionCode": "08",
      "DepartmentCode": "0024",
      "PhoneNumber": "+496227712345",
      "FaxNumber": "+4962277612345",
      "EmailAddress": "heinz.mueller@sap.com",
      "FunctionCode": "08",
      "DepartmentCode": "0025",
      "IsMain": false,
      "FromFacets": [
        {
          "Id": "323456789",
          "IdOrigin": "SAP_ERP_BUPA"
        }
      ]
    }
  ]
}
```



The result of that request depends on the data already in the system:

- If the contact person is created with this sample request the contact person will contain only the attributes listed in the request and will have no name, title, date of birth, and so on. If the related company has a postal address the contact person will also have that postal address.
- If the company of that sample request does not exist the request will not be saved but forwarded to the [Import Monitor](#) app. The request will be automatically reprocessed until the company is successfully imported into the system. Usually, a source system will only allow a relationship to be created if the referenced business partner has been created, so the relationship might reach SAP Marketing Cloud before the corresponding master data requests.
- If the contact person is already in the system and a relationship to a company with postal address exists the contact person will have the attributes sent with the relationship request, the company's postal address and the contact person's master data.

### 5.2.4.5.3.2 Delete Relationship

The following code snippet shows an example in JSON format of how to delete an existing relationship between a company and a contact:

- Request URI:  
`/sap/opu/odata/sap/CUAN_BUSINESS_PARTNER_IMP_SRV/ImportHeaders`
- - HTTP Method: POST

#### Sample Code

```
{
  "Id": "",
  "Timestamp": "2017-08-10T14:07:21.6779610",
  "SourceSystemType": "C4C",
  "SourceSystemId": "CLOUDFORCUSTOMER",
  "Relationships": [
    {
      "IdOrigin" : "SAP_C4C_BUPA",
      "IdFrom" : "323456789",
      "IdTo" : "123456789",
      "RelationType" : "01",
      "ActionCode" : "2",
      "LastChangeDate" : "2017-08-10T14:07:21.6779610"
    }
  ]
}
```

As a result of this request, the contact person's relationship to the company is deleted. Along with that, the following business related attributes are cleared:

- Postal Address
- Communication data (phone, email, fax, mobile)
- Function, department

## 5.2.5 Import Business Partners

CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV for importing business partner data from external source systems, like, for example, SAP ERP, SAP CRM, SAP S/4HANA On Premise.

OData service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV is used for standard SAP Marketing Cloud integration. It is used in marketing-driven and sales-driven processes to replicate data from SAP ERP, SAP CRM, or SAP S/4HANA On Premise to SAP Marketing Cloud interaction contacts.

For more information about the integration scenario, see [Integration with SAP ERP \[page 349\]](#) and [Order Management Data Replication to SAP Marketing Cloud \[page 348\]](#).

For more information about external interfaces that SAP Marketing Cloud provides for creating or updating interaction contacts, interactions, interests, corporate accounts, product categories, and products, see [Integration APIs \[page 387\]](#).

OData service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV can also be used to create SAP Marketing Cloud interaction contacts from any source system. The OData service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV supports the change of interaction contacts. Each interaction contact is identified by the key of the business partner in the external system.

### i Note

If you encounter issues with the OData service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV, create a support ticket under component CEC-MKT-DM-IC (Interaction Contacts).

The component is not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

### 5.2.5.1 Technical Prerequisites

OData service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV is available as part of the standard integration with SAP ERP, SAP S/4HANA Cloud, and SAP S/4HANA On Premise.

### 5.2.5.2 Basic Concepts

OData service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV supports only batch processing. Within a batch request only the operation PATCH (MERGE) on the entity type InteractionContact, or the operation POST on the entity type MarketingAttribute, or the function import DeleteMarketingAttributes are supported. Other operations, such as update or read are not supported.

Batch requests allow grouping multiple operations into a single HTTP request payload.

Batch requests are submitted as a single HTTP POST request to the \$batch endpoint of a service as described in [OData URI](#) . The batch request must contain a content-type header specifying a content type of **multipart/mixed** and a boundary specification.

A PATCH (MERGE) request updates only the properties indicated in the request body and leaves everything untouched that was not mentioned.

All properties that are not to be changed, can be omitted. The transmitted properties are merged with the data already stored in SAP Marketing Cloud.

Any processing errors are recorded in the SAP Marketing Cloud system in the Import Monitor app, where they can be monitored, restarted and discarded.

#### **i Note**

If you encounter issues with the OData service `CUAN_BUSINESS_PARTNER_IMPORT_SRV`, create a support ticket under component `CEC-MKT-DM-IC` (Interaction Contacts).

The component is not to be used for HTTP errors. For example, if the OData service is accepted by the gateway component in the SAP Marketing Cloud system, the HTTP status code 201 or 202 is returned. For more information, see [HTTP Response Status Codes \[page 408\]](#).

### **5.2.5.3 Structure of OData Service CUAN\_BUSINESS\_PARTNER\_IMPORT\_SRV**

The `CUAN_BUSINESS_PARTNER_IMPORT_SRV` OData service consists of the following entity sets and entity types:

<b>Entity Set</b>	<b>Entity Type</b>	<b>Entity Type Description</b>
<code>InteractionContacts</code>	<code>InteractionContact</code>	Interaction contacts refer to contacts in SAP Marketing Cloud. Interaction Contact is a generic term to group all natural persons (contacts, consumers, or suspects), companies and "unknowns", who interact with your company. Contact data is collected and merged from several sources into the master data tables within SAP Marketing Cloud. We distinguish between contacts, consumers and suspects to define the business relationship of a contact to a company.
<code>MarketingAttributes</code>	<code>MarketingAttribute</code>	Marketing attributes are assigned to an interaction contact. The marketing attribute category can be defined per source of the contact data.
<code>MarketingAreas</code>	<code>MarketingArea</code>	Use marketing areas as organizational units and to determine which interaction contacts a user can access.

The OData service `CUAN_BUSINESS_PARTNER_IMPORT_SRV` supports OData batch processing. Interaction contact data can be transferred by the OData `PATCH (MERGE)` operation for entity type `InteractionContact`.

Marketing attributes can be transferred by the OData `POST` operation on the entity type `InteractionContact` via the navigation parameter `MarketingAttributes`. Any operation on the `MarketingAttributes` entity set without navigation from the `InteractionContact` is not supported.

It is expected that all marketing attributes of an interaction contact are transferred via one change set within the batch request. The transmitted marketing attributes overwrite the existing entries imported from the same source.

The deletion of the marketing attributes of an interaction contact can be done via the OData function import `DeleteMarketingAttributes`.

Marketing areas can be transferred by the OData `POST` operation on the entity type `InteractionContact` via the navigation parameter `MarketingAreas`. The marketing area of the interaction contact can be deleted by the OData `DELETE` operation on the entity type `MarketingArea`.

## Request Header

The request header contains the following additional header fields:

Header Field	Description	Edm Core Type	Max Length	Mandatory
<code>Sap-Cuan-SequenceId</code>	Unique technical identifier of the imported data.	Edm.String	30	X
<code>Sap-Cuan-RequestTimestamp</code>	Timestamp of the data	Edm.DateTime	0	*
<code>Sap-Cuan-SequenceNumber</code>	Sequence number of the request. This number is normally incremented each time a new request for the same sequence id is created.	Edm.Int16	0	*
<code>Sap-Cuan-SourceSystemType</code>	Type of the source system	Edm.String	20	X
<code>Sap-Cuan-SourceSystemId</code>	Identifier of the source system. This is a free text field.	Edm.String	255	X

Header Field	Description	Edm Core Type	Max Length	Mandatory
Sap-Cuan-ExternalReferenceId	External reference of the inbound message	Edm.String	32	
Sap-Cuan-ExternalDocumentId	External identifier of the source document	Edm.String	20	

The header fields `Sap-Cuan-SequenceId` and `Sap-Cuan-RequestTimestamp` or `Sap-Cuan-SequenceNumber` are used to check the sequence of the received data. Data with a timestamp older or sequence number lower than data already imported, is ignored. Data with the same `Sap-Cuan-SequenceID` is also ignored.

The `Sap-Cuan-SourceSystemType` and `Sap-Cuan-SourceSystemId` fields allow you to distinguish between different source systems.

\*Either `Sap-Cuan-RequestTimestamp` or `Sap-Cuan-SequenceNumber` must be provided together with `Sap-Cuan-SequenceId`.

The `Sap-Cuan-ExternalReferenceId` and `Sap-Cuan-ExternalDocumentId` allow better error analysis because they contain external references to a source SOAP message and/or an IDoc.

## InteractionContact

Only the properties `Id` and `IdOrigin` are mandatory. All properties that are not to be changed can be omitted.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>Id</code>	ID of Interaction Contact	Edm.String	255	x	x
<code>IdOrigin</code>	Origin of Interaction Contact	Edm.String	20	x	x
<code>IsEndOfPurposeBlocked</code>	End of Purpose Reached	Edm.Boolean	0		
<code>Name</code>	Full Name	Edm.String	80		
<code>FirstName</code>	First Name	Edm.String	40		
<code>LastName</code>	Last Name	Edm.String	40		
<code>TitleCode</code>	Title Code	Edm.String	4		
<code>CountryCode</code>	Country Code	Edm.String	3		
<code>RegionCode</code>	Region Code	Edm.String	3		

Property	Description	Edm Core Type	Max Length	Mandatory	Key
City	City	Edm.String	40		
PostalCode	Postal Code	Edm.String	10		
Street	Street	Edm.String	60		
HouseNumber	House Number	Edm.String	10		
LanguageCode	Language Code	Edm.String	2		
GenderCode	The following fixed values are supported: <ul style="list-style-type: none"> <li>• 1 - Male</li> <li>• 2 - Female</li> <li>• 3 - Non-binary</li> <li>• 9 - Not specified</li> </ul>	Edm.String	1		
MaritalStatusCode	Marital Status Code	Edm.String	1		
IndustryCode	Industry Code	Edm.String	4		
DepartmentCode	Department Code	Edm.String	4		
FunctionCode	Function Code	Edm.String	4		
EmailAddress	E-Mail Address	Edm.String	241		
PhoneNumber	Phone Number	Edm.String	30		
MobilePhoneNumber	Mobile Phone Number	Edm.String	30		
FaxNumber	Fax Number	Edm.String	30		
DateOfBirth	Date of Birth	Edm.DateTime	0		
IsContact	Indicates whether person acts as contact for an account	Edm.Boolean	0		
IsMainContact	Indicates a person that acts as a main contact for an account	Edm.Boolean	0		

Property	Description	Edm Core Type	Max Length	Mandatory	Key
IsConsumer	Indicates whether person is a consumer	Edm.Boolean	0		
Obsolete	Obsolete	Edm.Boolean	0		
WebUri	Web Site	Edm.String	1.000		
Latitude	Latitude	Edm.Decimal	0		
Longitude	Longitude	Edm.Decimal	0		
IcType	Interaction Contact Type: 01 person 02 company	Edm.String	2		
CompanyId	ID of Company	Edm.String	255		
CompanyIdOrigin	Origin of Company	Edm.String	20		
MatchId	ID of matched Entity	Edm.String	255		
MatchIdOrigin	Origin of matched Entity	Edm.String	20		

The property `IdOrigin` indicates the source of the `Id`. It is maintained in the Self-Service Configuration app [Define Origins of Contact ID](#). To replicate SAP ERP data to SAP Marketing Cloud, interaction contacts origin `SAP_ERP_CONTACT` or `SAP_ERP_CUSTOMER` is used. To replicate SAP S/4HANA data to SAP Marketing Cloud, `SAP_S4_BUPA`, an additional facet for customer `SAP_S4_CUSTOMER`, for contact `SAP_S4_CONTACT` is used.

`CompanyId` and `CompanyIdOrigin` are used to create a relationship between a contact and a company.

`MatchId` and `MatchIdOrigin` are used to associate an interaction contact to data already created with different `Id` and `IdOrigin` in SAP Marketing Cloud, for example data transmitted from a different source system.

An interaction contact can be classified as contact (property: `IsContact = true`) for persons acting as contact for an account (B2B), and as a consumer (property: `IsConsumer = true`) that acts as an account (B2C). A person can be both a contact and a consumer at the same time.

Interaction contacts that are marked as `Obsolete` or `IsEndOfPurposeBlocked` are not visible and cannot be used in business processes. The `Obsolete` indicator can be removed. Setting the `IsEndOfPurposeBlocked` indicator is permanent.

## Marketing Attributes

Marketing attributes can be created only via the navigation property `MarketingAttributes` of the `InteractionContact` entity type.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>CategoryId</code>	ID of Marketing Category	Edm.String	75	x	x
<code>Value</code>	Value of Marketing Attribute	Edm.String	255	x	x

## Transferring Marketing Attributes

Marketing attributes and their assignments to business partners are transferred from a source system, for example SAP S/4HANA Enterprise, to SAP Marketing Cloud in two steps, as described in the following table:

Transfer of Marketing Attributes

Transfer of	From	To SAP Marketing Cloud
Master Data	<ul style="list-style-type: none"><li>Marketing Attribute Sets</li><li>Marketing Attributes</li></ul>	Marketing Attribute Categories For more information, see <a href="#">Marketing Attribute Categories [page 735]</a> .
Business Partner Assignments	Marketing Attributes Sets, including: <ul style="list-style-type: none"><li>Marketing Attribute Values</li><li>Marketing Attribute Value Descriptions</li></ul>	Marketing Attribute Values

## Integration

In Marketing, marketing attributes categories and marketing attribute values are visible in *Personal Data* of contacts, accounts, or individual customers. Marketing attribute categories, and marketing attribute values can be used in segmentation.

### i Note

- Marketing attribute categories always have a text in the system language. If no text is transferred from the source system, the marketing system automatically creates a text in the system language from the ID.
- Ensure that all attributes in the source system are named differently. Attributes with the same name cause an error that can be monitored in the [Import Monitor \[page 404\]](#).
- To prevent from overwriting attribute values, do not use the same attribute in different attribute sets in the source system.

Changes of master data and business partner assignments in the source system are automatically transferred to Marketing.



## MarketingArea

Marketing areas can be assigned to the interaction contact via the navigation property `MarketingAreas` of the `InteractionContact` entity type. The marketing area assigned to the interaction contact can be deleted by the OData `DELETE` operation on the entity type `MarketingArea`.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>Id</code>	ID of Interaction Contact	Edm.String	255	x	x
<code>IdOrigin</code>	Origin of Interaction Contact	Edm.String	20	x	x
<code>MarketingAreald</code>	ID of Marketing Area	Edm.String	40	x	x

## Function Imports

Function import `DeleteMarketingAttributes` can be used to delete all marketing attributes of an interaction contact.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>Id</code>	ID of Interaction Contact	Edm. String	255	x	x
<code>IdOrigin</code>	Origin of Interaction Contact	Edm. String	20	x	x

### 5.2.5.4 Field Extensibility

In addition to the pre-delivered attributes, you can add customer-specific fields using the [Custom Fields](#) app. For more information about how to do this, see [Custom Fields](#).

New fields can be added for the following BusinessContexts:

- Marketing: Interaction Contact
- Marketing Attributes for Contacts
- Marketing: Person
- Marketing: Company

If the field is added to the `BusinessContexts` `Person` or `Company`, the respective `ICType` (**01** for `Person` and **02** for `Company`) must be filled in the payload of new interaction contacts.

## 5.2.6 Products

Public OData API (`API_MKT_PRODUCT_SRV`) for Products.

### Overview

The OData service `API_MKT_PRODUCT_SRV` is used for standard SAP Marketing Cloud integration with other systems. It is used in marketing-driven and sales-driven processes to replicate product data to SAP Marketing Cloud.

For more information about the integration scenario, see SAP Marketing Cloud , [Integration with SAP ERP \[page 349\]](#)

For more information about external interfaces that SAP Marketing Cloud provides for creating or updating interaction contacts, interactions, interests, corporate accounts, product categories, and products, see [Integration APIs \[page 387\]](#).

OData service `API_MKT_PRODUCT_SRV` can be used to create SAP Marketing Cloud products from any source system. OData service `API_MKT_PRODUCT_SRV` supports the change of products. Each product is identified by the key of the product in the external system.



### Technical Data

OData Version	2.0
Root URI	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_PRODUCT_SRV;v=0002</code>
Service Metadata URI	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_PRODUCT_SRV;v=0002/\$metadata</code>
Authorizations	The following business catalog is required: <ul style="list-style-type: none"><li>• <code>SAP_CEC_BC_MKT_API_PRD2_PC</code></li></ul>
Communication Scenario ID	<code>SAP_COM_0171</code>
Component for Incidents	<ul style="list-style-type: none"><li>• <code>CEC-MKT-DM-PRO (Products and Product Categories)</code></li></ul>
Field Extensibility Supported	Yes

**i Note**  
Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/API_MKT_PRODUCT_SRV/ \$metadata?sap- documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Product Details Page</a> 	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li><li>Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Products (Metadata)</a> 	General access link takes you directly to the <i>Product</i> metadata file. One-time registration or logon is required.

### Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

[Basic Concepts \[page 583\]](#)

[Structure of OData Service API\\_MKT\\_PRODUCT\\_SRV \[page 585\]](#)

[Payload Examples for Products \[page 591\]](#)

Demonstrates creation and merge of products.

[Extensibility for Products \[page 602\]](#)

[Function Imports \[page 603\]](#)

Function imports are used to perform custom operations on an entity, which are typically not provided by OData operations.

## 5.2.6.1 Basic Concepts

OData service `API_MKT_PRODUCT_SRV` supports only batch processing for updates. Batch requests allow grouping multiple operations into a single HTTP request payload.

## i Note

For generally applicable recommendations and best practices, make sure you refer to the section [Best Practices and Recommended Package Sizes \[page 400\]](#).

Within a batch request, the following operations are supported:

- GET is supported for all entities.
- PATCH (MERGE) on the entity type `ProductOriginData`
- POST on the entity types `ProductCategoryAssignment` and `ProductName`
- DELETE on the entity type `ProductCategoryAssignment`
- DELETE on the entity type `MarketingArea`
- and the function `IMPORT` for `ProductOriginData`

Other operations, such as `CREATE` or `UPDATE` are not supported. Operation `DELETE` is only supported for entity type `ProductCategoryAssignment`.

Batch requests are submitted as a single `HTTP POST` request to the `$batch` endpoint of a service as described in [\[OData-URI ↗\]](#). The batch request must contain a content-type header specifying a content type of `multipart/mixed` and a boundary specification.

A `PATCH (MERGE)` request updates only the properties indicated in the request body and leaves everything untouched that was not mentioned. All properties that are not to be changed, can be omitted. The transmitted properties are merged with the data already stored in SAP Marketing Cloud.

If the OData service is not accessible, for example due to missing authorization, or because the system is not available, a corresponding `HTTP` status code is returned.

If the OData service is accepted by the gateway component in the SAP Marketing Cloud system, the `HTTP` status code 201 or 204 is returned. Potential processing errors are recorded in the SAP Marketing Cloud system in the Import Monitor app, where they can be monitored, restarted and discarded.

**Parent topic:** [Products \[page 582\]](#)

## Related Information

[Structure of OData Service API\\_MKT\\_PRODUCT\\_SRV \[page 585\]](#)

[Payload Examples for Products \[page 591\]](#)

[Extensibility for Products \[page 602\]](#)

[Function Imports \[page 603\]](#)

## 5.2.6.2 Structure of OData Service API\_MKT\_PRODUCT\_SRV

The API\_MKT\_PRODUCT\_SRV OData service consists of the following entity sets and entity types:

Entity Set	Entity Type	Entity Type Description
Products	Product	The Master Record
ProductOriginDataSet	ProductOriginData	Product origin data refer to products in SAP Marketing Cloud. Product data is collected and merged from several sources into the master data tables within SAP Marketing Cloud.
ProductCategoryAssignments	ProductCategoryAssignment	Product categories are assigned to a product.
ProductNames	ProductName	The product name master record.
ProductOriginDataNames	ProductOriginDataName	Product name and description can be defined per origin of the product data.
AdditionalIDs	AdditionalID	Additional ID of the product from a different product origin.
MarketingAreas	MarketingArea	Marketing area to which the product is assigned.

The OData service API\_MKT\_PRODUCT\_SRV supports OData batch processing for updates (i.e. insert, change, delete). Product origin data can be transferred by the OData PATCH (MERGE) operation for entity type `ProductOriginData`.

Product category assignments can be transferred by the OData POST operation on the entity type `ProductCategoryAssignment` via the navigation parameter `ProductCategoryAssignments`. A product category assignment can be deleted by the OData DELETE operation on the entity type `ProductCategoryAssignment`. Any operation on the `ProductCategoryAssignments` entity set without navigation from the `ProductOriginDataSet` is not supported. To delete all product category assignments of a special product category hierarchy the function import `DeleteProductCategoryAssignments` can be used.

Product name and description can be transferred by the OData POST operation on the entity type `ProductOriginDataName` via the navigation parameter `ProductNames`. Any operation on the `ProductOriginDataNames` entity set without navigation from the `ProductOriginDataSet` is not supported.

Additional ID can be transferred by the OData PATCH (MERGE) operation for the entity type `AdditionalID`.

The merge of two products from different source systems to one product within yMKT can be done via the OData function import `MergeProductOriginData`.

The product master record cannot be transferred by the OData POST/PATCH/MERGE operations. The master record is determined automatically based on the product origin data transferred from the source system and can be accessed via the GET operation.

The product name master record cannot be transferred by the OData POST/PATCH/MERGE operations. The product name master record is determined automatically based on the product origin data names transferred from the source system and can be accessed via the GET operation.

All entities supports the OData GET operation to read the data.

## Request Header

The request header contains the following additional header fields:

Header Field	Description	Edm Core Type	Max Length	Mandatory
Sap-Cuan-SequenceId	Unique technical identifier of the imported data.	Edm.String	30	X
Sap-Cuan-RequestTimestamp	Timestamp of the data	Edm.DateTime	0	*
Sap-Cuan-SequenceNumber	Sequence number of the request. This number is normally incremented each time a new request for the same sequence id is created.	Edm.Int16	0	*
Sap-Cuan-SourceSystemType	Type of the source system	Edm.String	20	X
Sap-Cuan-SourceSystemId	Identifier of the source system. This is a free text field.	Edm.String	255	X
Sap-Cuan-ExternalReferenceId	External reference of the inbound message	Edm.String	32	
Sap-Cuan-ExternalDocumentId	External identifier of the source document	Edm.String	20	

\* Either Sap-Cuan-RequestTimestamp or Sap-Cuan-SequenceNumber must be provided together with Sap-Cuan-SequenceId.

- The header fields Sap-Cuan-SequenceId and Sap-Cuan-RequestTimestamp or Sap-Cuan-SequenceNumber are used to check the sequence of the received data. Data with a timestamp older or sequence number lower than data already imported, is ignored.
- The Sap-Cuan-SourceSystemType and Sap-Cuan-SourceSystemId fields allow you to distinguish between different source systems.

- The `Sap-Cuan-ExternalReferenceId` and `Sap-Cuan-ExternalDocumentId` allow better error analysis because they contain external references to a source SOAP message and/or an IDoc.

## Product

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>ProductUUID</code>	UUID of the Product	Edm.Guid		x	x
<code>ProductID</code>	ID of the Product	Edm.String	50		
<code>ProductOrigin</code>	Origin of the Product	Edm.String	30		
<code>ProductImageURL</code>	Product Image URL	Edm.String	1333		
<code>WebsiteURL</code>	Website URL	Edm.String	1333		
<code>Brand</code>	Brand ID	Edm.String	50		
<code>IsBaseProduct</code>	Indicator: Is Base Product	Edm.Boolean			
<code>BaseProductID</code>	ID of the Base Product	Edm.String	50		
<code>BaseProductOrigin</code>	Origin of the Base Product	Edm.String	30		
<code>ProductValidEndDate</code>	End Date of Product Validity	Edm.DateTime			

## ProductOriginData

Only the properties `ProductID` and `ProductOrigin` are mandatory. All properties that are not to be changed can be omitted.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>ProductID</code>	ID of the Product	Edm.String	50	x	x
<code>ProductOrigin</code>	Origin of the Product	Edm.String	30	x	x

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductImageURL	Product Image URL	Edm.String	1333		
WebsiteURL	Website URL	Edm.String	1333		
Brand	Brand ID	Edm.String		50	
IsBaseProduct	Indicator Is Base Product	Edm.String			
BaseProductID	ID of the Base Product	Edm.String	50		
BaseProductOrigin	Origin of the Base Product	Edm.String	30		
ProductValidEndDate	End Date of Product Validity	Edm.DateTime			

The property `ProductOrigin` indicates the source of the `ProductID`. It is maintained in the Self-Service Configuration app `Define Origins of ProductID`. To replicate SAP ERP data to SAP Marketing Cloud, product origin `SAP_ERP_MATNR` is used.

The property `BaseProductOrigin` indicates the source of the `BaseProductID`. It is maintained in the Self-Service Configuration app `Define Origins of Product ID`.

Only products marked as base product (Indicator `IsBaseProduct` is `True`) can be assigned via properties `BaseProductID` and `BaseProductOrigin`. This reference can only be provided for products not marked as base product.

Brands can be maintained in the app [Brands](#) under [Import Data](#)

For the `ProductValidEndDate` only the date (without time portion) is relevant.

## ProductCategoryAssignment

Product category assignment can be created only via the navigation property `ProductCategoryAssignments` of the `ProductOriginData` entity type.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductID	ID of the Product	Edm.String	50	x	x
ProductOrigin	Origin of the Product	Edm.String	30	x	x



Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductCategory-HierarchyID	ID of Product Category Hierarchy	Edm.String	50	x	x
ProductCategoryID	ID of Product Category	Edm.String	50	x	x

## ProductName

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductUUID	UUID of the Product	Edm.Guid		x	x
Language	Language code	Edm.String	2	x	x
Name	Product Name	Edm.String	120		
ProductDescription	Product Description	Edm.String			

## ProductOriginDataName

Product name and description can be created only via the navigation property `ProductNames` of the `ProductOriginData` entity type.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductID	ID of the Product	Edm.String	50	x	x
ProductOrigin	Origin of the Product	Edm.String	30	x	x
Language	Language code	Edm.String	2	x	x
Name	Product Name from the source system	Edm.String	120		
ProductDescription	Product Description from the source system	Edm.String			

With version 2 of API\_MKT\_PRODUCT\_SRV you can transfer product descriptions that are longer than 512 characters. Only the first 512 characters are considered in the fuzzy search in the “Products Use and Resonance” Fiori App.

## AdditionalID

Additional IDs can be attached to the product only via the PATCH/MERGE operation. In case the product with product origin data identified by `AdditionalProductOrigin` and `AdditionalProductID` is already known in the system the product will be merged as additional product origin data to the product identified by `ProductOrigin` and `ProductID`. The processing is then analogue to the Function Import `MergeProductOriginData`.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>ProductOrigin</code>	Origin of the Product	Edm.String	30	x	x
<code>ProductID</code>	ID of the Product	Edm.String	50	x	x
<code>AdditionalProductOrigin</code>	Origin of the additional ID of the Product	Edm.String	30	x	x
<code>AdditionalProductID</code>	Additional ID of the Product	Edm.String	50	x	x

## MarketingArea

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>ProductOrigin</code>	Origin of the Product	Edm.String	30	x	x
<code>ProductID</code>	ID of the Product	Edm.String	50	x	x
<code>MktAreaId</code>	Marketing Area of the Product	Edm.String	40	x	x

Parent topic: [Products \[page 582\]](#)

## Related Information

[Basic Concepts \[page 583\]](#)

[Payload Examples for Products \[page 591\]](#)

[Extensibility for Products \[page 602\]](#)

[Function Imports \[page 603\]](#)

### 5.2.6.3 Payload Examples for Products

Demonstrates creation and merge of products.

The following examples show how you can use the products API. Insert your own data to fill the header and the entities.

#### Create 2 Products: Base Product and Product Variant

##### Sample Code

```
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ProductOriginDataSet (ProductID='CoffeeEspresso',ProductOrigin='SAP_ERP_MATNR')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductOrigin":"SAP_ERP_MATNR",
  "ProductID":"CoffeeEspresso",
  "WebsiteURL":"https://www.amazon.com/Organic-Espresso-Bean-Coffee-5-Pound/dp/
B002GWHAVM?th=1",
  "ProductImageURL":"https://images-na.ssl-images-amazon.com/images/I/
51tbABf6XKL._SX522_.jpg",
  "Brand":"","
  "IsBaseProduct":true
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ProductOriginDataSet (ProductID='CoffeeEspressoDecaf',ProductOrigin='SAP_ERP_MA
TNR') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
```

```

Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductOrigin":"SAP_ERP_MATNR",
  "ProductID":"CoffeeEspressoDecaf",
  "WebsiteURL":"https://www.amazon.com/Lavazza-Decaf-Ground-Coffee-Espresso/dp/
B000H6AX0E/ref=pd_sim_325_1?
_encoding=UTF8&pd_rd_i=B000H6AX0E&pd_rd_r=JQ5ZE9QB4PEC9J60QHZR&pd_rd_w=QGBTE&p
d_rd_wg=uPw2i&refRID=JQ5ZE9QB4PEC9J60QHZR&th=1",
  "Brand":"",
  "ProductImageURL":"https://images-na.ssl-images-amazon.com/images/I/91yLlLg
%2Be5L._SY679_.jpg",
  "BaseProductID":"CoffeeEspresso",
  "BaseProductOrigin":"SAP_ERP_MATNR"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Create Product with 2 Languages

### Sample Code

```

--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH ProductOriginDataSet(ProductID='Mocca',ProductOrigin='SAP_ERP_MATNR')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductOrigin":"SAP_ERP_MATNR",
  "ProductID":"Mocca",
  "WebsiteURL":"",
  "ProductImageURL":"",
  "Brand":"",
  "ProductValidEndDate":"9999-12-31T00:00:00",
  "BaseProductID":"",
  "BaseProductOrigin":"",
  "ProductValidEndDate":"2029-12-31T00:00:00",
  "IsBaseProduct":false
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ProductOriginDataSet(ProductID='Mocca',ProductOrigin='SAP_ERP_MATNR')/
ProductNames HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01

```

```

Sap-Cuan-ExternalReferenceId: 4711
{ "Language":"EN",
"Name":"Coffee Mocca",
"ProductDescription":"Ground Coffee - Caffe Mocca"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ProductOriginDataSet(ProductID='Mocca',ProductOrigin='SAP_ERP_MATNR')/
ProductNames HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "Language":"DE",
"Name":"Kaffee Mokka",
"ProductDescription":"Basis Kaffee - Mokka"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Create Product with 2 Product Category Assignments

### Sample Code

```

--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH ProductOriginDataSet(ProductID='Mocca',ProductOrigin='SAP_ERP_MATNR')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductOrigin":"SAP_ERP_MATNR",
"ProductID":"Mocca",
"WebsiteURL":"","
"ProductImageURL":"","
"Brand":"","
"ProductValidEndDate":"9999-12-31T00:00:00",
"IsBaseProduct":false
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ProductOriginDataSet(ProductID='Mocca',ProductOrigin='SAP_ERP_MATNR')/
ProductCategoryAssignments HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100

```

```

Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductCategoryHierarchyID":"Coffee",
  "ProductCategoryID":"FilterCoffee"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ProductOriginDataSet (ProductID='Mocca',ProductOrigin='SAP_ERP_MATNR')/
ProductCategoryAssignments HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductCategoryHierarchyID":"Coffee",
  "ProductCategoryID":"Mocca"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Create 2 Products and Merge into Best Record

### Sample Code

```

--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ProductOriginDataSet (ProductID='Cappuccino',ProductOrigin='SAP_ERP_MATNR')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductOrigin":"SAP_ERP_MATNR",
  "ProductID":"Cappuccino",
  "WebsiteURL":"https://www.amazon.com/Organic-Espresso-Bean-Coffee-5-Pound/dp/
B002GWHAVM?th=1",
  "ProductImageURL":"https://images-na.ssl-images-amazon.com/images/I/
51tbABf6XKL._SX522_.jpg",
  "Brand":"","
  "ProductValidEndDate":"9999-12-31T00:00:00"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ProductOriginDataSet (ProductID='CoffeeCap1',ProductOrigin='SAP_HYBRIS_PRODUCT'
) HTTP/1.1

```

```

Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductOrigin":"SAP_HYBRIS_PRODUCT",
"ProductID":"CoffeeCap1"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST MergeProductOriginData?
ProductID='Cappuccino'&ProductOrigin='SAP_ERP_MATNR'&AdditionalProductID='CoffeeCap1'&AdditionalProductOrigin='SAP_HYBRIS_PRODUCT' HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Update Product with Valid End Date

### Sample Code

```

--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ProductOriginDataSet(ProductID='CoffeeEspressoDecaf',ProductOrigin='SAP_ERP_MATNR') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{ "ProductOrigin":"SAP_ERP_MATNR",
"ProductID":"CoffeeEspressoDecaf",
"ProductValidEndDate":"2029-12-31T00:00:00"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Merge 2 Existing Products

### Sample Code

```
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST MergeProductOriginData?
ProductID='CoffeeEspresso'&ProductOrigin='SAP_ERP_MATNR'&AdditionalProductID='
407901109D5FBCF31500B0E4B2FD1696'&AdditionalProductOrigin='SAP_CRM_PRODUCT'
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Product Category Assignment

### Sample Code

```
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
ProductCategoryAssignments(ProductID='CoffeeEspressoDecaf',ProductOrigin='SAP_
ERP_MATNR',ProductCategoryHierarchyID='Coffee',ProductCategoryID='Espresso')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```



## Add Additional ID for Existing Product

### Sample Code

```
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
AdditionalIDs(ProductID='CoffeeEspresso',ProductOrigin='SAP_ERP_MATNR',AdditionalProductOrigin='SAP_C4C_PRODUCT',AdditionalProductID='407901109D5FBCF31500B0E4B2FD1696') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODUCT
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "ProductOrigin":"SAP_ERP_MATNR",
  "ProductID":"CoffeeEspresso",
  "AdditionalProductOrigin":"SAP_C4C_PRODUCT",
  "AdditionalProductID":"407901109D5FBCF31500B0E4B2FD1696"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete all Product Category Assignments of a Product Category Hierarchy for a Product

### Sample Code

```
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST DeleteProductCategoryAssignments?
HierarchyID='Coffee'&ProductID='CoffeeEspressoDecaf'&ProductOrigin='SAP_ERP_MATNR' HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Create Products with Marketing Area

### Sample Code

```
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ProductOriginDataSet(ProductID='JMAT_PROD_API_XX',ProductOrigin='XX')/
MarketingAreaSet HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: JMAT_PRODCATHIER_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20210112174358.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320210112174358_01
{
  "ProductOrigin" : "XX",
  "ProductId" : "JMAT_PROD_API_XX",
  "MktAreaId" : "GLOBAL"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch-
```

## Delete a Marketing Area

### Sample Code

```
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
MarketingAreas(ProductOrigin='SAP_ERP_MATNR3',ProductId='XXJMAT_PROD_APIXX',Mk
tAreaId='ICMA_DRINK') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: JMAT_XX
Sap-Cuan-RequestTimestamp: 20210125110620.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320210125110620_01
{
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch-
```

## Post Product Origin Data (Non-Batch)

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_PRODUCT_SRV;v=2/ProductOriginDataSet
POST data:
{ "ProductOrigin":"SAP_ERP_MATNR",
  "ProductID":"xxx_Product_xxx",
  "WebsiteURL":"www.test.de",
  "Brand":"BRAND"
}
```

## Post Product Origin Data Name Assignment (Non-Batch)

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_PRODUCT_SRV;v=2/ProductOriginDataNames
POST data:
{
  "ProductDescription" : "UI Product",
  "ProductOrigin" : "SAP_ERP_MATNR",
  "ProductID" : " xxx_Product_xxx",
  "Language" : "EN",
  "Name" : "JMAT_PROD _Name Non Batch Mode"
}
```

## Post Product Category Assignment (Non-Batch)

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_PRODUCT_SRV;v=2/MarketingAreas
POST data:
{"ProductOrigin" : "SAP_ERP_MATNR",
  "ProductId" : " xxx_Product_xxx",
  "MktAreaId" : "GLOBAL"
}
```

## Delete Product Marketing Area Assignment (Non-Batch)

### Sample Code

```
DELETE https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_PRODUCT_SRV;v=2/
```

```
MarketingAreas(ProductOrigin='SAP_ERP_MATNR',ProductId='xxx_Product_xxx',MktAreaId='GLOBAL')
```

## Delete Product Category Assignment (Non-Batch)

### Sample Code

```
DELETE https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/ProductCategoryAssignments(ProductID='xxx_Product_xxx',ProductOrigin='SAP_ERP_MATNR',ProductCategoryHierarchyID='Prod_Hier_1',ProductCategoryID='Prod_Cat_1')
```

## GET Requests

### Get Product Origin Data Names with specific Product ID and Origin

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV/ProductOriginDataNames?$filter=ProductOrigin eq 'SAP_ERP_MATNR' and ProductID eq 'JMAT_PROD_API'&$inlinecount=allpages&$top=10&$inlinecount=allpages
```

### Get Product Names of Product Origin Data Set with specific Product ID and Origin

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV/ProductOriginDataSet(ProductOrigin='SAP_ERP_MATNR',ProductID='JMAT_PROD_API')/ProductNames?$skip=0&$top=10&$inlinecount=allpages
```

### Get Product with Product UUID

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV/Products(guid'b5f39901-144d-a679-1700-236c971bc74e')
```

### Get Product with Special Brand ID

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV/Products?$skip=0&$top=10&$filter=ProductImageURL eq 'www.productimage.de'
```

### Get Product with Special Product Image URL

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV/Products?$skip=0&$top=10&$filter=ProductImageURL eq 'www.productimage.de'
```

### Get Product Names of Product with specific Product ID and Origin with ServiceVersion 2; filter on Language

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/Products(guid'14f49901-144d-a679-1700-236c971bc74e')/ProductNames?$inlinecount=allpages
```

## Get specific Product with Product ID and Origin with ServiceVersion 2

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/
ProductOriginDataSet?$filter=ProductID eq 'JMAT_PROD_API' and ProductOrigin eq
'SAP_ERP_MATNR'&$inlinecount=allpages
```

## Get first 10 Products with Service Version 2

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/
ProductOriginDataSet?$skip=0&$top=10&$inlinecount=allpages
```

## Get Product Category Assignments of specific Product with Product ID and Origin with Service Version 2

```
https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/
ProductCategoryAssignments?$filter=ProductID eq 'JMAT_Prod' and ProductOrigin eq
'SAP_ERP_MATNR'&$inlinecount=allpages
```

## Get Product Names of specific Product with Product UUID with Service Version 2

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/
Products(guid'faf49901-144d-a679-1700-236c971bc74e')/ProductNames?
$inlinecount=allpages
```

## Get Product Origin Data Set of specific Product with Product UUID with serviceVersion 2

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/
Products(guid'fef49901-144d-a679-1700-236c971bc74e')/ProductOriginDataSet?
$inlinecount=allpages
```

## Get ProductNames of specific Product with Product UUID with serviceVersion 2

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/
Products(guid'faf49901-144d-a679-1700-236c971bc74e')/ProductNames?
$inlinecount=allpages
```

## Get Product Category Assignments of specific Product Origin Data Set with ID and Origin with ServiceVersion 2

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/
ProductOriginDataSet(ProductID='JMAT_Prod',ProductOrigin='SAP_ERP_MATNR')/
ProductCategoryAssignments?$inlinecount=allpages
```

## Get Product Names of specific Product Origin Data Set with ID and Origin with ServiceVersion 2

```
GET https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_PRODUCT_SRV;v=2/
ProductOriginDataSet(ProductID='JMAT_Prod',ProductOrigin='SAP_ERP_MATNR')/
ProductNames?$inlinecount=allpages
```

Parent topic: [Products \[page 582\]](#)

## Related Information

[Basic Concepts \[page 583\]](#)

[Structure of OData Service API\\_MKT\\_PRODUCT\\_SRV \[page 585\]](#)

[Extensibility for Products \[page 602\]](#)

[Function Imports \[page 603\]](#)

## 5.2.6.4 Extensibility for Products

### Extending Attributes

In addition to the pre-delivered attributes, you can add customer-specific fields using the *Custom Fields* app. For more information about how to do this, see [Custom Fields](#).

New fields can be added for the following BusinessContext:

- Marketing: `Product`

### Create Extensibility Associations

You as an administrator in Marketing can define an association between a business object and the product or product categories to support customer-specific use cases. Example: Running a campaign for an event, which is defined as a product. You also want to analyze the campaign performance afterwards.

To create an association, proceed as follows:

- Open the app *Custom Fields*.
- Create a new field and enter:
  1. Business Context: *Marketing Campaign*
  2. Label: <Name of the Field>
  3. Identifier <Technical Name of the Field>
  4. Tooltip <Full Name or Help for the Field>
  5. Type *Association to Business Object*
  6. Business Object: *Marketing Product*

After you have done this, a reference to the standard product, including a proper value help is available.

For reporting purposes, the product or product category is also available as additional dimension in CDS queries which expose the enhanced business object.

Extensibility associations can also be used for Custom Business Objects, for example to define product-specific discounts or vouchers.

Parent topic: [Products \[page 582\]](#)

## Related Information

[Basic Concepts \[page 583\]](#)

[Structure of OData Service API\\_MKT\\_PRODUCT\\_SRV \[page 585\]](#)

[Payload Examples for Products \[page 591\]](#)

[Function Imports \[page 603\]](#)

### 5.2.6.5 Function Imports

Function imports are used to perform custom operations on an entity, which are typically not provided by OData operations.

#### Function Imports for Products

Function import **MergeProductOriginData** can be used to merge products with different `ProductIDs` from different origins into one product in SAP Marketing Cloud. Both products need to be already replicated. The `ProductOriginData` addressed with `AdditionalProductID` and `AdditionalProductOrigin` will then be moved as additional product data to product addressed with `ProductID` and `ProductOrigin`.

The product master data is determined from the main product origin data (addressed with `ProductID` and `ProductOrigin`). The properties not sent from the main origin are taken from the last updated additional origin. Every Update of the `ProductOriginData` leads to a redetermination of the product master data (golden record).

It is not possible to merge a product as additional product to two different main products.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
<code>AdditionalProductID</code>	Additional ID of the Product	Edm.String	50	x	x
<code>AdditionalProductOrigin</code>	Origin of the additional ID if the Product	Edm.String	30	x	x
<code>ProductID</code>	ID of the Product	Edm.String	50	x	x
<code>ProductOrigin</code>	Origin of the Product	Edm.String	30	x	x

Function import **DeleteProductCategoryAssignments** can be used to delete all product category assignments of a special product category hierarchy for a specified product.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductID	ID of the Product	Edm.String	50	x	x
ProductOrigin	Origin of the Product	Edm.String	30	x	x
ProductCategory-HierarchyID	ID of Product Category Hierarchy	Edm.String	50	x	x

Parent topic: [Products \[page 582\]](#)

## Related Information

[Basic Concepts \[page 583\]](#)

[Structure of OData Service API\\_MKT\\_PRODUCT\\_SRV \[page 585\]](#)

[Payload Examples for Products \[page 591\]](#)

[Extensibility for Products \[page 602\]](#)

## 5.2.7 Product Hierarchies and Categories

Public OData API (`API_MKT_PROD_CAT_HIERARCHY_SRV`) for Product Hierarchies and Categories.

The OData service `API_MKT_PROD_CAT_HIERARCHY_SRV` is used for standard SAP Marketing Cloud integration with SAP Commerce Cloud. It is used in commerce marketing to replicate product hierarchies and product categories to SAP Marketing Cloud.

For more information about the integration scenario, see SAP Marketing Cloud, [Integration with SAP Commerce Cloud \[page 62\]](#).

OData service `API_MKT_PROD_CAT_HIERARCHY_SRV` can be used to create SAP Marketing Cloud product hierarchies and product categories from any source system.

### i Note



If you encounter issues with the OData service `API_MKT_PROD_CAT_HIERARCHY_SRV`, create a support ticket under component `CEC-MKT-DM-PRO` (Products and Product Categories).

The component is not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).



## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_PRODCAT_HIERARCHY_SRV/ \$metadata?sap- documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Product Category and Product Hierarchy Details Page</a> 	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>1. On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li><li>2. Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Product Category and Product Hierarchy API</a> 	General access link takes you directly to the <i>Product Category and Product Hierarchy</i> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

[Basic Concepts \[page 605\]](#)

[Structure of OData Service API\\_MKT\\_PRODCAT\\_HIERARCHY\\_SRV \[page 606\]](#)

[Payload Example for Product Hierarchies \[page 611\]](#)

Demonstrates creation of product hierarchies and categories.

[Extensibility for Product Categories \[page 614\]](#)

## 5.2.7.1 Basic Concepts


### i Note

For generally applicable recommendations and best practices, make sure you refer to the section [Best Practices and Recommended Package Sizes \[page 400\]](#).

OData service `API_MKT_PRODCAT_HIERARCHY_SRV` supports batch processing. Within a batch request only the operation `PATCH (MERGE)` on the entity type `ProductCategory` or the operation `POST` on the entity types `ProductHierarchy`, `ProductHierarchyName` and `ProductCategoryName` are supported. Other operations, such as create, update or delete are not supported.

Additionally for all entities the `GET` operation is supported.

Batch requests allow grouping multiple operations into a single HTTP request payload.

Batch requests are submitted as a single HTTP POST request to the `$batch` endpoint of a service as described in [OData-URI ]. The batch request must contain a content-type header specifying a content type of `multipart/mixed` and a boundary specification.

A `PATCH (MERGE)` request updates only the properties indicated in the request body and leaves everything untouched that was not mentioned. All properties that are not to be changed, can be omitted. The transmitted properties are merged with the data already stored in SAP Marketing Cloud.

If the OData service is not accessible - for example due to missing authorization, or because the system is not available - a corresponding HTTP status code is returned.

If the OData service is accepted by the gateway component in the SAP Marketing Cloud system, the HTTP status code 201 or 204 is returned. Potential processing errors are recorded in the SAP Marketing Cloud system in the Import Monitor app, where they can be monitored, restarted and discarded.

**Parent topic:** [Product Hierarchies and Categories \[page 604\]](#)

## Related Information

[Structure of OData Service API\\_MKT\\_PRODCAT\\_HIERARCHY\\_SRV \[page 606\]](#)

[Payload Example for Product Hierarchies \[page 611\]](#)

[Extensibility for Product Categories \[page 614\]](#)

### 5.2.7.2 Structure of OData Service API\_MKT\_PRODCAT\_HIERARCHY\_SRV

The `API_MKT_PRODCAT_HIERARCHY_SRV` OData service consists of the following entity sets and entity types:

Entity Set	Entity Type	Entity Type Description
ProductHierarchies	ProductHierarchy	Product Hierarchy entity refers to data kept in the master data tables within SAP Marketing Cloud.
ProductHierarchyNames	ProductHierarchyNames	Product Hierarchy Names entity has the related name and description fields and can be maintained for several languages.

Entity Set	Entity Type	Entity Type Description
ProductCategories	ProductCategory	Product Category entity refers to data kept in the master data tables within SAP Marketing Cloud. The ID, a parent category ID, the type, and the Hierarchy ID are the fields which can be maintained.
ProductCategoryNames	ProductCategoryName	Similar to the hierarchy, the entity Product Category has a related Product Category Name entity which holds the Name and Description in several languages.

The OData service `API_MKT_PRODCA_T_HIERARCHY_SRV` supports OData batch processing.

Product hierarchies can be transferred by the OData `POST` operation for entity type `ProductHierarchy`.

Product categories can be transferred by the OData `PATCH (MERGE)` operation on the entity type `ProductCategory`.

Product hierarchy name and description can be transferred by the OData `POST` operation on the entity type `ProductHierarchyName` via the navigation parameter `ProductHierarchyNames`. Any operation on the `ProductHierarchyNames` entity set without navigation from the `ProductHierarchies` is not supported.

Product category name and description can be transferred by the OData `POST` operation on the entity type `ProductCategoryName` via the navigation parameter `ProductCategoryNames`. Any operation on the `ProductCategoryNames` entity set without navigation from the `ProductCategories` is not supported.

All entities support the OData `GET` operation to read the data.

## Request Header

The request header contains the following additional header fields:

Header Field	Description	Edm Core Type	Max Length	Mandatory
<code>Sap-Cuan-SequenceId</code>	Unique technical identifier of the imported data.	Edm.String	30	X
<code>Sap-Cuan-RequestTimestamp</code>	Timestamp of the data	Edm.DateTime	0	*

Header Field	Description	Edm Core Type	Max Length	Mandatory
Sap-Cuan-SequenceNumber	Sequence number of the request. This number is normally incremented each time a new request for the same sequence id is created.	Edm.Int16	0	*
Sap-Cuan-SourceSystemType	Type of the source system	Edm.String	20	X
Sap-Cuan-SourceSystemId	Identifier of the source system. This is a free text field.	Edm.String	255	X
Sap-Cuan-ExternalReferenceId	External reference of the inbound message	Edm.String	32	
Sap-Cuan-ExternalDocumentId	External identifier of the source document	Edm.String	20	

The header fields `Sap-Cuan-SequenceId` and `Sap-Cuan-RequestTimestamp` or `Sap-Cuan-SequenceNumber` are used to check the sequence of the received data. Data with a timestamp older or sequence number lower than data already imported, is ignored.

The `Sap-Cuan-SourceSystemType` and `Sap-Cuan-SourceSystemId` fields allow you to distinguish between different source systems.

\* Either `Sap-Cuan-RequestTimestamp` or `Sap-Cuan-SequenceNumber` must be provided together with `Sap-Cuan-SequenceId`.

The `Sap-Cuan-ExternalReferenceId` and `Sap-Cuan-ExternalDocumentId` allow better error analysis because they contain external references to a source SOAP message and/or an IDoc.

## ProductHierarchy

ProductHierarchy is the root entity and can be created or updated via the POST operation.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductHierarchyID	ID of the Product Hierarchy	Edm.String	50	x	x

Property	Description	Edm Core Type	Max Length	Mandatory	Key
SourceSystemType	Type code of the source system	Edm.String	20		
ProductHierarchyUsage	Usage Code of the Product Hierarchy	Edm.String	1		

The allowed code values are:

Property	Code Value	Description
SourceSystemType		undefined
	CRM	SAP CRM System
	C4C	SAP Cloud for Customer System
	YCOM	SAP Commerce System
	PMR	SAP Promotion Management for Retail
	ERP	SAP ERP System
	S4	SAP S4 System
ProductHierarchyUsage	A	Product Category Hierarchy
	B	Service Category Hierarchy

## ProductHierarchyName

Product Hierarchy Name can be created only via the navigation property `ProductHierarchyNames` of the `ProductHierarchy` entity type.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductHierarchyID	ID of the Product Hierarchy	Edm.String	50	x	x
Language	Language	Edm.String	2	x	x
Name	Name of the Product hierarchy	Edm.String	120		
Description	Description of the Product Hierarchy	Edm.String	512		

## ProductCategory

Product Category can be created via the Merge/Patch Operation. The corresponding root entity ProductHierarchy needs to be created before Product Categories can be created.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductHierarchyID	ID of the Product Hierarchy	Edm.String	50	x	x
ProductCategoryID	ID of the Product Category	Edm.String	50	x	x
ParentProductCategoryID	ID of the superordinate Product Category	Edm.String	50		
ProductCategoryType	Type code of the Product Category	Edm.String	2		

The allowed code values for ProductCategoryType are:

Property	Code Value	Description
ProductCategoryType		undefined
	A	Process Category
	B	Incident Category
	C	Object Category
	D	Cause Category
	E	Solution Category

All properties that are not to be changed can be omitted.

## ProductCategoryName

Product Category Name can be created only via the navigation property ProductCategoryNames of the ProductCategory entity type.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ProductHierarchyID	ID of the Product Hierarchy	Edm.String	50	x	x
ProductCategoryID	ID of the Product Category	Edm.String	50	x	x

Property	Description	Edm Core Type	Max Length	Mandatory	Key
Language	Language	Edm.String	2	x	x
Name	Name of the Product Category	Edm.String	120		
Description	Description of the Product Category	Edm.String	512		

Property	Description	Edm Core Type	Max Length	Mandatory	Key
Language	Language	Edm.String	2	x	x
Name	Name of the product hierarchy	Edm.String	120		
Description	Description of the product hierarchy	Edm.String	512		

Parent topic: [Product Hierarchies and Categories \[page 604\]](#)

## Related Information

[Basic Concepts \[page 605\]](#)

[Payload Example for Product Hierarchies \[page 611\]](#)

[Extensibility for Product Categories \[page 614\]](#)

### 5.2.7.3 Payload Example for Product Hierarchies

Demonstrates creation of product hierarchies and categories.

The following example shows how you can use the product hierarchies and categories API. Insert your own data to fill the header and the entities.

#### Create Product Hierarchy and Product Categories

##### Sample Code

```
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
```

```

--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ProductHierarchies HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "ProductHierarchyID": "Coffee"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ProductHierarchies(ProductHierarchyID='Coffee')/ProductHierarchyNames
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "Language": "EN",
  "Name": "Coffee",
  "Description": "Coffee"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST ProductHierarchies(ProductHierarchyID='Coffee')/ProductHierarchyNames
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "Language": "DE",
  "Name": "Kaffee",
  "Description": "Kaffee"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ProductCategories(ProductHierarchyID='Coffee',ProductCategoryID='Espresso')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "ProductHierarchyID": "Coffee",
  "ProductCategoryID": "Espresso",

```



```

    "ParentProductCategoryID": ""
  }
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST
ProductCategories(ProductHierarchyID='Coffee',ProductCategoryID='Espresso')/
ProductCategoryNames HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "Language": "EN",
  "Name": "Espresso",
  "Description": "Espresso"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ProductCategories(ProductHierarchyID='Coffee',ProductCategoryID='FilterCoffee'
) HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: YMQCLNT100
Sap-Cuan-SourceSystemType: SAP_ERP
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "ProductHierarchyID": "Coffee",
  "ProductCategoryID": "FilterCoffee",
  "ParentProductCategoryID": ""
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST
ProductCategories(ProductHierarchyID='Coffee',ProductCategoryID='FilterCoffee'
)/ProductCategoryNames HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "Language": "EN",
  "Name": "Filtered Coffee",
  "Description": "Filtered Coffee"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
ProductCategories(ProductHierarchyID='Coffee',ProductCategoryID='DecafEspresso
') HTTP/1.1

```

```

Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "ProductHierarchyID": "Coffee",
  "ProductCategoryID": "DecafEspresso",
  "ParentProductCategoryID": "Espresso"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST
ProductCategories(ProductHierarchyID='Coffee',ProductCategoryID='DecafEspresso')/ProductCategoryNames HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-SourceSystemId: XXXCLNT100
Sap-Cuan-SourceSystemType: XXX_XXX
Sap-Cuan-SequenceId: XXX_PRODHIER
Sap-Cuan-RequestTimestamp: 2017-12-06T08:01:01
Sap-Cuan-ExternalReferenceId: 4711
{
  "Language": "EN",
  "Name": "Decaf Espresso",
  "Description": "Espresso - Coffein free"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

Parent topic: [Product Hierarchies and Categories \[page 604\]](#)

## Related Information

[Basic Concepts \[page 605\]](#)

[Structure of OData Service API\\_MKT\\_PRODCA\\_T\\_HIERARCHY\\_SRV \[page 606\]](#)

[Extensibility for Product Categories \[page 614\]](#)

### 5.2.7.4 Extensibility for Product Categories

#### Extending Attributes

In addition to the pre-delivered attributes, you can add customer-specific fields using the *Custom Fields* app. For more information about how to do this, see [Custom Fields](#).

New fields can be added for the following BusinessContext:

- Marketing: `ProductCategory`

Parent topic: [Product Hierarchies and Categories \[page 604\]](#)

## Related Information

[Basic Concepts \[page 605\]](#)

[Structure of OData Service API\\_MKT\\_PRODCAT\\_HIERARCHY\\_SRV \[page 606\]](#)

[Payload Example for Product Hierarchies \[page 611\]](#)

## 5.2.8 Interactions

Public OData API (`API_MKT_INTERACTION_SRV`) for Interactions.

### Overview

#### i Note

For **business documents** (leads, opportunities, sales orders and so on), we recommend that you use the API Service `CUAN_BUSINESS_DOCUMENT_IMP_SRV`, since it provides an **upsert** function and updates an already existing entry depending on timestamp information.

OData Version	2.0
Root URI	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_SRV</code>
Service Metadata URI:	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_SRV/\$metadata</code>
Authorizations	<p>SAP delivers the following template role, which you can copy: <code>SAP_CEI_API_INTERACTION</code> .</p> <p>The following business catalog role is required: <code>SAP_CEC_BC_MKT_API_IA_PC</code></p>
Communication Scenario ID	<code>SAP_COM_0206</code>

**i Note**



Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

**i Note**

You can convert the XML file to an XML table to make it easier to read.

Access Link	Remarks
<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_SRV/\$metadata?sap-documentation=all">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTERACTION_SRV/\$metadata?sap-documentation=all</a>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Interaction Integration Page</a> 	<p>General access to the <i>Interaction Integration</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>1. On the <i>Interaction Integration</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>2. Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Interactions API</a> 	General access to the <i>Interactions</i> metadata file. One-time registration or logon is required.

### [Basic Concepts \[page 617\]](#)

The public API for Interactions `API_MKT_INTERACTION` supports operations on the *Interaction Business Object*. Interactions refer to communication and information exchange of any kind between a user's company and a contact, such as emails to a company, phone calls to a contact, and posts written in social networks about a company or the company's products.

### [Structure of OData Service API\\_MKT\\_INTERACTION \[page 621\]](#)

This document describes the structure of the Public OData API service `API_MKT_INTERACTION`. Make sure you read the *Basic Concepts* topic before you start.

### [Payload Examples for Interactions \[page 633\]](#)

Payload examples for `API_MKT_INTERACTION`

### [Error Handling for Interactions \[page 647\]](#)

This section contains some troubleshooting tips for handling common errors involving interaction imports.

## 5.2.8.1 Basic Concepts

The public API for Interactions `API_MKT_INTERACTION` supports operations on the `Interaction Business Object`. Interactions refer to communication and information exchange of any kind between a user's company and a contact, such as emails to a company, phone calls to a contact, and posts written in social networks about a company or the company's products.

### i Note

For generally applicable recommendations and best practices, make sure you refer to the section [Best Practices and Recommended Package Sizes \[page 400\]](#).

## Prerequisites

Before interaction data can be imported, the following prerequisites must be met:

- **Upload sequence:** Before you upload interactions, you should upload any related products and product hierarchies first.
- Interaction type and communication medium have been assigned to a channel in the Self-Service Configuration app [Manage Interaction Content](#).
- Interests must be edited or uploaded in [Business Administration](#) > [Manage Interests](#).
- The BADl: Revise Interaction Data Before Import (CUAN\_IA\_REVISE\_FOR\_IMPORT) can be used to change data during import.
- The BADl: Process Follow-Up Steps After Successful Import of Interaction Data (CUAN\_IA\_IMPORT\_FOLLOW\_UP) can be used to update your custom business objects (CBOs) based on successfully saved interactions. For example, you can use a custom business object to count a contact's likes.

## Structure

The following fields are mandatory for POST operations of interaction data:

Attribute	Description
ID_ORIGIN	Qualifier for source system
ID	ID of contact of interaction in source system
IA_TYPE	Interaction type
COMM_MEDIUM	Communication medium/place

Attribute	Description
TIMESTAMP	<p>UTC time stamp in long form (YYYYMMDDhhmmssmmuuun)</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <p>The timestamp must always be provided as UTC time, so you have to adjust your local time accordingly before import. For example, if the timestamp shows New York local time 14:00:00, you have to adjust this to UTC by adding 5 hours to it: 19:00:00.</p> </div> <p>Examples for OData Format:</p> <ul style="list-style-type: none"> <li>• Number of milliseconds since midnight Jan 1, 1970. / <i>Date(1406014140922)/</i></li> <li>• <i>YY-MM-DDThh:mm:ss</i></li> </ul>

## Semantic Keys

### i Note

This refers to the semantic/external key of an interaction that is **not** a Business Document, that is where the communication medium is not BUSINESS\_DOCUMENT. If the communication medium **is** BUSINESS\_DOCUMENT, then the semantic/external key of a BusinessDocument entity is defined by the unique combination of the fields `SourceSystemId`, `SourceSystemType`, `ExternalObjectType`, `ExternalObjectId`.

The semantic key for interactions determines the uniqueness of an interaction record. The semantic key is defined by the following 7 fields, 5 mandatory and 2 optional:

- These 5 fields are **mandatory for POST operations** and are checked during import:
  - `ID_ORIGIN`: Origin of the interaction contact data (except in the case of ANONYMOUS interactions, as described below.)
  - `ID`: External ID of the interaction contact data (except in the case of ANONYMOUS interactions, as described below.)
  - `IA_TYPE`: Interaction type
  - `COMM_MEDIUM`: Communication medium
  - `TIMESTAMP`: UTC time stamp in long form (YYYYMMDDhhmmss.mmmuuun)
- These 2 semantic key fields are **optional** and are not checked during import:
  - `SOURCE_OBJECT_TYPE`: Object type of the source object, for example, an opportunity in SAP Cloud for Customer.

### i Note

If the communication medium is a **Business Document**, the `SOURCE_OBJECT_TYPE` is required. In this case the system checks to see whether the `SOURCE_OBJECT_TYPE` field is filled and returns an error if the field is empty.

- `SOURCE_OBJECT_ID`: Object ID of the source object, for example, the GUID of the SAP Cloud for Customer opportunity or the original post ID of the respective social media network (such as TW or FB).

### i Note

One exception to this is in the case of a social posting. For a social posting, a `SOURCE_OBJECT_ID` is required. In this case the system checks to see whether the `SOURCE_OBJECT_ID` field is filled and returns an error if the field is empty. The check behavior is described in the table.

If IA_TYPE is	SOURCE_OBJECT_ID should be filled with	Remarks
SOCIAL_POSTING	Original Post ID of the respective social media network	This is checked during upload.
SHOP_ITEM_VIEW	WEB_SESSION_ID	For recommendations only. Not checked during upload.
WEBSITE_SEARCH	WEB_SESSION_ID	For recommendations only. Not checked during upload.
CLICK_THROUGH	WEB_SESSION_ID	For recommendations only. Not checked during upload.
SALES_ORDER	SALES_ORDER_ID	For recommendations only. Not checked during upload.

## Interaction Types That Cannot Be Imported

You cannot import the following interaction types:

- MKT\_PERM\_OPTIN
- MKT\_PERM\_OPTIN\_PRE
- MKT\_PERM\_OPTOUT
- MKT\_PERM\_OPTOUT\_PRE
- NEWSLETTER\_SUBSCR
- NEWSLETTER\_UNSUBSCR
- NEWSL\_SUBSCR\_PRE
- NEWSL\_UNSUBSCR\_PRE
- DIG\_ACC\_SUBSCR

- DIG\_ACC\_UNSUBSCR
- EMAIL\_BOUNCE\_HARD
- EMAIL\_BOUNCE\_SOFT

If you want to import **Marketing Permissions** and **Newsletter Subscriptions**, you can use one of these services: `API_MKT_CONTACT`, `API_MKT_INTERACTION_CONTACT`, or `API_MKT_CORPORATE_ACCOUNT`.

For more information about **email bounces**, see [Email: Get Bounces \[page 125\]](#).

## Anonymous Interactions

To import anonymous interactions, activate the `IS_ANONYMOUS` field. The system does **not** create a contact for anonymous interactions.

The system behavior for anonymous interactions is as follows:

- **Usual Case for Anonymous Interactions:**

The `ID_ORIGIN` is anonymous and the `IS_ANONYMOUS` indicator is set ("X"). The system stores the interaction as an anonymous one.

- **Deviations from Usual Case:**

*Termination of Import if:*

1. The `ID_ORIGIN` is not anonymous and the `IS_ANONYMOUS` indicator is set ("X"). The system terminates the import with a corresponding notification.
2. The `ID_ORIGIN` is empty and the `IS_ANONYMOUS` indicator is not set (""). The system terminates the import with a corresponding notification.

*System Correction of Import Data:*

1. The `ID_ORIGIN` is anonymous and the `IS_ANONYMOUS` indicator is not set (""). The system sets the `IS_ANONYMOUS` indicator and stores the interaction as an anonymous one.
2. The `ID_ORIGIN` is anonymous, the `IS_ANONYMOUS` indicator is set ("X"), and the ID is empty. The system creates a new GUID for field ID.
3. The `ID_ORIGIN` is empty (and, in addition, the ID is empty) and the `IS_ANONYMOUS` indicator is set ("X"). The system sets the `ID_ORIGIN` to anonymous, stores the interaction as an anonymous one, and creates a new GUID for field ID.

## Standard Fields and Custom Fields

### Custom Fields

- You can add customer-specific fields using the [Custom Fields](#) app. For more information, see [Custom Fields](#). Custom fields are then automatically included during the OData uploads.

#### i Note

You can see the structure of your data in the OData metadata structure that is displayed when you log onto the system as follows: [https://<server&port>/sap/opu/odata/sap/cuan\\_import\\_srv/\\$metadata](https://<server&port>/sap/opu/odata/sap/cuan_import_srv/$metadata)



- You can find a full list of all valid values for interaction types in the Configuration activity [Manage Interaction Content](#). For more information, see [Managing Interaction Content](#).

Parent topic: [Interactions \[page 615\]](#)

## Related Information

[Structure of OData Service API\\_MKT\\_INTERACTION \[page 621\]](#)

[Payload Examples for Interactions \[page 633\]](#)

[Error Handling for Interactions \[page 647\]](#)

### 5.2.8.2 Structure of OData Service API\_MKT\_INTERACTION

This document describes the structure of the Public OData API service `API_MKT_INTERACTION`. Make sure you read the Basic Concepts topic before you start.

The Interaction OData API provides the following entity sets:

Entity Set	Description	Path
Interactions	This entity contains the interaction data.	/Interactions
InteractionInterests	This entity contains the interests assigned to an interaction.	/InteractionInterests
InteractionProducts	This entity contains the products in an interaction.	/InteractionProducts
InteractionProductCategories	This entity contains the product categories in an interaction.	/InteractionProductCategories
InteractionDigitalAssets	This entity contains the digital assets in an interaction.	/InteractionDigitalAssets
InteractionOffers	This entity contains the offers in an interaction.	/InteractionOffers
InteractionTags	This entity contains the tags in an interaction.	/InteractionTags
InteractionAdditionalObjects	This entity contains the additional objects referred to by an interaction.	/InteractionAdditionalObjects

Entity Set	Description	Path
InteractionLoyaltyPrograms	This entity contains the Loyalty Program attributes in an interaction.	/InteractionLoyaltyPrograms
InteractionAdditionalInteractionContact	This entity contains the additional contacts involved in an interaction.	/InteractionAdditionalInteractionContact
InteractionEvent	This entity contains the event referred to by an interaction.	/InteractionEvent

Furthermore, the following entity has the dedicated function of enabling high performance to be maintained when importing large amounts of data:

Entity Set	Description	Path
InteractionsDeepInsert	This entity is used to import large amounts of Interactions data.	/InteractionsDeepInsert

## Entity Sets

### i Note

To prevent an ODATA error message (**The metadata do not allow a null value.**), you must set a fixed initial value `00000000-0000-0000-0000-000000000000` for the UUID of all nodes, both root and subnodes.

## InteractionsDeepInsert

**Entity Path:** /InteractionsDeepInsert

The InteractionsDeepInsert entity enables high performant import of up to 100,00 interaction's data. You can perform the following operations on the InteractionsDeepInsert entity:

HTTP Method	Description	Path
<i>POST</i>	<p>You can <b>create</b> interactions including one or more of these sub-entities using DEEP INSERT:</p> <ul style="list-style-type: none"> <li>• InteractionInterest</li> <li>• InteractionProducts</li> <li>• InteractionProductCategories</li> <li>• InteractionDigitalAssets</li> <li>• InteractionOffers</li> <li>• InteractionTags</li> <li>• InteractionAdditionalObjects</li> <li>• InteractionLoyaltyPrograms</li> <li>• InteractionAdditionalContacts</li> <li>• InteractionEvents</li> </ul>	<i>/InteractionsDeepInsert</i>

**i Note**

This is the recommended method for the mass import of interactions data.

## Interactions

**Entity Path:** */Interactions*

You can perform the following operations on the *Interactions* entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of interactions. This method supports standard OData parameters such as <i>\$filter</i> , <i>\$select</i> , <i>\$top</i> , <i>\$skip</i> , <i>\$count</i> , <i>\$inlinecount</i> , and <i>\$orderby</i>	<i>/Interactions?\$top=1</i>
	Get the details of a specific interaction using the Interaction UUID.	<i>/Interactions(guid '&lt;Interaction UUID&gt;')</i>

HTTP Method	Description	Path
<i>POST</i>	<p>Create an interaction. If the interaction already exists, it is not created. An error is returned.</p> <p>Semantic key: (* = mandatory)</p> <ul style="list-style-type: none"> <li>• InteractionContactOrigin *</li> <li>• InteractionContactId *</li> <li>• CommunicationMedium*</li> <li>• InteractionType*</li> <li>• InteractionTimeStampUTC*</li> </ul> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>○ The timestamp must always be UTC time.</li> <li>○ When you <b>import</b> data, the local timestamp plus the difference to UTC is also allowed. For example, for New York local time 14:00:00, which is 5 hours before UTC, you could import the timestamp as:  <b>2017-12-18T14:00:00-05:00</b> or <b>2017-12-18T19:00:00</b>.</li> <li>○ When data is <b>read</b>, the UTC timestamp is always returned.</li> </ul> </div> <ul style="list-style-type: none"> <li>• InteractionSourceObjectType</li> <li>• InteractionSourceObject</li> </ul> <p>Batch mode is also supported.</p>	/Interactions
<i>POST</i>	<p>You can create multiple interactions including one or more of these sub-entities using DEEP INSERT:</p> <ul style="list-style-type: none"> <li>• InteractionInterest</li> <li>• InteractionProducts</li> <li>• InteractionProductCategories</li> <li>• InteractionDigitalAssets</li> <li>• InteractionOffers</li> <li>• InteractionTags</li> <li>• InteractionAdditionalObjects</li> <li>• InteractionLoyaltyPrograms</li> <li>• InteractionEvent</li> </ul> <p>Batch mode is also supported.</p>	/Interactions

HTTP Method	Description	Path
<i>PATCH</i>	<p>Update an interaction.</p> <ul style="list-style-type: none"> <li>• If the interaction does not exist, it is not created. An error is returned (no <b>upsert</b> is supported).</li> <li>• The interaction key (InteractionUUID) must be provided to check its existence of the interaction.</li> <li>• With the exception of the interaction key, all properties can be updated. If semantic key fields (see POST) are updated, a check is carried out to ensure that no duplicate interactions exist after the update. If duplicates exist, an error is returned.</li> <li>• Batch mode is also supported.</li> </ul>	<p>/ Interactions(guid'&lt;Interaction UUID&gt;')</p>
<i>DELETE</i>	<p>Deletion of interactions is not supported by this service. You must use one of the application jobs to delete interactions.</p>	<p>For more information, see <a href="#">Interactions</a>.</p>

## InteractionInterests

**Entity Path:** /InteractionInterests

You can perform the following operations on the `InteractionInterest` entity:

HTTP Method	Description	Path
<i>GET</i>	<p>Get a list of interests assigned to interactions. This method supports standard OData parameters such as <code>\$filter</code>, <code>\$select</code>, <code>\$top</code>, <code>\$skip</code>, <code>\$count</code>, <code>\$inlinecount</code>, and <code>\$orderby</code></p>	<p>/ Interactions(guid'InteractionUUID')/InteractionInterests</p>
<i>POST</i>	<p>Create an interest assignment to an interaction. If the interaction interest master data does not already exist, it is not created. An error is returned.</p> <p>Batch mode is also supported.</p>	<p>/ InteractionInterests</p>
<i>PATCH</i>	<p>Update the assignment of an interest to an interaction.</p> <ul style="list-style-type: none"> <li>• If the interest assignment does not exist, it is not created. An error is returned (no <b>upsert</b> is supported).</li> <li>• The entity key (InteractionInterestUUID) must be provided to check whether the interest exists.</li> <li>• With the exception of the interaction and the entity key, all properties can be updated.</li> <li>• Batch mode is also supported.</li> </ul>	<p>/ InteractionInterests</p>

HTTP Method	Description	Path
<i>DELETE</i>	Delete the assignment of an interest to an interaction. If the interest assignment does not exist, it is not deleted. An error is returned  Batch mode is also supported.	/InteractionInterests

## InteractionProducts

**Entity Path:** /InteractionProducts

You can perform the following operations on the `InteractionProducts` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of products assigned to an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	/Interactions(guid'InteractionUUID')/InteractionProducts
<i>POST</i>	Create a product assignment to an interaction. If the product master data does not already exist, it is not created. An error is returned.  Batch mode is also supported.	/InteractionProducts
<i>PATCH</i>	Update the assignment of a product to an interaction. <ul style="list-style-type: none"> <li>If the product assignment does not exist, it is not created. An error is returned (no upsert is supported).</li> <li>The entity key (InteractionProductUUID) must be provided to check its existence.</li> <li>With the exception of the key properties (InteractionUUID, ProductUUID, and InteractionProductUUID), all properties can be updated.</li> <li>Batch mode is also supported.</li> </ul>	/InteractionProducts
<i>DELETE</i>	Delete the assignment of a product to an interaction. If the product assignment does not exist, it is not deleted. An error is returned. <ul style="list-style-type: none"> <li>The entity key (InteractionProductUUID) must be provided to check its existence.</li> <li>Batch mode is also supported.</li> </ul>	/InteractionProducts

## InteractionProductCategories

**Entity Path:** /InteractionProductCategories

You can perform the following operations on the `InteractionProductCategories` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of categories assigned to an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	<code>/ Interactions (guid' Interaction UUID' ) / InteractionProductCategories</code>
<i>POST</i>	Create a product category assignment to an interaction. If the product category master data does not already exist, it is not created. An error is returned.  Batch mode is also supported.	<code>/InteractionProductCategories</code>
<i>PATCH</i>	Update the assignment of a product category to an interaction. <ul style="list-style-type: none"> <li>• If the product category assignment does not exist, it is not created. An error is returned (no upsert is supported).</li> <li>• The entity key (<code>InteractionProductCategoryUUID</code>) must be provided to check its existence.</li> <li>• With the exception of the key properties (<code>InteractionUUID</code>, <code>ProductCategoryUUID</code> and <code>InteractionProductCategoryUUID</code>), all properties can be updated.</li> <li>• Batch mode is also supported.</li> </ul>	<code>/InteractionProductCategories</code>
<i>DELETE</i>	Delete the assignment of a product category to an interaction. If the assignment does not exist, it is not deleted. An error is returned. <ul style="list-style-type: none"> <li>• The entity key (<code>InteractionProductCategoryUUID</code>) must be provided to check its existence.</li> <li>• Batch mode is also supported.</li> </ul>	<code>/InteractionProductCategories</code>

## InteractionDigitalAssets

Entity Path: `/InteractionDigitalAssets`

You can perform the following operations on the `InteractionDigitalAssets` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of digital assets assigned to an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	<code>/ Interactions (guid' Interaction UUID' ) / InteractionDigitalAssets</code>

HTTP Method	Description	Path
<i>POST</i>	Create a digital asset assignment to an interaction.  Batch mode is also supported.	<code>/InteractionDigitalAssest</code>
<i>PATCH</i>	Update the assignment of a digital asset to an interaction.  <ul style="list-style-type: none"> <li>• If the assignment does not exist, it is not created. An error is returned (no upsert is supported).</li> <li>• The entity key (InteractionDigitalAssetUUID) must be provided to check its existence.</li> <li>• With the exception of the interaction and the entity key, all properties can be updated.</li> <li>• Batch mode is also supported.</li> </ul>	<code>/InteractionDigitalAssest</code>
<i>DELETE</i>	Delete the assignment of a digital asset to an interaction. If the assignment does not exist, it is not deleted. An error is returned  Batch mode is also supported.	<code>/InteractionDigitalAssest</code>

## InteractionOffers

Entity Path: `/InteractionOffers`

You can perform the following operations on the `InteractionOffers` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of offers assigned to an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	<code>/ Interactions (guid' InteractionUUID') / InteractionOffers</code>
<i>POST</i>	Create an offer assignment to an interaction.  Batch mode is also supported.	<code>/InteractionOffers</code>
<i>PATCH</i>	Update the assignment of an offer to an interaction.  <ul style="list-style-type: none"> <li>• If the assignment does not exist, it is not created. An error is returned (no upsert is supported).</li> <li>• The entity key (InteractionOfferUUID) must be provided to check its existence.</li> <li>• With the exception of the interaction and the entity key, all properties can be updated.</li> <li>• Batch mode is also supported.</li> </ul>	<code>/InteractionOffers</code>



HTTP Method	Description	Path
<i>DELETE</i>	Delete the assignment of an offer to an interaction. If the assignment does not exist, it is not deleted. An error is returned  Batch mode is also supported.	/InteractionOffers

## InteractionTags

**Entity Path:** /InteractionTags

You can perform the following operations on the `InteractionTags` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of tags assigned to an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	/Interactions(guid'InteractionUUID')/InteractionTags
<i>POST</i>	Create a tag assignment to an interaction.  Batch mode is also supported.	/InteractionTags
<i>PATCH</i>	Update the assignment of a tag to an interaction. <ul style="list-style-type: none"> <li>• If the assignment does not exist, it is not created. An error is returned (no upsert is supported).</li> <li>• The entity key (InteractionTagUUID) must be provided to check its existence.</li> <li>• With the exception of the interaction, the entity key, and the TagOrigin, all properties can be updated.</li> <li>• Batch mode is also supported.</li> </ul>	/InteractionTags
<i>DELETE</i>	Delete the assignment of an interest to an interaction. If the interest assignment does not exist, it is not deleted. An error is returned  Batch mode is also supported.	/InteractionTags

## InteractionAdditionalObjects

**Entity Path:** /InteractionAdditionalObjects

You can perform the following operations on the `InteractionAdditionalObjects` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of additional objects referred to by an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	<code>/ Interactions (guid' InteractionUUID' ) / InteractionAdditionalObjects</code>
<i>POST</i>	Create an additional object assignment to an interaction.  Batch mode is also supported.	<code>/ InteractionAdditionalObjects</code>
<i>PATCH</i>	Update the assignment of an additional object to an interaction. <ul style="list-style-type: none"> <li>• If the assignment does not exist, it is not created. An error is returned (no upsert is supported).</li> <li>• The entity key (<code>InteractionAdditionalObjUUID</code>) must be provided to check its existence.</li> <li>• With the exception of the interaction and the entity key, all properties can be updated.</li> <li>• Batch mode is also supported.</li> </ul>	<code>/ InteractionAdditionalObjects</code>
<i>DELETE</i>	Delete the assignment of an additional object to an interaction. If the assignment does not exist, it is not deleted. An error is returned  Batch mode is also supported.	<code>/ InteractionAdditionalObjects</code>

## InteractionLoyaltyPrograms

Entity Path: `/InteractionLoyaltyPrograms`

You can perform the following operations on the `InteractionLoyaltyPrograms` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of tags referred to by an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	<code>/ Interactions (guid' InteractionUUID' ) / InteractionLoyaltyPrograms</code>
<i>POST</i>	Create a loyalty subnode assignment to an interaction. <ul style="list-style-type: none"> <li>• Batch mode is also supported.</li> </ul>	<code>/ InteractionLoyaltyPrograms</code>
<i>PATCH</i>	Update the assignment of loyalty attributes to an interaction. <ul style="list-style-type: none"> <li>• Batch mode is also supported.</li> </ul>	<code>InteractionLoyaltyPrograms (guid' InteractionLoyaltyUUIDUID)</code>

HTTP Method	Description	Path
<i>DELETE</i>	Delete the assignment of a Loyalty Subnode to an interaction. If the assignment does not exist, it is not deleted. An <b>error</b> is returned. <ul style="list-style-type: none"> <li>Batch mode is also supported.</li> </ul>	InteractionLoyaltyPrograms(guid'InteractionLoyaltyUUIDUID)

## InteractionAdditionalInteractionContacts

Entity Path: /InteractionAdditionalInteractionContact

### i Note

The AdditionalInteractionContact entity is not visible on the UI, however it can still be used to perform the below operations.

You can perform the following operations on the `InteractionAdditionalInteractionContact` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of additional contacts referred to by an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	/ Interactions(guid'InteractionUUID') / InteractionAddiotnalInteractionContact
<i>POST</i>	Create an additional contact assignment for an interaction. <div data-bbox="483 1361 999 1547" data-label="Text"> <p><b>i Note</b></p> <p>Unlike on the root entity, if the additional contact does not exist, an error is returned. The contact will not be created.</p> </div> <p>Batch mode is also supported.</p>	/ InteractionAdditionalInteractionContact
<i>PATCH</i>	Update the assignment of additional Contacts to an interaction. <ul style="list-style-type: none"> <li>If the contact does not exist, the interaction is not created. An error is returned (no upsert is supported).</li> <li>The entity key (InteractionAdditionalContactUUID) must be provided to check its existence.</li> <li>With the exception of the interaction and the entity key, all properties can be updated.</li> <li>Batch mode is also supported.</li> </ul>	/ InteractionAdditionalInteractionContact

HTTP Method	Description	Path
<i>DELETE</i>	Delete the assignment of an additional contacts to an interaction. If the assignment does not exist, it is not deleted. An error is returned  Batch mode is also supported.	/InteractionAdditionalInteractionContact

## InteractionEvents

Entity Path: /InteractionEvent

### i Note

You can perform the following operations on the `InteractionEvent` entity:

HTTP Method	Description	Path
<i>GET</i>	Get a list of events referred to by an interaction. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	/Interactions(guid'InteractionUUID')/InteractionEvent
<i>POST</i>	Create an event assignment for an interaction.  <div data-bbox="481 1193 572 1229" data-label="Section-Header"> <h3>i Note</h3> </div> <div data-bbox="481 1249 968 1348" data-label="Text"> <p>Unlike on the root entity, if the event does not exist, an error is returned. The event will not be created.</p> </div> <p>Batch mode is also supported.</p>	/InteractionEvent
<i>PATCH</i>	Update the assignment of an event to an interaction. <ul style="list-style-type: none"> <li>• If the event does not exist, the interaction is not created. An error is returned (no upsert is supported).</li> <li>• The entity key (InteractionEventUUID) must be provided to check its existence.</li> <li>• With the exception of the interaction and the entity key, all properties can be updated.</li> <li>• Batch mode is also supported.</li> </ul>	/InteractionEvent
<i>DELETE</i>	Delete the assignment of an event to an interaction. If the assignment does not exist, it is not deleted. An error is returned  Batch mode is also supported.	/InteractionEvent

Parent topic: [Interactions \[page 615\]](#)

## Related Information

[Basic Concepts \[page 617\]](#)

[Payload Examples for Interactions \[page 633\]](#)

[Error Handling for Interactions \[page 647\]](#)

### 5.2.8.3 Payload Examples for Interactions

Payload examples for API\_MKT\_INTERACTION

#### i Note

An InteractionUUID value must be included in the payload in cases where the sender checks the metadata (as on SAP Cloud Integration). Otherwise you will get the error message 'The metadata do not allow a null value.'

- JSON version: "InteractionUUID": "00000000-0000-0000-0000-000000000000"
- XML version: <InteractionUUID>00000000-0000-0000-0000-000000000000</InteractionUUID>

Available Payload Examples

Entities	Payload Examples
Import Interactions Using InteractionsDeepInsert	<a href="#">Mass Import Using InteractionsDeepInsert [page 633]</a>
General Payload Examples	<a href="#">General Payload Examples [page 636]</a>
GET Requests	<a href="#">GET Requests [page 641]</a>
Offer Redemption - Interaction Offers	<a href="#">Offer Redemption - Entity Set: InteractionOffers [page 642]</a>
Event Subnode	<a href="#">Event Subnode [page 644]</a>

## Mass Import Using InteractionsDeepInsert

### Importing Interactions Using InteractionsDeepInsert

InteractionsDeepInsert is used only for the mass import of Interactions.

#### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/InteractionsDeepInsert

POST data:
{"Interactions":
[
{"InteractionContactOrigin":"EMAIL",
```

```

"InteractionContactId":"example@email.com",
"CommunicationMedium":"EMAIL",
"InteractionType":"EMAIL_BOUNCE_SOFT",
"InteractionTimeStampUTC":"2019-01-21T09:42:48",
"InteractionSourceObjectType":"ERP",
"InteractionSourceObject":"12345678",
"MarketingArea":"ICMA_DRINK",
"CampaignID":"12121212",
"InteractionLanguage":"EN",
"InteractionAmount":"12.34",
"InteractionCurrency":"EUR",
"InteractionLatitude":"49.304864",
"InteractionLongitude":"8.641526",
"InteractionInterests":
[
  {
    "ItemOfInterest":"DRINK"
  }
],
"InteractionProductCategories":
[
  {
    "ProductCategoryHierarchy":"Prod_Cat_1",
    "ProductCategory":"Product_CAT1 "
  }
],
"InteractionProducts":
[
  {
    "ProductOrigin":"SAP_PRODUCT",
    "Product":"Product_1",
    "InteractionProductAmount":"99.99",
    "InteractionProductQuantity":"1",
    "InteractionProductUnit":"m"
  }
]
],
[
  {
    "InteractionContactOrigin":"EMAIL",
    "InteractionContactId":"example2@email.com",
    "CommunicationMedium":"EMAIL",
    "InteractionType":"EMAIL_BOUNCE_SOFT",
    "InteractionTimeStampUTC":"2019-01-21T09:42:50",
    "InteractionSourceObjectType":"ERP",
    "InteractionSourceObject":"12345679",
    "MarketingArea":"ICMA_FOOD",
    "CampaignID":"12121212",
    "InteractionLanguage":"EN",
    "InteractionAmount":"12.34",
    "InteractionCurrency":"EUR",
    "InteractionLatitude":"49.304864",
    "InteractionLongitude":"8.641526",
    "InteractionInterests":
    [
      {
        "ItemOfInterest":"FOOD"
      }
    ]
  }
]
]
}

```

## Post Interaction with InteractionAdditionalInteractionContacts Subnodes in Batch

### Sample Code

```
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InteractionsDeepInsert HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"Interactions":
[
{
"InteractionUUID":"539968d3-6aa5-f08f-a29a-b71ec6632712",
"InteractionContactOrigin":"EMAIL",
"InteractionContactId":"2020012109125288_apitest@teamwdf02.de",
"CommunicationMedium":"EMAIL",
"InteractionType":"EMAIL_BOUNCE_SOFT",
"InteractionTimeStampUTC":"2020-01-21T09:12:52",
"InteractionSourceObjectType":"ERP",
"InteractionSourceObject":"12345678",
"SourceSystemType":"ERP",
"SourceSystem":"ERP001",
"InteractionTags":
[
{
"TagOrigin":"INTERNAL",
"TagType":"SEARCHTERM",
"TagName":"geocode:49.3,8.65,10km"
}
],
"InteractionAdditionalInteractionContacts":
[
{
"InteractionContactOrigin":"SAP_CRM_BUPA",
"InteractionContactId":"Contact_NB_1_WDF02_apitest@teamwdf02.de"}
]
}
,
{
"InteractionUUID":"ec4d21d9-c42e-2e82-8649-9da266dbafdc",
"InteractionContactOrigin":"EMAIL",
"InteractionContactId":"_2_88_2020012109125288_apitest@teamwdf02.de",
"CommunicationMedium":"FB",
"InteractionType":"SOCIAL_POSTING",
"InteractionTimeStampUTC":"2020-01-21T06:12:52",
"InteractionSourceObjectType":"ERP",
"InteractionSourceObject":"12345678",
"SourceSystemType":"ERP",
"SourceSystem":"ERP001",
"InteractionTags":
[
{
"TagOrigin":"INTERNAL",
"TagType":"SEARCHTERM",
"TagName":"geocode:49.3,8.65,10km"
}
],
"InteractionAdditionalInteractionContacts":
[
{
"InteractionContactOrigin":"SAP_CRM_BUPA",
"InteractionContactId":"Contact_NB_2_WDF02_apitest@teamwdf02.de",
```

```

"InteractionContactUUID":"6c0b84b7-5523-1ee9-96d6-7d9a4ff26047"
}
,
{
"InteractionContactUUID":"6c0b84b7-5523-1ee9-96d6-7d9a4ff26047"
}
]
}
]
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch-

```

## General Payload Examples

### Import 2 Interactions with 3 Products and 3 Interests as Sub-Nodes

#### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST Interactions HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
"InteractionContactOrigin":"SAP_HYBRIS_CONSUMER",
"InteractionContactId":"4711",
"CommunicationMedium":"ONLINE_SHOP",
"InteractionType":"SHOP_CART_VIEW",
"InteractionTimeStampUTC":"2018-04-25T08:16:53",
"InteractionSourceObjectType":"COMMERCE_SC",
"InteractionSourceObject":"4444",
"MarketingArea":"CXXGLOBAL",
"CampaignID":"12121212",
"MarketingLocationOrigin":"","
"MarketingLocation":"","
"DigitalAccountType":"","
"DigitalAccount":"","
"MKT_AgreementOrigin":"","
"MKT_AgreementExternalID":"","
"InteractionStatus":"","
"InteractionReason":"","
"InteractionLanguage":"EN",
"InteractionAmount":"12.34",
"InteractionCurrency":"EUR",
"InteractionLatitude":"49.304864",
"InteractionLongitude":"8.641526",
"SpatialReferenceSystem":"","
"DeviceType":"","
"InteractionDeviceName":"","
"SourceSystemType":"COMMERCE",
"SourceSystem":"HC121",
"InteractionSourceObjectAddlID":"","

```



```

"InteractionSourceObjectStatus":"","
"InteractionSourceDataURL":"http://www.sap.com/shoppingcartlink",
"CampaignContentLinkURL":"","
"CampaignContentLinkName":"","
"InteractionLastChangedByUser":"","
"InteractionContentSubject":"","
"InteractionContent":"","
"InteractionInterests":
[
{
  "ItemOfInterest":"MarketingCloud",
  "InteractionIntrstWeightingFctr":1,
  "InteractionIntrstSentimentVal":1
},
{
  "ItemOfInterest":"BigData",
  "InteractionIntrstWeightingFctr":1,
  "InteractionIntrstSentimentVal":1
},
{
  "ItemOfInterest":"SAPHana",
  "InteractionIntrstWeightingFctr":1,
  "InteractionIntrstSentimentVal":1
}
],
"InteractionProducts":
[
{
  "ProductOrigin":"SAP_HYBRIS_PRODUCT",
  "Product":"PRD-0",
  "InteractionProdWeightingFactor":1,
  "InteractionProductSentimentVal":1,
  "InteractionProductAmount":"12.12",
  "InteractionProductQuantity":"1",
  "InteractionProductUnit":"m",
  "ProductRecommendationModelType":"","
  "InteractionProductStatus":"","
  "InteractionProductReason":""
},
{
  "ProductOrigin":"SAP_HYBRIS_PRODUCT",
  "Product":"PRD-1",
  "InteractionProdWeightingFactor":1,
  "InteractionProductSentimentVal":1,
  "InteractionProductAmount":"12.12",
  "InteractionProductQuantity":"1",
  "InteractionProductUnit":"m",
  "ProductRecommendationModelType":"","
  "InteractionProductStatus":"","
  "InteractionProductReason":""
},
{
  "ProductOrigin":"SAP_HYBRIS_PRODUCT",
  "Product":"PRD-2",
  "InteractionProdWeightingFactor":1,
  "InteractionProductSentimentVal":1,
  "InteractionProductAmount":"12.12",
  "InteractionProductQuantity":"1",
  "InteractionProductUnit":"m",
  "ProductRecommendationModelType":"","
  "InteractionProductStatus":"","
  "InteractionProductReason":""
}
]
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary

```

```

POST Interactions HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
  "InteractionContactOrigin": "SAP_HYBRIS_CONSUMER",
  "InteractionContactId": "4712",
  "CommunicationMedium": "ONLINE_SHOP",
  "InteractionType": "SHOP_ITEM_ADD",
  "InteractionTimeStampUTC": "2018-04-25T08:16:53",
  "InteractionSourceObjectType": "COMMERCE_SC",
  "InteractionSourceObject": "55599",
  "MarketingArea": "CXXGLOBAL",
  "CampaignID": "12121212",
  "MarketingLocationOrigin": "",
  "MarketingLocation": "",
  "DigitalAccountType": "",
  "DigitalAccount": "",
  "MKT_AgreementOrigin": "",
  "MKT_AgreementExternalID": "",
  "InteractionStatus": "",
  "InteractionReason": "",
  "InteractionLanguage": "EN",
  "InteractionAmount": "12.34",
  "InteractionCurrency": "EUR",
  "InteractionLatitude": "49.304864",
  "InteractionLongitude": "8.641526",
  "SpatialReferenceSystem": "",
  "DeviceType": "",
  "InteractionDeviceName": "",
  "SourceSystemType": "COMMERCE",
  "SourceSystem": "HC121",
  "InteractionSourceObjectAddlID": "",
  "InteractionSourceObjectStatus": "",
  "InteractionSourceDataURL": "http://www.sap.com/shoppingcartlink",
  "CampaignContentLinkURL": "",
  "CampaignContentLinkName": "",
  "InteractionLastChangedByUser": "",
  "InteractionContentSubject": "",
  "InteractionContent": "",
  "InteractionInterests":
  [
    {
      "ItemOfInterest": "MarketingCloud",
      "InteractionIntrstWeightingFctr": 1,
      "InteractionIntrstSentimentVal": 1
    },
    {
      "ItemOfInterest": "BigData",
      "InteractionIntrstWeightingFctr": 1,
      "InteractionIntrstSentimentVal": 1
    },
    {
      "ItemOfInterest": "SAPHana",
      "InteractionIntrstWeightingFctr": 1,
      "InteractionIntrstSentimentVal": 1
    }
  ],
  "InteractionProducts":
  [
    {
      "ProductOrigin": "SAP_HYBRIS_PRODUCT",
      "Product": "PRD-0",
      "InteractionProdWeightingFactor": 1,
      "InteractionProductSentimentVal": 1,
      "InteractionProductAmount": "12.12",
      "InteractionProductQuantity": "1",
      "InteractionProductUnit": "m",

```

```

        "ProductRecommendationModelType":"","
        "InteractionProductStatus":"","
        "InteractionProductReason":""
    },
    {
        "ProductOrigin":"SAP_HYBRIS_PRODUCT",
        "Product":" PRD-1",
        "InteractionProdWeightingFactor":1,
        "InteractionProductSentimentVal":1,
        "InteractionProductAmount":"12.12",
        "InteractionProductQuantity":"1",
        "InteractionProductUnit":"m",
        "ProductRecommendationModelType":"","
        "InteractionProductStatus":"","
        "InteractionProductReason":""
    },
    {
        "ProductOrigin":"SAP_HYBRIS_PRODUCT",
        "Product":"PRD-2",
        "InteractionProdWeightingFactor":1,
        "InteractionProductSentimentVal":1,
        "InteractionProductAmount":"12.12",
        "InteractionProductQuantity":"1",
        "InteractionProductUnit":"m",
        "ProductRecommendationModelType":"","
        "InteractionProductStatus":"","
        "InteractionProductReason":""
    }
}
}
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Update Interaction 1 Root and Sub-Node Product via BATCH Request

### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH Interactions(InteractionUUID=guid'60b1329a-1d04-
a325-1600-236ca577cc6a') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
  "InteractionSourceObject":"87654321",
  "InteractionUUID":"60b1329a-1d04-a325-1600-236ca577cc6a",
  "InteractionContactId":"4711",
  "MarketingArea":"CXXGLOBAL",
  "CampaignID":"986532",
  "InteractionLatitude":"50.304864",
  "InteractionLongitude":"5.228967",
  "InteractionLanguage":"DE",
  "CommunicationMedium":"WEB",
  "InteractionType":"WEBSITE_REGISTRATION"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary

```

```

PATCH InteractionProducts(InteractionProductUUID=guid'63b1329a-1d04-
a325-1600-236ca577cc6a') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
  "InteractionUUID":"60b1329a-1d04-a325-1600-236ca577cc6a",
  "ProductOrigin":"SAP_HYBRIS_PRODUCT",
  "Product":"PRD-2",
  "InteractionProdWeightingFactor":2,
  "InteractionProductSentimentVal":2
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Post New Sub-Node, Update Root and Sub-Nodes and Delete Sub-Node:

### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InteractionInterests HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
  "InteractionUUID":"60b1329a-1d04-a325-1600-236ca577cc6a",
  "ItemOfInterest":"HybrisMarketing",
  "InteractionIntrstWeightingFctr":1,
  "InteractionIntrstSentimentVal":1
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH Interactions(InteractionUUID=guid'60b1329a-1d04-a325-1600-236ca577cc6a)
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
  "InteractionSourceObject":"22222",
  "InteractionUUID":"60b1329a-1d04-a325-1600-236ca577cc6a",
  "InteractionContactId":"4711",
  "MarketingArea":"CXXGLOBAL",
  "CampaignID":"0000033333",
  "InteractionLatitude":"70.304864",
  "InteractionLongitude":"7.228967",
  "InteractionLanguage":"DE",
  "CommunicationMedium":"WEB",
  "InteractionType":"WEBSITE_VISIT"
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH InteractionProducts(InteractionProductUUID=guid'63b1329a-1d04-
a325-1600-236ca577cc6a') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json

```

```

{
  "InteractionUUID":"60b1329a-1d04-a325-1600-236ca577cc6a",
  "ProductOrigin":"SAP_HYBRIS_PRODUCT",
  "Product":"PRD-2",
  "InteractionProdWeightingFactor":3,
  "InteractionProductSentimentVal":3
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH InteractionInterests (InteractionInterestUUID=guid'8aac579a-1d04-
a325-1600-236ca577cc6a') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
  "InteractionUUID":"60b1329a-1d04-a325-1600-236ca577cc6a",
  "ItemOfInterest":"BigData",
  "InteractionIntrstWeightingFctr":1,
  "InteractionIntrstSentimentVal":2
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
InteractionProductCategories (InteractionProductCategoryUUID=guid'63b1329a-1d04
-a325-1600-236ca577cc6a') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
  "InteractionUUID":"60b1329a-1d04-a325-1600-236ca577cc6a"
}

--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## GET Requests

### Get the first 10 Interactions sorted by InteractionTimeStamp

```

/sap/opu/odata/sap/API_MKT_INTERACTION_srv/Interactions?$top=10&
$orderby=InteractionTimeStampUTC desc

```

### Get the first 3 Interactions sorted by InteractionType

```

/sap/opu/odata/sap/API_MKT_INTERACTION_srv/Interactions?$filter=InteractionType eq
'OFFER_CLICK'&$top=3

```

### Get Interaction for specific InteractionUUID

#### i Note

This returns the interaction with the specified UUID, however there must be a number provided in the TOP Parameter

```

/sap/opu/odata/sap/API_MKT_INTERACTION_srv/Interactions?$filter=(InteractionUUID eq
guid'00000063-4657-49c8-1500-8f6c10e2ef5b')&$top=10

```

## Get the first 20 Interactions filtered by multiple entity types

You can filter by InteractionType, CommunicationMedium, InteractionSourceObject, InteractionDeviceName, InteractionContactOrigin and InteractionContactId and Interaction. Subnodes such as InteractionInterests, InteractionDigitalAssets, InteractionOffers, InteractionTags and InteractionAdditionalObjects will be expanded.

```
/sap/opu/odata/sap/API_MKT_INTERACTION_SRV/Interactions?
$expand=InteractionInterests,InteractionDigitalAssets,InteractionOffers,Interaction
Tags,InteractionAdditionalObjects$filter=((InteractionType eq 'EMAIL_OUTBOUND') and
(CommunicationMedium eq 'EMAIL') and (InteractionSourceObject eq 'XXX') and
(InteractionDeviceName eq 'XXX') and (InteractionContactOrigin eq 'XXX') and
(InteractionContactId eq '2612d9ed1631856b'))&$skip=0&$top=20&
$orderby=InteractionSourceObject&$inlinecount=allpages
```

## Get the first 10 Interactions within a specific time slot filtered by InteractionTimeStampUTC

```
/sap/opu/odata/sap/API_MKT_INTERACTION_srv/Interactions?
$filter=((InteractionTimeStampUTC gt datetimeoffset'2010-06-10T22:00:00.165') and
(InteractionTimeStampUTC le datetimeoffset'2018-06-10T22:00:00.165'))&$top=10
```

## Offer Redemption - Entity Set: InteractionOffers

### i Note

For an interaction of type OFFER\_REDEMPTION it is necessary to fill the InteractionOffers entity. At least one of the following property combinations have to be provided:

Single coupon codes:	Multi coupon codes:
MarketingOffer	MarketingOffer, Coupon Code
ExternalOffer, ExternalOfferOrigin	MarketingOffer, Serial Number
Coupon	ExternalOffer, ExternalOfferOrigin, CouponCode
-	ExternalOffer, ExternalOfferOrigin, SerialNumber
-	Coupon, CouponCode
-	Coupon, SerialNumber

- There is no dedicated property for the "SerialNumber"; it is imported using the property "CouponCode" with "CouponCodeType" set to "S").
- You can import several redemptions for a dedicated coupon code using the property "NumberOfRedemption". You can also cancel redemptions by setting this property to a negative value.

## Single Coupon Code, Import with External Offer and External Offer Origin

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/Interactions
POST data:
{
```

```

"InteractionTimeStampUTC": "2018-10-10T08:15:54",
"InteractionContactId": "john.doe@company.com",
"InteractionContactOrigin": "EMAIL",
"CommunicationMedium": "ONLINE_SHOP",
"InteractionType": "OFFER_REDEMPTION",
"InteractionIsAnonymous": false,
"InteractionOffers": [{
  "ExternalOffer": "offer_3452",
  "ExternalOfferOrigin": "EXT_SYSTEM"
}]
}

```

### Single Coupon Code, Import with Coupon (Redemption Cancellation)

#### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/Interactions
POST data:
{
  "InteractionTimeStampUTC": "2018-10-10T07:15:54",
  "InteractionContactId": "john.doe@company.com",
  "InteractionContactOrigin": "EMAIL",
  "CommunicationMedium": "ONLINE_SHOP",
  "InteractionType": "OFFER_REDEMPTION",
  "InteractionIsAnonymous": false,
  "InteractionOffers": [{
    "Coupon": "single_coupon564",
    "NumberOfRedemption": -1
  }]
}

```

### Multi Coupon Code, Import with Coupon and Serial Number

#### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/Interactions
POST data:
{
  "InteractionTimeStampUTC": "2018-10-10T08:13:23",
  "InteractionContactId": "john.doe@company.com",
  "InteractionContactOrigin": "EMAIL",
  "CommunicationMedium": "ONLINE_SHOP",
  "InteractionType": "OFFER_REDEMPTION",
  "InteractionIsAnonymous": false,
  "InteractionOffers": [{
    "Coupon": "multi_coupon241",
    "CouponCode": "SN4421",
    "CouponCodeType": "S",
    "NumberOfRedemption": 1
  }]
}

```

### Multi Coupon Code, Import with External Offer, External Offer Origin and Coupon Code

#### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/Interactions

```

```

POST data:
{
  "InteractionTimeStampUTC": "2018-10-09T09:34:11",
  "InteractionContactId": "john.doe@company.com",
  "InteractionContactOrigin": "EMAIL",
  "CommunicationMedium": "ONLINE_SHOP",
  "InteractionType": "OFFER_REDEMPTION",
  "InteractionIsAnonymous": false,
  "InteractionOffers": [{
    "ExternalOffer": "offer_5526",
    "ExternalOfferOrigin": "EXT_SYSTEM2",
    "CouponCode": "CODE33511",
    "CouponCodeType": ""
  }]
}

```

## Event Subnode

### Post Interaction and Event Subnode Non Batch Mode:

#### Sample Code

```

POST:
https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_INTERACTION_SRV/
Interactions
POST data:
{"InteractionContactOrigin":"EMAIL","InteractionContactId":"20190927151824_api
test@teamwdf02.de","CommunicationMedium":"EMAIL","InteractionType":"EMAIL_BOU
CE_SOFT","InteractionTimeStampUTC":"2019-09-27T15:18:24","InteractionSourceOb
jectType":"ERP","InteractionSourceObject":"12345678","MarketingArea":"ICMA_DRIN
K","InteractionEvents":[{"MktgEventExternalId":"WDF02
1E2019-09-27T15:18:16","MktgEventProvider":"ON24_ID","MktgEventProviderAccount
":"123"}]}

```

### Post Interaction Event Subnode DeepInsert:

#### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/InteractionsDeepInsert
POST data:
{"Interactions":
[{"InteractionContactOrigin":"EMAIL","InteractionContactId":"20190927151822_ap
itest@teamwdf02.de","CommunicationMedium":"EMAIL","InteractionType":"EMAIL_BOU
NCE_SOFT","InteractionTimeStampUTC":"2019-09-27T15:18:22","InteractionSourceOb
jectType":"ERP","InteractionSourceObject":"12345678","MarketingArea":"ICMA_DRI
NK","InteractionEvents":[{"MktgEventExternalId":"WDF02
2E2019-09-27T15:18:16","MktgEventProvider":"ON24_ID","MktgEventProviderAccount
":"123"}]}]}

```

### Post Interaction Event Subnode Batch Mode:

#### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/$batch

```



```

POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InteractionEvents HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"InteractionUUID":"d11415a6-e69b-64bb-
b8d7-1667395b10d6","InteractionEventUUID":"6c0b84b7-5523-1ed9-
b8a7-40fa9950e7ce","MktgEventExternalId":"WDF02
3E2019-09-27T15:18:16","MktgEventProvider":"ON24_ID","MktgEventProviderAccount
":"123"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch-

```

## POST Interaction Event Subnode Non Batch Mode

### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/InteractionEvents
POST data:
{"InteractionUUID":"d11415a6-e69b-64bb-
b8d7-1667395b10d6","InteractionEventUUID":"6c0b84b7-5523-1ed9-
b8a7-40fa995107ce","MktgEventExternalId":"WDF02
4E2019-09-27T15:18:16","MktgEventProvider":"ON24_ID","MktgEventProviderAccount
":"123"}

```

## Update Interaction Event Sub Node Non Batch Mode:

### Sample Code

```

PATCH https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/
InteractionEvents(InteractionEventUUID=guid'24c63568-4daf-3dd6-1600-236c2e825c
e5')
PATCH data:
{"MktgEventAttendanceType":"01"}

```

## Update Interaction Event Subnode Batch Mode:

### Sample Code

```

POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH
InteractionEvents(InteractionEventUUID=guid'24c63568-4daf-3dd6-1600-236c2e825c
e5') HTTP/1.1
Content-Length: 1035

```

```
Accept: application/json
Content-Type: application/json
{"MktgEventAttendanceType":"02"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

### Delete Interaction Event Subnode Non Batch Mode:

#### Sample Code

```
DELETE https://wdciwel.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/
InteractionEvents(InteractionEventUUID=guid'24c63568-4daf-3dd6-1600-236c2e825c
e5')
```

### Delete Interaction Event Subnode Batch Mode

#### Sample Code

```
POST https://wdciwel.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTERACTION_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
InteractionEvents(InteractionEventUUID=guid'24c63568-4daf-3dd6-1600-236c2e825c
e5') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"InteractionUUID":"d11415a6-e69b-64bb-b8d7-1667395b10d6"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

Parent topic: [Interactions \[page 615\]](#)

## Related Information

[Basic Concepts \[page 617\]](#)

[Structure of OData Service API\\_MKT\\_INTERACTION \[page 621\]](#)

[Error Handling for Interactions \[page 647\]](#)

## 5.2.8.4 Error Handling for Interactions

This section contains some troubleshooting tips for handling common errors involving interaction imports.

Error / Error Message	What It Means	What You Can Do
Invalid entry in column &2: '&1'	Either: <ul style="list-style-type: none"><li>• Configuration is missing</li><li>• Invalid characters have been used</li><li>• The format is incorrect</li></ul>	Maintain Configuration Correct invalid entries: for example Phone numbers have to start with '+' or '00': <div data-bbox="1007 689 1396 808"><b>i Note</b> Refer to the message long text.</div>
ODATA error: The metadata do not allow a null value.	The value of the UUID cannot be empty.	To prevent an ODATA error message, you must set a fixed initial value <b>00000000-0000-0000-0000-000000000000</b> for the UUID of all nodes, both root and subnodes.
Timestamp is incorrect	Microsoft Excel formats the timestamp incorrectly when opening the file.	Format the value in Microsoft Excel as follows: <ol style="list-style-type: none"><li>1. Change the type of the field to <b>number</b>.</li><li>2. Remove all decimal places.</li><li>3. Remove the separator.</li><li>4. Enter a valid time (YYYYMMDDhhmmss).</li><li>5. Save</li></ol>
Special characters are not imported	The data is not loaded in utf-8 encoding.	Send data in utf-8 encoding. Use an editor that supports this.

Parent topic: [Interactions \[page 615\]](#)

### Related Information

[Basic Concepts \[page 617\]](#)

[Structure of OData Service API\\_MKT\\_INTERACTION \[page 621\]](#)

[Payload Examples for Interactions \[page 633\]](#)

[HTTP Response Status Codes \[page 408\]](#)

[Best Practices and Recommended Package Sizes \[page 400\]](#)

[Import Monitor \[page 404\]](#)

## 5.2.9 Interest Items

Public OData API (API\_MKT\_INTEREST\_SRV) for InterestItems. An interest represents the content or subject of a contact's interaction.

### Overview

The public API for InterestItems supports operations on the Business Object CUAN\_INTEREST.

OData Version	2.0
Root URI	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTEREST_SRV</code>
Service Metadata URI:	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/\$metadata</code>
Authorizations	The following business catalog role is required:
Communication Scenario ID	SAP_COM_0340
Component for Incidents	CEC-MKT-DM-IA

**i Note**

Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).



### Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

#### i Note

You can convert the XML file to an XML table to make it easier to read.

Access Link	Remarks
<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_Interests_SRV/\$metadata?sap-documentation=all</code>	Only for internal access. You need to provide the server and port name

Access Link	Remarks
<a href="https://api.sap.com/api/API_MKT_INTEREST_SRV/overview">https://api.sap.com/api/API_MKT_INTEREST_SRV/overview</a>  Marketing - Interest Items Integration Page	<p>General access to the <i>Interest Item Integration</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>1. On the <i>Interest Item Integration</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>2. Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="https://api.sap.com/api/API_MKT_INTEREST_SRV/overview">https://api.sap.com/api/API_MKT_INTEREST_SRV/overview</a> 	<p>General access to the <i>Interest Items</i> metadata file. One-time registration or logon is required.</p>

## 5.2.9.1 Structure of API\_MKT\_INTEREST\_SRV

This document describes the structure of the Public OData API service `API_MKT_INTEREST`. Make sure you read the Basic Concepts topic before you start.

The InterestItem OData API provides the following entity sets:

Entity Set	Description	Path
InterestItems	This entity contains interest items data.	/InterestItems
InterestItemProdCats	This entity contains the product categories of interest items. Product categories can be assigned to products so related products can be grouped together.	/InterestItemProdCats
InterestItemTags	This entity contains the tags assigned to interests. Tags are terms or groups of terms assigned to an entity to help classify its content.	/InterestTags
InterestItemTexts	This entity contains the language specifications of an interest item text.	/InterestItemTexts

## Entity Sets

### InterestItems

**Entity Path:** `/InterestItems`

You can perform the following operations on the `InterestItem` entity:

HTTP Method	Description	Path
GET	Return a list of InterestItems. This method supports standard OData parameters such as \$filter, \$select,	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItems</code>
	Get the top 2 InterestItems	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItems?\$top=2</code>
	Return specific InterestItem "xxx"	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItems?\$filter=ItemOfInterest eq 'xxx'</code>
	Return a list of InterestItems omitting the top 2 InterestItems	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItems?\$skip=2</code>
	Return a list of InterestItemTexts assigned to the specified InterestItem "xxx"	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItems('xxx')/InterestItemTexts</code>
	Return a list of InterestItemTags assigned to the specified InterestItem "xxx"	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItems('xxx')/InterestItemTags</code>
	Returns a list of InterestItemProdCats assigned to the specified InterestItem "xxx"	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItems('xxx')/InterestItemProdCats</code>
	Return details of a specified InterestItem "xxx"	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItems('xxx')</code>

## InterestItemProdCats

Entity Path: `/InterestItemProdCats`

You can perform the following operations on the `InterestItemProdCat` entity:

HTTP Method	Description	Path
GET	Return a list of <code>InterestItemProdCat</code> . This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code>	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItemProdCats/</code>

## InterestItemTexts

Entity Path: `/InterestItemText`

You can perform the following operations on the `InterestItemText` entity:

HTTP Method	Description	Path
GET	Return a list of <code>InterestItemTextst</code> . This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code>	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItemTexts</code>

## InterestItemTags

Entity Path: `/InterestItemTags`

You can perform the following operations on the `InterestItemTags` entity:

HTTP Method	Description	Path
GET	Return a list of <code>InterestItemTags</code> . This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code>	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItemTags</code>
	Return a list of <code>InterestItems</code> with the <code>Tag Type</code> field provided	<code>/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/InterestItemTags?\$select=TagType</code>

## 5.2.9.2 Payload Examples for Interest Items

Payload examples for `API_MKT_INTEREST`.

The following examples demonstrate how you can use the Interests API.

Method		Payload Examples - Batch Mode	Payload Examples - Non-Batch Mode
Deep Create Interests		<a href="#">Deep Create Interests Batch Mode [page 652]</a>	<a href="#">Deep Create Interests Non-Batch Mode [page 654]</a>
Create Interest		<a href="#">Create Interests (Non-Deep) Non-BATCH Mode [page 654]</a>	<a href="#">Create Interests (Non-Deep) Non-BATCH Mode [page 655]</a>
Create Item of Interest Assignments	Create Text Assignments	<a href="#">Create Interest Item Text Assignments in Batch Mode [page 655]</a>	<a href="#">Create Interest Item Text Assignments in Non-Batch Mode [page 656]</a>
	Create Tag Assignments	<a href="#">Create Interest Item tag Assignments in Batch Mode [page 656]</a>	<a href="#">Create Interest Item tag Assignments in Non-Batch Mode [page 656]</a>
	Create Product Category Assignments	<a href="#">Create Interest Item Product Category Assignments in Batch Mode [page 657]</a>	<a href="#">Create Interest Item Product Category Assignments in Non-Batch Mode [page 657]</a>
Delete Item Of Interest		<a href="#">Delete Interest items in Batch Mode [page 657]</a>	<a href="#">Delete Interest items in Non-Batch Mode [page 658]</a>
Delete Item of Interest Assignments	Delete Text Assignments	<a href="#">Delete Interest Items Text Assignment in Batch Mode [page 658]</a>	<a href="#">Delete Interest Item Text Assignment in Non-Batch Mode [page 658]</a>
	Delete Tag Assignments	<a href="#">Delete Interest Item Tag Assignments in Batch Mode [page 659]</a>	<a href="#">Delete Interest Item tag Assignments in Non-Batch Mode [page 659]</a>
	Delete Product Category Assignments	<a href="#">Delete Interest Item Product Category Assignments in Batch Mode [page 659]</a>	<a href="#">Delete Interest Item Product Category Assignments in Non-Batch Mode [page 660]</a>
Update Interest Text Assignments	Update Interest Text Assignments	<a href="#">Update Interest Text Assignments in Batch Mode [page 660]</a>	<a href="#">Update Interest Text Assignments in Non-Batch Mode [page 660]</a>

## Deep Create Interests Batch Mode

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InterestItems HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
```



```

"ItemOfInterest":"T220190325153232",
"InterestItemTexts":
[
  {
    "Language":"EN",
    "ItemOfInterest":"T220190325153232",
    "ItemOfInterestName":"Test API Interest Srv"
  },
  {
    "Language":"DE",
    "ItemOfInterest":"T220190325153232",
    "ItemOfInterestName":"Deutsch Test"
  }
],
"InterestItemTags":
[
  {
    "TagName":"gallo d'oro",
    "ItemOfInterest":"T220190325153232"
  }
],
"InterestItemProdCats":
[
  {
    "ProductCategoryHierarchy":"TEAM_WDF02_JS",
    "ProductCategory":"IOI_API_CAT_EF20190325153225",
    "ItemOfInterest":"T220190325153232"
  }
]
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InterestItems HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{
"ItemOfInterest":"T320190325153232",
"InterestItemTexts":
[
  {
    "Language":"EN",
    "ItemOfInterest":"T320190325153232",
    "ItemOfInterestName":"Test API Interest Srv"
  },
  {
    "Language":"DE",
    "ItemOfInterest":"T320190325153232",
    "ItemOfInterestName":"Deutsch Test"
  }
],
"InterestItemTags":
[
  {
    "TagName":"#taste",
    "ItemOfInterest":"T320190325153232"
  }
]
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Deep Create Interests Non-Batch Mode

### Sample Code

```
POST: https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/InterestItems
POST data:
{
  "ItemOfInterest": "T3320190325153242",
  "InterestItemTexts":
  [
    {
      "Language": "EN",
      "ItemOfInterest": "T3320190325153242",
      "ItemOfInterestName": "Test API Interest Srv"
    },
    {
      "Language": "DE",
      "ItemOfInterest": "T3320190325153242",
      "ItemOfInterestName": "Deutsch Test"
    }
  ],
  "InterestItemTags":
  [
    {
      "TagName": "best",
      "ItemOfInterest": "T3320190325153242"
    },
    {
      "TagName": "gallo d'oro",
      "ItemOfInterest": "T3320190325153242"
    }
  ],
  "InterestItemProdCats":
  [
    {
      "ProductCategoryHierarchy": "TEAM_WDF02_JS",
      "ProductCategory": "IOI_API_CAT_MM20190325153228",
      "ItemOfInterest": "T3320190325153242"
    },
    {
      "ProductCategoryHierarchy": "TEAM_WDF02_JS",
      "ProductCategory": "IOI_API_CAT_oo20190325153229",
      "ItemOfInterest": "T3320190325153242"
    }
  ]
}
```

## Create Interests (Non-Deep) BATCH Mode

### Sample Code

```
Post: https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type: multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
```

```
content-type: application/http
content-transfer-encoding: binary
POST InterestItems HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
  {"ItemOfInterest":"T7720190325153247"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Create Interests (Non-Deep) Non-BATCH Mode

```
POST: https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_INTEREST_SRV/
InterestItems
POST data:
{"ItemOfInterest":"T9920190325153230"}
```

## Create Interest Item Text Assignments in Batch Mode

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InterestItemTexts HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"ItemOfInterest":"T93220190516122937","Language":"EN","ItemOfInterestName":"E
nglish Text Assinment"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InterestItemTexts HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"ItemOfInterest":"T93220190516122937","Language":"CS","ItemOfInterestName":"C
zech test"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Create Interest Item Text Assignments in Non-Batch Mode

### Sample Code

```
POST https://wdciwe1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/InterestItemTexts
POST data:
{"ItemOfInterest":"T91720190516123049","Language":"EN","ItemOfInterestName":"E
nglish Text Assinment"}
```

## Create Interest Item tag Assignments in Batch Mode

### Sample Code

```
POST https://wdciwe1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InterestItemTags HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"ItemOfInterest":"T67620190516123147","TagName":"#sapbyd"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InterestItemTags HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"ItemOfInterest":"T67620190516123147","TagName":"hamburg"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Create Interest Item tag Assignments in Non-Batch Mode

```
POST https://wdciwe1.wdf.sap.corp:60100/sap/opu/odata/sap/API_MKT_INTEREST_SRV/
InterestItemTags
POST data:
{"ItemOfInterest":"T0020190516123150","TagName":"#sapbyd"}
```

## Create Interest Item Product Category Assignments in Batch Mode

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST InterestItemProdCats HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"ItemOfInterest":"T60020190516123503","ProductCategoryHierarchy":"TEAM_WDF02_
JS","ProductCategory":"IOI_API_CAT_8820190516123502"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Create Interest Item Product Category Assignments in Non-Batch Mode

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/InterestItemProdCats
POST data:
{"ItemOfInterest":"T0020190516123506","ProductCategoryHierarchy":"TEAM_WDF02_J
S","ProductCategory":"IOI_API_CAT_9920190516123503"}
```

## Delete Interest items in Batch Mode

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE InterestItems(ItemOfInterest='T24420190516122658') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
```

```

content-transfer-encoding: binary
DELETE InterestItems (ItemOfInterest='T25520190516122658') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch-

```

## Delete Interest items in Non-Batch Mode

### Sample Code

```

HTTP Method: DELETE
http://ldciabd.wdf.sap.corp:50000/sap/opu/odata/SAP/API_MKT_INTEREST_SRV/
InterestItems (ItemOfInterest='T24420190515131554')

```

## Delete Interest Items Text Assignment in Batch Mode

### Sample Code

```

Delete IOI- Text Batch Mode:
POST https://wdciwe1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE InterestItemTexts (ItemOfInterest='T93220190927151105', Language='EN')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## Delete Interest Item Text Assignment in Non-Batch Mode

### Sample Code

```

DELETE https://wdciwe1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/ InterestItemTexts
(ItemOfInterest='T887720190927151109', Language='EN')

```

## Delete Interest Item Tag Assignments in Batch Mode

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
InterestItemTags (ItemOfInterest='T11223320190927151108', TagName='tagtodelete')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Delete Interest Item tag Assignments in Non-Batch Mode

### Sample Code

```
DELETE https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/
InterestItemTags (ItemOfInterest='T887720190927151109', TagName='tagtodelete')
```

## Delete Interest Item Product Category Assignments in Batch Mode

### Sample Code

```
POST https://wdciw1.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
DELETE
InterestItemProdCats (ItemOfInterest='T60020190927151058', ProductCategory='IOI_
API_CAT_8820190927151027', ProductCategoryHierarchy='TEAM_WDF02_JS') HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
```

```
--batch--
```

## Delete Interest Item Product Category Assignments in Non-Batch Mode

### Sample Code

```
DELETE https://wdciwel.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/ InterestItemProdCats
(ItemOfInterest='T887720190927151109',ProductCategory='IOI_API_CAT_88201909271
51027',ProductCategoryHierarchy='TEAM_WDF02_JS')
```

## Update Interest Text Assignments in Batch Mode

### Sample Code

```
POST https://wdciwel.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/$batch
POST data:
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
PATCH InterestItemTexts (ItemOfInterest='T93220201215103449',Language='AR')
HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
{"ItemOfInterestName":"Batch Update AR Text Assinment"}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--
```

## Update Interest Text Assignments in Non-Batch Mode

### Sample Code

```
PATCH https://wdciwel.wdf.sap.corp:60100/sap/opu/odata/sap/
API_MKT_INTEREST_SRV/
InterestItemTexts (ItemOfInterest='T93220201215103449',Language='EN')
PATCH data:
{"ItemOfInterestName":"NonBatch Update English Text Assinment"}
```



## 5.2.10 Business Documents

Public OData API (CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV) for importing business documents, such as leads and opportunities, from external SAP or non-SAP systems to SAP Marketing Cloud. Use this version of the service when you want to import business documents related to Offers and Coupons. Lower versions are not suitable for this purpose.

### i Note

We recommend using the current version 0003 of this service. If you want to use version 0001 or version 0002, you'll find more information under:

- Version 0001: [Business Documents API, Version 0001](#)
- Version 0002: [Business Documents API, Version 0002](#)

## Communication Scenarios

CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV can be used in the following communication scenarios:

Communication Scenario	Description
SAP_COM_0017 Marketing	Presales/Sales Integration
SAP_COM_0060 Marketing	ERP Order and Business Partner Integration
SAP_COM_0082 Marketing	SAP Commerce Data Integration
SAP_COM_0329 Marketing	Business Document Interaction Integration

For more information on how to set up the communication scenarios, see [Communication Management](#).

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ CUAN_BUSINESS_DOCUMENT_IMP_SRV /\$metadata?sap- documentation=all</code>	Only for internal access. You need to provide the server and port names.

Access Link	Remarks
<a href="#">Import of Business Documents Details Page</a>	<p>General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Business Documents OData API</a>	<p>General access link takes you directly to the <i>Business Documents</i> metadata file. One-time registration or logon is required.</p>
Component for Incidents	CEC-MKT-DM-IA

**i Note**

Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

[Overview \[page 662\]](#)

Import of business documents to marketing using the OData service  
CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV (Version 3).

[Structure of OData Service CUAN\\_BUSINESS\\_DOCUMENT\\_IMP\\_SRV \[page 664\]](#)

[Basic Concepts \[page 672\]](#)

[Modes \[page 674\]](#)

[Specifics for SAP Cloud for Customer Integration \[page 677\]](#)

[Payload Examples for Business Documents \[page 678\]](#)

This section contains payload examples for CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV.

## 5.2.10.1 Overview

Import of business documents to marketing using the OData service CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV (Version 3).

The OData service CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV is used to import business documents from external systems, that is SAP systems or non-SAP systems, into SAP Marketing Cloud . Each business

document is represented by an interaction and is identified by the key of the business document in the external system. The OData service `CUAN_BUSINESS_DOCUMENT_IMP_SRV` supports the change of interactions.

Furthermore, the OData service `CUAN_BUSINESS_DOCUMENT_IMP_SRV` is used for standard SAP Marketing Cloud integration with SAP Cloud for Customer. It is used in **marketing-driven**, and **sales-driven processes** to replicate SAP Cloud for Customer business documents to SAP Marketing Cloud interactions.

### **i** Note

In standard SAP Marketing Cloud integration with SAP Cloud for Customer OData service `CUAN_BUSINESS_PARTNER_IMP_SRV` is used to replicate customers of SAP Cloud for Customer to SAP Marketing Cloud contacts.

Within **marketing-driven processes**, SAP Marketing Cloud campaign actions are used to create business documents in SAP Cloud for Customer (leads, appointments, phone calls, and tasks). For each SAP Cloud for Customer business document, an SAP Marketing Cloud interaction is created in the campaign action. The business document of SAP Cloud for Customer is created with reference to the SAP Marketing Cloud interaction, that is, it stores the IDs of the SAP Marketing Cloud interaction and campaign. When an SAP Cloud for Customer business document is created, a **confirmation message** is returned, mapped to OData service `CUAN_BUSINESS_DOCUMENT_IMP_SRV`, and the SAP Marketing Cloud interaction is updated with the IDs of the SAP Cloud for Customer business document.

Within **sales-driven processes**, SAP Cloud for Customer business documents (leads, opportunities, appointments, visits, and phone calls) can be replicated to SAP Marketing Cloud to create corresponding interactions. Whenever a business document is created, or changed the Simple Object Access Protocol (SOAP) outbound service **request message** is triggered containing all business document data and mapped in SAP Cloud Integration middleware to OData service `CUAN_BUSINESS_DOCUMENT_IMP_SRV`. Request messages are also created in marketing-driven processes whenever the business document created via a campaign action is changed afterwards.

**Parent topic:** [Business Documents \[page 661\]](#)

## **Related Information**

[Structure of OData Service CUAN\\_BUSINESS\\_DOCUMENT\\_IMP\\_SRV \[page 664\]](#)

[Basic Concepts \[page 672\]](#)

[Modes \[page 674\]](#)

[Specifics for SAP Cloud for Customer Integration \[page 677\]](#)

[Payload Examples for Business Documents \[page 678\]](#)

## 5.2.10.2 Structure of OData Service CUAN\_BUSINESS\_DOCUMENT\_IMPORT\_SRV

The CUAN\_BUSINESS\_DOCUMENT\_IMPORT\_SRV (Version 3) OData service consists of the following entity sets and entity types:

Entity Sets and Entity Types

Entity Set	Entity Type	Entity Type Description
ImportHeaders	ImportHeader	Technical Import Message Header
BusinessDocuments	BusinessDocument	Business Document
ProductItems	ProductItem	Product Items
Offers	Offer	Offers
AdditionalObjectReferences	AdditionalObjectReference	Additional object reference of an interaction
ProductCategories	ProductCategory	Product Category
MarketingArea	MarketingArea	Marketing Area
LoyaltyProgram	LoyaltyProgram	Loyalty Program
AdditionalInteractionContacts	AdditionalInteractionContact	Additional Interaction Contact

### Note

If you use *OData* in SAP Cloud Integration then fill the key fields with an initial value. This is required since the OData adapter requires that mandatory OData fields are filled.

## Entity Types

### ImportHeader

The entity type *ImportHeader* describes the technical header of an import of multiple business documents. The properties *ID* and *Timestamp* are used for logging the external data request. If an error occurs during the posting of the business documents, additionally to the import header data the error message and the failed record are saved. This data can be checked with the *Import Monitor* app. For more information, see [Import Monitor \[page 404\]](#).

If no ID or timestamp values are provided, they are defaulted internally. If you do provide an ID, it must be unique.

The *SourceSystemID* and *SourceSystemType* properties allow you to distinguish between different source systems. The *SourceSystemID* and *SourceSystemType* are mandatory attributes. This is the semantic key of an interaction with the communication medium BUSINESS\_DOCUMENT.

### i Note

The value *C4C* for the *SourceSystemType* is exclusively for the integration to SAP Cloud for Customer.

Properties of ImportHeader

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ID	Unique technical identifier of import run.	Edm.String	32	x	x
TimeStamp	Timestamp of the run	Edm.DateTime	0	x	x
SourceSystemType	Type of the source system, that is ERP	Edm.String	20	x	x
SourceSystemId	Identifier of the source system	Edm.String	23	x	x

## BusinessDocument

For each *ImportHeader*, several business documents can be passed. *BusinessDocument* is mapped to interactions.

Properties of BusinessDocument

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ID	Key of the interaction that is updated. Only mandatory when <i>ActionCode = 02</i>	Edm.Guid	0	x	x

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ContactIDOrigin	ID origin of the contact. If the ContactID is filled the ID origin from the external system must be set, that is, SAP_C4C_BUPA. Mandatory when ActionCode = 04	Edm.String	20	X1	
ContactId	ID of the contact in the external system. Mandatory when ActionCode = 04	Edm.String	255	X1	
InternalContactID	obsolete	Edm.String	255		
InternalObjectType	Interaction type only mandatory when ActionCode = 04	Edm.String	20	X1	
ExternalObjectType	Type of the external object, that is, MARKETING_LEAD	Edm.String	30	x	
ExternalId	ID of the external object	Edm.String	50	x	
ExternalStatusCode	Status of the external object	Edm.String	2		
ExternalTimestamp	Timestamp of the external object. Timestamp is used to process messages in the right sequence	Edm.DateTimeOffset	2	x	
Content	Long description of the external object	Edm.String	0		
ExpectedRevenue	Expected revenue of an opportunity	Edm.Decimal	31,2		

Property	Description	Edm Core Type	Max Length	Mandatory	Key
Currency	Currency of the amount fields. Must be set if ExpectedRevenue Or Amount is filled	Edm.String	5		
PredecessorId	Lead predecessor ID can be passed. Campaign is copied from predecessor document. Can only be used on request mode.	Edm.String	50		
ActionCode	The action code controls how an interaction is posted. The following values are supported:  02 confirmation mode  04 request mode  05 remove	Edm.String	2	x	
ContentTitle	Short description of the external object	Edm.String	255		
EndTimeStamp	End Time Stamp	Edm.DateTimeOffset	0		
Reason	Reason	Edm.String	20		
ExternalAdditionalId	Additional ID of the external object	Edm.String	50		
Amount	Amount of the external object. Currency field must be filled if amount is populated.	Edm.Decimal	0		

Property	Description	Edm Core Type	Max Length	Mandatory	Key
StatusCode	<p>Internal status of the interaction. The following statuses are defined:</p> <p>00 New</p> <p>01 In Process</p> <p>02 Released</p> <p>03 Completed</p> <p>04 Canceled</p> <p>05 Converted</p> <p>06 Successful</p> <p>07 Unsuccessful</p>	Edm.String	2		
Timestamp	<p>Timestamp of external object in the source system, that is, order date. Filled with ExternalTimeStamp if empty</p>	Edm.DateTimeOffset	0		
CampaignId	<p>SAP Marketing Cloud Campaign ID</p> <p>If a campaign ID is passed the field <i>PredecessorId</i> is not evaluated. For more information, see <a href="#">Modes [page 674]</a>, section <i>PredecessorId (Marketing-Driven Process)</i></p>	Edm.String	10		
MarketingAreaId	Marketing area ID of the interaction	Edm.String	40		
MarketingLocation	Marketing location ID of the interaction	Edm.String	50		
MarketingLocationOrigin	Origin of marketing location ID	Edm.String	30		



Property	Description	Edm Core Type	Max Length	Mandatory	Key
MKT_AgreementOrigin	Agreement Origin	Edm.String	30		
MKT_AgreementExternalID	Agreement External ID	Edm.String	80		
InteractionProcessingDuration	Processing time in seconds	Edm.Int32	0		
InteractionPriority	Priority	Edm.String	1		

### Note

The value *X1* is only mandatory when the field `ActionCode` has the value 04.

## ProductItem

Several product items can be passed per business document.

Properties of Entity Type ProductItem

Property	Description	Edm Core Type	Max Length	Mandatory	Key
Id	Item number for product item from external system	Edm.String	32	X1	x
ObjectType	Object type from external system, that is, SAP_C4C_PRODUCT	Edm.String	30	x	x
ObjectId	Product key from external system	Edm.String	50	x	
Amount	Amount of product. Currency information is derived from assigned interaction-	Edm.Decimal	31,2		
Quantity	Quantity of product	Edm.Decimal	22,5		
Unity of Measure	Unit of product	Edm.String	3		

With version 3 of `CUAN_BUSINESS_DOCUMENT_IMP_SRV`, a product cannot be created as a master product. The product must exist in the system. It has to be imported beforehand using one of the provided services, for example, `API_MKT_PRODUCT`. The properties **ObjectType** and **ObjectId** refer to the properties **ProductType**

and **ProductID** of the service API\_MKT\_PRODUCT. If you upload a product that does not yet exist in the system you will receive an error.

X1 initial value can be passed. The value of the field is not persisted.

## Offer

Several Offers can be passed per business document.

The system determines the corresponding Offer ID and Coupon ID **only** when the ExternalObjectType is SALES\_ORDER. In the case of all other external object types the data sent is stored unchecked, exactly as it is provided. The system does not determine the Offer ID or the Coupon ID.

Properties of Entity Type Offer

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ExternalOfferOrigin	Origin of the offer	Edm.String	30	x	x
ExternalOffer	Identifier of the external offer	Edm.String	60	x	x
MarketingOffer	ID of the marketing offer	Edm.String	10		
MarketingOfferContent	Content Item Number	Edm.String	5		
Coupon	Coupon	Edm.String	32		
CouponCodeType	Coupon Code Type	Edm.String	1		
CouponCode	Coupon Code	Edm.String	128		
NumberOfRedemption	Number of Redemption	Edm.Int32			

## Additional Object Reference

With entity type [AdditionalObjectReference](#), several additional object references per business document can be passed.

Properties of Entity Type AdditionalObjectReference

Property	Description	Edm Core Type	Max Length	Mandatory	Key
ObjectType	Type of the object referenced	Edm.String	30	x	x
ObjectId	ID of the object referenced	50	50	x	x

## Product Category

With entity type [ProductCategory](#), several product categories per business document can be passed.

Properties of Entity Type ProductCategory

Property	Description	Edm Core Type	Max Length	Mandatory	Key
Id	ID of the product category	Edm.String	50	x	x
Hierarchy	ID of product category hierarchy	50	50	x	x

Before you can upload references of product categories that are assigned to business documents, the master data of the product category and product category hierarchy must be uploaded.

### Marketing Area

This entity can only be used in standard SAP Marketing Cloud integration with SAP Cloud for Customer for replication of leads. For more details see [Specifics for SAP Cloud for Customer Integration \[page 677\]](#) Several marketing areas per business document can be passed.

Property	Description	Edm Core Type	Max Length	Mandatory	Key
MarketingAreaId	Marketing Area Id	Edm.String	40	x	x

Before you can upload references of product categories that are assigned to business documents the master data of the product category, and product category hierarchy must be uploaded.

### Loyalty Program

With entity type *LoyaltyProgram*, several Loyalty Programs can be passed per business document entity.

Properties of Entity Type LoyaltyProgram

Property	Description	Edm Core Type	Max Length	Mandatory	Key
LoyaltyUUID	Loyalty Program GUID	Edm.Guid	0	X1	X
ActivitySubtype	Loyalty Activity Subtype	Edm.String	255		
AccruedPoints	Accrued Loyalty Points	Edm.Decimal	31,2		
RedeemedPoints	Redeemed Loyalty Points	Edm.Decimal	31,2		
QualifyingPoints	Qualifying Loyalty Points	Edm.Decimal	31,2		

X1 Initial GUID must be passed "00000000-0000-0000-0000-000000000000".

### Additional Interaction Contacts

With entity type *AdditionalInteractionContact*, several additional interaction contact references can be passed per business document entity. So, several interaction contacts can be assigned to one interaction. The

referenced interaction contact must exist. If the contact does not exist message status is set to blocked and the message is reprocessed. A contact might not exist because the master data message for the interaction contact is delayed.

Properties of Entity AdditionalObjectReference

Property	Description	Edm Core Type	Max Length	Mandatory	Key
InteractionAdditionalIntactnContactUUID	Interaction Additional Interaction Contact UUID	Edm.Guid	0	X1	X
ActiInteractionContactOriginvitySubtype	Contact Origin	Edm.String	20		
InteractionContactId	Contact Id	Edm.Decimal	255		
InteractionContactUUID	Interaction Contact UUID	Edm.Guid	0		

X1 Initial GUID must be passed "00000000-0000-0000-0000-000000000000".

Parent topic: [Business Documents \[page 661\]](#)

## Related Information

[Overview \[page 662\]](#)

[Basic Concepts \[page 672\]](#)

[Modes \[page 674\]](#)

[Specifics for SAP Cloud for Customer Integration \[page 677\]](#)

[Payload Examples for Business Documents \[page 678\]](#)

### 5.2.10.3 Basic Concepts

The OData service `CUAN_BUSINESS_DOCUMENT_IMPORT_SRV` (Version 3) supports only the method *deep create* on the entity type *ImportHeader* and the dependent entity type *BusinessDocument*. Other methods, such as *create*, *update* or *delete* are not supported. The field *ActionCode* controls how a *BusinessDocument* is processed. For more information, see [Modes \[page 674\]](#).

OData service `CUAN_BUSINESS_DOCUMENT_IMPORT_SRV` supports one message of the entity type *ImportHeader* with multiple lines of the entity type *BusinessDocument*. Up to 10,000 business documents can be sent at once with the OData service `CUAN_BUSINESS_DOCUMENT_IMPORT_SRV`.

#### Update Logic

You must send a complete snapshot of the data every time. It is not possible to send only data that has changed with respect to a previous state. A new upload always overwrites the existing record.

The snapshot must also contain all entries for all sub-entity sets, for example, `ProductItems`, `Offers`, and `AdditionalObjectReferences`. If no entries are provided for a sub-entity set, any existing entries are deleted.

### Timestamp

The Attribute `ExternalTimeStamp` of entity `BusinessDocument` defines when the business document was last changed or created. Before updating the record, the system compares the timestamp (`TIMESTAMP`) of the interaction in SAP Marketing Cloud and the timestamp of the incoming message (`ExternalTimeStamp`). Outdated messages, where the value of the field `ExternalTimeStamp` is lower than the field `TIMESTAMP` of the interaction, are ignored since changes are already stored, and the most recent interaction carries the complete and most recent snapshot data.

### Error Handling

If the OData service is not accessible (for example no authorization, system not available, too many business documents sent) a corresponding HTTP status code is returned. After the OData service has been accepted by the Gateway component in SAP Marketing Cloud, the HTTP status code 201 is always returned. Any processing errors are recorded in the SAP Marketing Cloud system and can be monitored, restarted, or discarded in the *Import Monitor* app. For more information see, [Import Monitor \[page 404\]](#) and [HTTP Response Status Codes \[page 408\]](#).

### Semantic Key

This refers to the semantic key of an interaction that has the communication medium `BUSINESS_DOCUMENT`. The external key of a `BusinessDocument` entity is defined by unique combination of the fields:

- `SourceSystemId`
- `SourceSystemType`
- `ExternalObjectType`
- `ExternalObjectId`

**Parent topic:** [Business Documents \[page 661\]](#)

## Related Information

[Overview \[page 662\]](#)

[Structure of OData Service CUAN\\_BUSINESS\\_DOCUMENT\\_IMP\\_SRV \[page 664\]](#)

[Modes \[page 674\]](#)

[Specifics for SAP Cloud for Customer Integration \[page 677\]](#)

[Payload Examples for Business Documents \[page 678\]](#)

## 5.2.10.4 Modes

As described in the overview chapter, the OData service `CUAN_BUSINESS_DOCUMENT_IMP_SRV` supports confirmation and request messages.

In request mode, messages are processed that import (create, update, or delete) interactions. The messages are triggered by changes in the external system.

The confirmation mode is used in marketing-driven scenarios to confirm messages that were sent to external systems. The confirmation message mainly contains the key of the business document in the external system.

The different modes are defined by attribute `ActionCode` in entity `BusinessDocument`. Action code 04 defines creation or update in request mode. Action code 05 defines deletion in request mode. Action code 02 defines the update of an interaction by the external key in confirmation mode.

Customer enhancements are only supported when the field `ActionCode` has the value 04.

See the following chapters for more details.

### Request Mode: Create and Update (ActionCode 04)

The external key (see table below) of the business document is used to check the existence of an interaction in SAP Marketing Cloud. If an interaction is found it is updated. If no interaction is found an interaction is created.

The Attribute `Id` is optional, and only relevant for marketing-driven processes. For more information, see [Overview \[page 662\]](#).

If additionally the optional `Attribute Id` is provided an additional search step is performed in case no interaction with fitting external key was found. The `Attribute Id` is used to retrieve an interaction by its internal key. If an interaction is found it is updated. If no interaction is found an error is raised and the corresponding message can be found in the *Import Monitor* app.

The following table provides you with an overview of mandatory attributes and the definition of the key of the external business document.

Entity	Property	Description	Mandatory	Key
Import Header	Id	Unique technical identifier of import run	x	
	Timestamp	Timestamp of the run		
	SourceSystemType	Type of the source system, such as ERP	x	x
	SourceSystemId	Identifier of the source system	x	x
BusinessDocument	Id	Key of the interaction that is updated.		

Entity	Property	Description	Mandatory	Key
	ContactIdOrigin	ID origin of the contact. If the ContactID is filled the ID Origin from the external system must be set, such as SAP_C4C_BUPA.	x	
	ContactId	Id of the contact in the external system.	x	
	InternalObjectType	Interaction type only mandatory for request mode	x	
	ExternalObjectType	Type of the external object, such as MARKETING_LEAD	x	x
	ExternalId	ID of the external object	x	x
	ExternalTimeStamp	Timestamp of the external object. Timestamp is used to process messages in the right sequence	x	
	<b>ActionCode</b>	<b>04</b>	x	

The main contact of the business document is defined by attribute pair *ContactIdOrigin* and *ContactId*. If no contact with this attribute pair (facet) exists in SAP Marketing Cloud a new contact is created. Contact data itself is replicated separately via, for example, OData service `CUAN_BUSINESS_PARTNER_IMP_SRV`.

## PredecessorId (Marketing-Driven Process)

The attribute *PredecessorId* is only relevant for marketing-driven processes.

If it is filled a predecessor lead interaction is determined and the ID of the campaign in which the lead predecessor interaction was created is copied to the current interaction to be created or updated.

The external key of the predecessor lead interaction is defined by attributes *SourceSystemType*, *SourceSystemId*, *ExternalObjectType*, and *ExternalId*. *SourceSystemType* and *SourceSystemId* are taken from entity *ImportHeader*, *ExternalId* is given by value of *PredecessorId* in entity *BusinessDocument* and *ExternalObjectType* is fixed to value "MARKETING\_LEAD".

If no predecessor lead interaction is found an error is raised which can be seen in the *Import Monitor* app.

## Request Mode: Delete (ActionCode 05)

If a business document is deleted in external system then action code "05" has to be used.

An interaction is determined according to the rules described under *Request Mode: Create and Update (ActionCode 04)*.

If an interaction is found the interaction is not removed from database. A so called obsolete flag is set for the interaction. The interaction can then be deleted in a subsequent step by standard deletion reports.

Besides *ExternalTimeStamp* and the obsolete flag no other interaction data is updated.

If no interaction is found an error is raised which is logged and can be seen in *Import Monitor* app.

### i Note

A of release 1902, the *obsolete* flag is **not** set, but the interaction is deleted physically from the database.

## Confirmation Mode: Set External Keys (ActionCode 02)

If the field action code has the value "02" the OData service is executed in confirmation message mode. The confirmation mode is only relevant for marketing-driven processes.

The main purpose of the confirmation message process step is to update existing interaction with the external business document key.

Only a small subset of fields contained in table below are taken into account in confirmation message mode.

Entity	Property	Description	Mandatory	External Key
ImporHeader	Id	Unique technical identifier of import run	x	
	TimeStamp	Timestamp of the run		
	SourceSystemType	Type of the source system, such as ERP	x	x
	SourceSystemId	Identifier of the source system	x	x
BusinessDocument	<b>Id</b>	<b>Internal key of the interaction that is to be updated</b>	x	
	ExternalObjectType	Type of the external object, such as MARKETING_LEAD	x	x
	ExternalId	ID of the external object	x	x
	ExternalTimeStamp	Timestamp of the external object. Timestamp is used to process messages in the right sequence	x	



Entity	Property	Description	Mandatory	External Key
	ActionCode	02	x	

The Property Id denoting the internal Id of the interaction is used to retrieve an existing interaction. If the interaction cannot be retrieved an error is raised and logged in the *Import Monitor* app. If the interaction can be retrieved then the mentioned fields above are updated in the interaction.

**Parent topic:** [Business Documents \[page 661\]](#)

## Related Information

[Overview \[page 662\]](#)

[Structure of OData Service CUAN\\_BUSINESS\\_DOCUMENT\\_IMP\\_SRV \[page 664\]](#)

[Basic Concepts \[page 672\]](#)

[Specifics for SAP Cloud for Customer Integration \[page 677\]](#)

[Payload Examples for Business Documents \[page 678\]](#)

[Import Monitor \[page 404\]](#)

## 5.2.10.5 Specifics for SAP Cloud for Customer Integration

Some specifics have to be considered for SAP Cloud for Customer Integration. SAP Cloud for Customer integration is defined by *SourceSystemType* c4c.

### Entity Type ProductItem

If the field *ObjectId* is initial the corresponding product item is neglected and no error is raised.

**Parent topic:** [Business Documents \[page 661\]](#)

## Related Information

[Overview \[page 662\]](#)

[Structure of OData Service CUAN\\_BUSINESS\\_DOCUMENT\\_IMP\\_SRV \[page 664\]](#)

[Basic Concepts \[page 672\]](#)

[Modes \[page 674\]](#)

[Payload Examples for Business Documents \[page 678\]](#)

## 5.2.10.6 Payload Examples for Business Documents

This section contains payload examples for CUAN\_BUSINESS\_DOCUMENT\_IMP\_SRV.

### Sales-Driven Process - Create or Change Business Document

#### Sample Code

```
{
  "Id": "MSG_20190820_I",
  "Timestamp": "2019-08-20T12:36:04.0000000",
  "SourceSystemType": "C4C",
  "SourceSystemId": "Z123",
  "BusinessDocuments": [
    {
      "Id": "00000000-0000-0000-0000-000000000000",
      "InternalObjectType": "MARKETING_LEAD",
      "ExternalObjectType": "MARKETING_LEAD",
      "ExternalId": "LEAD_20190820_I",
      "ExternalStatusCode": "1",
      "ContactIdOrigin": "SAP_C4C_BUPA",
      "ActionCode": "04",
      "ContactId": "IC_20190820_I",
      "ExternalTimeStamp": "2019-08-25T12:55:59.0000000"
    }
  ]
}
```

### Sales-Driven Process - Create or Change Business Document with Product Items

#### Sample Code

```
{
  "Id": "MSG_20190820_II",
  "Timestamp": "2019-08-20T12:36:04.0000000",
  "SourceSystemType": "C4C",
  "SourceSystemId": "Z123",
  "BusinessDocuments": [
    {
      "Id": "00000000-0000-0000-0000-000000000000",
      "InternalObjectType": "MARKETING_LEAD",
      "ExternalObjectType": "MARKETING_LEAD",
      "ExternalId": "LEAD_20190820_II",
      "ExternalStatusCode": "1",
      "ContactIdOrigin": "SAP_C4C_BUPA",
      "ActionCode": "04",
      "ContactId": "IC_20190820_II",
      "ExternalTimeStamp": "2019-08-25T12:55:59.0000000",
      "ProductItems": [
        {

```

```

        "Id" : "10",
        "ObjectId": "MAT_20180420_II ",
        "ObjectType": "SAP_C4C_PRODUCT",
        "ProductName": "Camera",
        "ProductDesc": "Camera",
        "Quantity": "5",
        "UnitOfMeasure": "ST"
    }
}
]
}

```

## Sales-Driven Process – Create or Change Business Document with Additional Interaction Contacts

### Sample Code

```

{
  "Id": "MSG_20190820_III ",
  "Timestamp": "2019-08-20T12:36:04.0000000",
  "SourceSystemType": "C4C",
  "SourceSystemId": "Z123",
  "BusinessDocuments": [
    {
      "Id": "00000000-0000-0000-0000-000000000000",
      "InternalObjectType": "MARKETING_LEAD",
      "ExternalObjectType": "MARKETING_LEAD",
      "ExternalId": "LEAD_20190820_III",
      "ExternalStatusCode": "1",
      "ContactIdOrigin": "SAP_C4C_BUPA",
      "ActionCode": "04",
      "ContactId": "IC_20190820_III",
      "ExternalTimeStamp": "2019-08-02T19:38:16.0000000",
      "TimeStamp": "2019-08-16T16:36:04.0000000",
      "AdditionalInteractionContacts": [
        {
          "InteractionAdditionalIntactnContactUUID":
"00000000-0000-0000-0000-000000000000",
          "InteractionContactOrigin": "SAP_C4C_BUPA",
          "InteractionContactId": " IC_20190820_IV "
        },
        {
          "InteractionAdditionalIntactnContactUUID":
"00000000-0000-0000-0000-000000000000",
          "InteractionContactOrigin": "SAP_C4C_BUPA",
          "InteractionContactId": " IC_20190820_V "
        }
      ]
    }
  ]
}

```

## Create Business Document with an Offer

### Sample Code

```
POST data:
{
  "Id": "MSG_20201221_II",
  "Timestamp": "2020-12-21T15:39:36.0000000",
  "SourceSystemType": "C4C",
  "SourceSystemId": "Z123",
  "BusinessDocuments": [
    {
      "Id": "00000000-0000-0000-0000-000000000000",
      "InternalObjectType": "SALES_ORDER",
      "ActionCode": "04",
      "ExternalObjectType": "SALES_ORDER",
      "ExternalId": "BUS-JMT ",
      "StatusCode": "06",
      "ContactIdOrigin": "SAP_C4C_BUPA",
      "ContactId": "JMT_ContactID",
      "ExternalTimeStamp": "2020-11-25T09:12:00.0000000",
      "MarketingAreaId": "MA01",
      "Currency": "EUR",
      "Offers": [
        {
          "ExternalOfferOrigin": "GENERIC",
          "ExternalOffer": "",
          "CouponCode": "CpnCode"
        }
      ]
    }
  ]
}
```

Parent topic: [Business Documents \[page 661\]](#)

## Related Information

[Overview \[page 662\]](#)

[Structure of OData Service CUAN\\_BUSINESS\\_DOCUMENT\\_IMP\\_SRV \[page 664\]](#)

[Basic Concepts \[page 672\]](#)

[Modes \[page 674\]](#)

[Specifics for SAP Cloud for Customer Integration \[page 677\]](#)

## 5.2.11 Agreements

Public OData API (`API_MKT_AGREEMENT_SRV`) for agreements. An agreement can be any kind of customer contract, for example, a sales contract or a contract that comprises specific services.



### Overview

The OData service `API_MKT_AGREEMENT_SRV` is used to replicate agreement data from different source systems into SAP Marketing Cloud (standard integration with other systems). One agreement instance can only have one origin system for its business data. However, to identify business documents from other systems, you can replicate additional external IDs with different origins of an agreement.

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://{host}:{port}/sap/opu/odata/sap/API_MKT_AGREEMENT_SRV</code>
<b>Service Metadata URI</b>	<code>https://{host}:{port}/sap/opu/odata/sap/API_MKT_AGREEMENT_SRV/\$metadata</code>
<b>Communication Scenario ID</b>	<code>SAP_COM_0175</code>
<b>Authorization</b>	The following business catalog is required: <ul style="list-style-type: none"><li>• <code>SAP_CEC_BC_MKT_API_AGR_PC</code></li></ul>
<b>Component for Incidents</b>	<ul style="list-style-type: none"><li>• CEC-MKT-DM-AGR (Agreement)</li></ul> <div data-bbox="821 1294 1394 1447"><p><b>i Note</b></p><p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p></div>
<b>Field Extensibility Supported</b>	Yes <div data-bbox="821 1525 1394 1749"><p><b>i Note</b></p><p>You must enable the Data Source under <i>UIs and Reports</i> for Data Source <code>API_MKT_AGREEMENT_SRV 0001</code> (API for Marketing Agreement) and <code>I_MKT_AGREEMENTTP</code> (Marketing: Agreement TP).</p></div> <p>For more information, see <a href="#">Agreements</a>.</p>

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_AGREEMENT_SRV/ \$metadata?sap- documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Agreements Details Page</a> 	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li><li>Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Agreements API</a> 	General access link takes you directly to the <i>Agreements</i> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## 5.2.11.1 Basic Concepts

This OData API allows updates of agreements from different source systems.

### OData Operations Supported

This OData API supports the following operations:

- The operation GET is supported for all entities.
- \$batch processing is only supported for updates. Within a batch request, only the following operations are supported:
  - PATCH (MERGE) for the entity types `Agreement` and `AgreementTerms`
  - PUT/DELETE for entity type `AgrmtAdditionalExtID`

Batch requests allow multiple operations to be grouped into a single HTTP request payload. Batch requests are submitted as a single `HTTP POST` request to the `$batch` endpoint of a service as described in [\[OData-URI ↗\]](#). The batch request must contain a content-type header specifying a content type of `multipart/mixed` as well as a boundary specification.

## Important Points to Note About Processing

- A `PATCH (MERGE)` request only updates the properties indicated in the request body and leaves everything which was not mentioned untouched. All properties that are not to be changed can be omitted. The transmitted properties are merged with the data already stored in SAP Marketing Cloud.
- If an agreement cannot be found based on its external key (`MKT_AgreementOrigin` and `MKT_AgreementExternalID`), the `PATCH` request creates a new agreement.
- **Create Agreements with Agreement Terms:** Although it is technically possible to create a root entry `/Agreement` without the corresponding entry for the `/AgreementTerms`, this would be semantically incorrect and these entries would not be usable in the segmentation and should therefore be avoided. A root entry without terms can only exist temporarily in the system due to a time lag between the creation of both entities. Segmentation only checks the timestamp of the Agreement Terms, not of the Agreement.
- Timestamps **must** always be loaded in the API service in UTC format, they are not automatically converted to UTC in the service. In Segmentation and Analytics, UTC timestamps are then converted to local time.
- Contacts are enhanced with the marketing area assignment that is used in the agreement.

## Checks

When an agreement is created or changed, the system performs checks for mandatory fields, field values and the existence of referenced entities. Failing checks results in an error and the incorrect data sets are displayed in the import monitor. For detailed information, refer to the details of each entity type in the section [Structure of OData Service API\\_MKT\\_AGREEMENT\\_SRV \[page 684\]](#).

## Agreement Bundles

You can implement the hierarchical relationships of agreements as bundles. An agreement may have a relation to a bundle (=related bundle). The bundle is an agreement itself. An agreement with such a relation is called a bundle member. For more information, see [Agreements](#)

There are no checks between the agreement bundle and the members of the agreement bundle. The source system needs to synchronize the data between the agreement bundle and its members if needed. For example, if a reference to a bundle product is replicated via the terms of the members of the agreement bundle, the source system should replicate any change of the product in the agreement bundle and update the terms of the corresponding members of the agreement bundle.

## Update Logic for Agreement Terms

When an agreement terms record is uploaded, it is compared to the relevant existing agreement terms records on the database. If the imported agreement terms record has the same start and end date as the existing agreement terms record, then the existing record is updated by the new record. Otherwise a new agreement terms record is created.

**Inactive Terms:** Any existing agreement terms record that overlaps with the new record is given the status **Inactive = I**. Inactive agreement terms are filtered out by default in the standard Analytics and Segmentation delivered, so that they do not clutter up the display. So, if you create custom Analytics or Segmentation, you should make sure to select Agreement Terms with the status **Active = A** only. Inactive agreement terms cannot be reactivated. Although they are not deleted in the database, they are excluded from usage.

### i Note

The PATCH request updates only the properties transmitted in the request body. All properties that are not to be changed can be omitted. The transmitted properties are merged with the data already stored.

The following examples illustrate the update logic when a new Agreement Term is uploaded for an Agreement for which there are existing Agreement Terms:

Examples of Update Logic

Validities of Existing Terms	New Term with New Validity	Update Logic
<ul style="list-style-type: none"><li>01.01.2016 to 31.12.2016</li><li>01.01.2017 to 31.12.2017</li><li>01.01.2018 to 31.12.2018</li></ul>	01.07.2016 to 31.07.2018	<ol style="list-style-type: none"><li>A new term is created.</li><li>All existing terms with overlapping validities are set to inactive.</li></ol> <div data-bbox="1018 757 1394 1070"><h3>i Note</h3><p>If you want the agreement term to remain valid in the Marketing system for the periods 01.01.2016 to 31.06.2016 and 01.08.2018 to 31.12.2018, then you have to upload new agreement terms for these two periods to the Marketing system.</p></div>
<ul style="list-style-type: none"><li>01.01.2016 to 31.12.2016</li><li>01.01.2017 to 31.12.2017</li></ul>	01.07.2016 to 31.07.2017	This new term invalidates both of the existing terms. The old terms are set to <b>inactive</b> . <div data-bbox="1018 1205 1394 1518"><h3>i Note</h3><p>If you want the agreement term to remain valid in the Marketing system for the periods 01.01.2016 to 31.06.2016 and 01.08.2017 to 31.12.2017, then you have to upload new agreement terms for these two periods to the Marketing system.</p></div>

## 5.2.11.2 Structure of OData Service API\_MKT\_AGREEMENT\_SRV

This document describes the structure of the Public OData API service `API_MKT_AGREEMENTS`.

Make sure you read the information in the topic [Basic Concepts \[page 682\]](#) before you start.



## Request Header

The request header contains the additional header fields listed in the table. Remember to include at least the mandatory request header fields in each payload.

Header Field	Description	Edm Core Type	Max Length	Mandatory
Sap-Cuan-SequenceId	Unique technical identifier of the imported data	Edm.String	30	X
Sap-Cuan-RequestTimestamp	OData service Timestamp of the data	Edm.DateTime	0	*
Sap-Cuan-SequenceNumber	Sequence number of the request. This number is normally incremented each time a new request for the same sequence ID is created.	Edm.Int16	0	*
Sap-Cuan-SourceSystemType	Type of the source system	Edm.String	20	X
Sap-Cuan-SourceSystemId	Identifier of the source system. This is a free text field.	Edm.String	255	X
Sap-Cuan-ExternalReferenceId	External reference of the inbound message	Edm.String	32	
Sap-Cuan-ExternalDocumentId	External identifier of the source document	Edm.String	20	

\* Either Sap-Cuan-RequestTimestamp or Sap-Cuan-SequenceNumber must be provided together with Sap-Cuan-SequenceId.

- The header fields Sap-Cuan-SequenceId and Sap-Cuan-RequestTimestamp or Sap-Cuan-SequenceNumber are used to check the sequence of the data received. Data with the **same** Sap-cuan-SequenceID **and** a timestamp older or sequence number lower than data already imported is ignored.
- The Sap-Cuan-SourceSystemType and Sap-Cuan-SourceSystemId fields allow you to distinguish between different source systems.
- The Sap-Cuan-ExternalReferenceId and Sap-Cuan-ExternalDocumentId allow better error analysis because they contain external references to a source SOAP message or an IDoc.

## Entity Sets and Entity Types in API MKT\_Agreement\_SRV

The API MKT\_AGREEMENT\_SRV OData service consists of the following entity sets and entity types:

Entity Set	Description	Path
Agreements	Agreement time independent attributes refer to agreements in SAP Marketing Cloud. Agreement data is collected and merged from several sources into the master data tables within SAP Marketing Cloud.	/Agreements
AgreementTerms	Agreement time-dependent attributes representing agreement terms (conditions) for a particular time slice of an agreement.	/AgreementTerms
AgrmtAdditionalExtIDs	<p>Additional external ID of the agreement from a different agreement origin.</p> <p>Agreement additional external IDs are optional. They are used in a scenario, when interactions are linked to agreements by using an alternative key (e.g. the agreement additional external ID is stored in the interaction). In this case the mapping of the alternative agreement key to the primary agreement key is done here.</p>	/AgrmtAdditionalExtIDs

## Agreement

You can perform the following operations on the Agreement entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/Agreement`
- **PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/$batch`

HTTP Method	Description	Path
<a href="#">GET</a>	<p>Get a list of agreements.</p> <p>This method supports standard OData parameters such as <code>\$filter</code>, <code>\$select</code>, <code>\$top</code>, <code>\$skip</code>, <code>\$count</code>, <code>\$inlinecount</code>, and <code>\$orderby</code></p>	/Agreements?\$top=1

HTTP Method	Description	Path
	Get a single agreement.	/ Agreements (MKT_AgreementOrigin= '<Agreement Origin>',MKT_AgreementExternalI D='<Agreement ID>')
<i>POST (Batch)</i>	Update or create an Agreement in batch mode. (Full Update) See payload example: <a href="#">Payload Examples for Agreements [page 693]</a>	https:// <Server>:<Port>/sap/opu/ odata/SAP/ API_MKT_AGREEMENT_SRV/\$batch

## Agreement

Property	Description	Edm. Core Type	Max Length	Mandatory	Key
MKT_AgreementOrigin	Agreement Origin.	Edm.String	30	x	x
MKT_AgreementExternalID	Agreement External ID.	Edm.String	80	x	x
MKT_AgrmtCancellationReason	The customer or provider reason for canceling the agreement. The reason also encodes which party triggered the cancellation.	Edm.String	10		
ContactID	External ID of Interaction Contact Data.	Edm.String	255	x*	
ContactOrigin	Origin of Interaction Contact Data.	Edm.String	20	x*	
MKT_AgreementIsBundle	The agreement represents an agreement bundle. An agreement bundle groups multiple agreements.  An agreement bundle may contain additional terms and itself have associated interactions.	Edm.Boolean	0		
MKT_AgreementIsBundleMember	Specifies whether this agreement is part of an agreement .	Edm.Boolean	0		
MKT_AgreementType	Used to categorize an agreement. Example agreement types: loan, car insurance, life insurance, mobile phone, electricity supply.	Edm.String	10	x*	

Property	Description	Edm. Core Type	Max Length	Mandatory	Key
MKT_MarketingArea	Marketing Area of the Agreement , may be used to restrict the access to agreements.	Edm.String	40		
MKT_AgreementStartDateTime	The initial start date of the first version of the agreement. An agreement renewal, for example, does not change this date.	Edm.DateTimeOffset	0	x*	
MKT_AgreementEndTime	Date at which the agreement ends. If the contract is open ended, then the date shall be 31.12.9999.	Edm.DateTimeOffset	0	x*	
MKT_AgreementIsCancelled	The agreement is canceled.	Edm.Boolean	0		
OriginDataLastChangeUTCDateTime	Last Change Timestamp of the source system.	Edm.DateTimeOffset			
MKT_AgreementBundleOrigin	Origin of the Agreement Bundle.	Edm.String	30		
MKT_AgreementBundleExternalID	External ID of the Agreement Bundle.	Edm.String	80		
MKT_AgrmtBundleStartDateTime	The start date and time of the agreement bundle.	Edm.DateTimeOffset	0		
MarketingLocationOrigin	Origin of Marketing Location	Edm.String	30		
MarketingLocationExternalID	ID of Marketing Location	Edm.String	50		

\*If you update an existing agreement term with the same key, you can omit the properties `ContactID`, `ContactOrigin`, `MKT_AgreementType`, `MKT_AgreementStartDateTime`, and `MKT_AgreementEndTime`. In this case, the missing properties will be taken from the existing record. If you create a new agreement you must provide all mandatory properties.

When a new agreement is created, the system checks whether this agreement already exists based on its external keys (`MKT_AgreementOrigin` and `MKT_AgreementExternalID`), for example, it checks whether the agreement is already referenced via an additional external ID in another agreement instance (`MKT_AgrmtAddlExternalOrigin` and `MKT_AgreementAddlExternalID`).

If you want to control access to agreement information, you must ensure that marketing area information is part of the agreements payload.

- **Field Value Checks in Agreement are performed for the following attributes:**
  - `MKT_AgreementOrigin`.
  - `MKT_AgreementBundleOrigin` for Customizing values.

- MKT\_AgrmtCancellationReason for Customizing values.
- MKT\_AgreementType for Customizing values.
- MKT\_MarketingArea for Customizing values.
- MKT\_AgreementStartDateTime before MKT\_AgreementEndDateTime.
- **The following dependencies are verified:**
  - The referenced marketing location has to exist.
  - The referenced interaction contact has to exist.

### Note

The contact is enhanced with the assignment to the marketing area that is used in the agreement.

## AgreementTerms

You can perform the following operations on the AgreementTerms entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/AgreementTerms`
- **PATCH in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/$batch`

HTTP Method	Description	Path
<i>GET</i>	Get a list of Agreement Terms.  This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code>	<code>/AgreementTerms?\$top=1</code>
	Get one single Agreement Term.	<code>/AgreementTerms(MKT_AgreementOrigin='&lt;AgreementOrigin&gt;',MKT_AgreementExternalID='&lt;AgreementID&gt;',MKT_AgrmtTermsStartDateTime=datetimetypeoffset'&lt;AgreementTerm Start DateTime&gt;',MKT_AgrmtTermsEndDateTime=datetimetypeoffset'&lt;AgreementTerm End DateTime&gt;')</code>
<i>POST (Batch)</i>	Update or create Agreement Terms in batch mode. (Full Update) See payload example: <a href="#">Payload Examples for Agreements [page 693]</a>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/\$batch</code>

## Agreement Terms

Property	Description	EDM Core Type	Max Length	Mandatory	Key
MKT_AgreementOrigin	Agreement Origin.	Edm.String	30	x	x
MKT_AgreementExternalID	Agreement External ID.	Edm.String	80	x	x
MKT_AgrmtTermsStartDateTime	The Start DateTime to End	Edm.String	0	x	x
MKT_AgrmtTermsEndDateTime	DateTime specify the period of which an agreement terms record is valid (agreement terms are time dependent).	Edm.String	0	x	x
MKT_AgreementStatus	The status of the agreement terms.	Edm.String	1	x*	
ProductOrigin	Product Origin.	Edm.String	30		
ProductID	External ID of the Product.	Edm.String	50		
MKT_AgrmtCanclnConditions	Code describing what conditions must be met to cancel the agreement.	Edm.String	10		
MKT_AgrmtCanclnDcsnToDateTime	The Customer must notify the provider about the cancellation before this date. Otherwise the contract will automatically renew when the AgreementtEndDateTime is reached.	Edm.DateTimeOffset	0		
MKT_AgreementRenewalType	Agreement renewal type.	Edm.String	0		
MKT_AgrmtProlngnDcsnFromDteTime	The customer may request a prolongation of their agreement between the dates AgreementProlongationDecisionFromDateTime and AgreementEndDateTime. This is relevant for agreements that do not automatically renew.	Edm.DateTimeOffset	0		
Mkt_AgrmtPaymentIsInAdvance	Indicates that the customer makes his payments in advance of the period of service.	Edm.Boolean	0		

Property	Description	EDM Core Type	Max Length	Mandatory	Key
MKT_AgreementPaymentFrequency	Code describing how often the customer makes payments to the provider. Examples: Yearly, Monthly.	Edm.String	4		
MKT_AgrmtBundleProductOrigin	Reference to the Product of an Agreement Bundle.	Edm.String	30		Mkt
MKT_AgrmtBundleProductID		Edm.String	50		
OriginDataLastChgUTCDateTime	Last Change Timestamp of the source system in UTC.	Edm.DateTimeOffset	0		

\* You can omit the property `MKT_AgreementStatus` if you update an existing agreement term with the same key. Then the status will be taken from the existing record. If you create a new agreement term you have to provide the agreement status.

When a new agreement term is created, the system checks whether this agreement already exists using its external keys (`MKT_AgreementOrigin` and `MKT_AgreementExternalID`). You need to create `/Agreement` (root entry) before you can create `/AgreementTerms`.

## Checks Performed

- **Field Value checks in `AgreementTerms`** are performed for the following attributes:
  - `MKT_AgrmtTermsStartDateTime` before `MKT_AgrmtTermsEndTime`
  - `MKT_AgreementStatus` for valid values:
    - "A" - Active
    - "S" - Suspended
    - "Q" - Quote
    - "P" - Application
  - `MKT_AgreementRenewalType` for valid values:
    - "" - Unknown
    - "1" - No renewal possible
    - "2" - Customer must request
    - "3" - Automatic
  - `MKT_AgrmtCanclnConditons` for Customizing values
  - `MKT_AgreementPaymentFrequency` for Customizing values
  - `MKT_AgrmtCorrespondenceMedium` for valid values:
    - "EMAIL" - Email
    - "FAX" - Fax
    - "PAPER" - Postal Mail
    - "SMS" - Text Message
    - "WEB" - Online Account
  - `MKT_AgreementPaymentMethod` for valid values:
    - "DD" - Direct Debit
    - "CC" - Credit Card

- "BC" - Bank Collection
- "BT" - Bank Transfer

The following dependencies are verified:

- The referenced product has to exist.
- The referenced bundle product has to exist.  
A bundle product can be referenced only if the agreement is a member of an agreement bundle.

## AgrmtAdditionalExtIDs

You can perform the following operations on the AgrmtAdditionalExtIDs entity set:

- **GET:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/AgrmtAdditionalExtIDs`
- **PUT, PATCH, or DELETE in batch:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/$batch`

HTTP Method	Description	Path
<i>GET</i>	Get a list of External Additional IDs.  This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> .	<code>/AgrmtAdditionalExtIDs?\$top=3</code>
<i>POST (Batch)</i>	Update or create an Agreement Additional External ID with PATCH in batch mode. (Full Update)	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/\$batch</code>
	Delete an Agreement Additional External ID with DELETE in batch mode. See payload example: <a href="#">Payload Examples for Agreements [page 693]</a>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/\$batch</code>
	Add one new Agreement Additional External ID via PUT in batch mode. See payload example: <a href="#">Payload Examples for Agreements [page 693]</a>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_AGREEMENT_SRV/\$batch</code>

Property	Description	Edm Core Type	Max Length	Mandatory	Key
MKT_AgreementOrigin	Agreement Origin.	Edm.String	30	x	x
MKT_AgreementExternalID	Agreement External ID.	Edm.String	80	x	x



Property	Description	Edm Core Type	Max Length	Mandatory	Key
MKT_AgrmtAddlExternalOrigin	Additional Agreement Origin.	Edm.String	30	x	x
MKT_AgreementAddlExternalID	Agreement External ID from an Additional Agreement Origin.	Edm.String	80	x	x

When a new additional external ID is created, the system checks based on its agreement external keys (MKT\_AgreementOrigin and MKT\_AgreementExternalID) whether this agreement already exists. You need to create /Agreement (root entry) before you can create /AgrmtAdditionalExtIDs.

When a new additional external ID is created, the system checks based on its additional external keys (MKT\_AgrmtAddlExternalOrigin and MKT\_AgreementAddlExternalID) whether this agreement instance exists (MKT\_AgreementOrigin and MKT\_AgreementExternalID) or is referenced already in any other agreement as additional external key (MKT\_AgrmtAddlExternalOrigin and MKT\_AgreementAddlExternalID).

### 5.2.11.3 Payload Examples for Agreements

Demonstrates creation and change of agreements.

The following examples show how you can use the agreements API. Insert your own data to fill the header and the entities.

#### Create Agreement with 1 Term

##### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
Agreements(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='20180410-165542-917') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementOrigin":"TEST_ORIGIN","MKT_AgreementIsCanceled":true,
"MKT_AgreementBundleOrigin":"","MKT_AgreementBundleExternalID":"","
"MKT_AgrmtBundleStartDateTime":null,
```

```

"MKT_AgreementExternalID":"20180410-165542-917",
"MKT_AgrmtCancellationReason":"","
>ContactID":"20180410-165542-277","ContactOrigin":"SAP_C4C_BUPA",
"MKT_AgreementIsBundle":false,
"MKT_AgreementIsBundleMember":true,
"MKT_AgreementType":"1",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/",
"MKT_AgreementStartDateTime":"/
Date(1325376000000+0000)/","MKT_AgreementEndDateTime":"/
Date(1640995200000+0000)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
AgreementTerms(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='2018
0410-165542-917',MKT_AgrmtTermsStartDateTime=datetimetypeoffset'2012-01-01T00%3A00
%3A00Z',MKT_AgrmtTermsEndDateTime=datetimetypeoffset'2013-12-31T00%3A00%3A00Z')
HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementStatus":"A",
"MKT_AgreementRenewalType":"1",
"ProductOrigin":"SAP_CRM_PRODUCT",
"ProductID":"SP_CS_CASH",
"MKT_AgrmtCanclnConditions":"1",
"MKT_AgrmtCanclnDcsnToDateTime":"2018-01-01T00:00:00Z",
"MKT_AgrmtProlngnDcsnFromDteTme":"2019-01-01T00:00:00Z",
"MKT_AgreementPaymentFrequency":"6",
"MKT_AgrmtPaymentIsInAdvance":true,
"MKT_AgrmtBundleProductOrigin":"TEST_ORIGIN",
"MKT_AgrmtBundleProductID":"TEST_BUNDLE",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621--
--batch--

```

## Create Agreement with 2 Terms

### Sample Code

```

--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
Agreements(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='20180410
-165542-917') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01

```

```

{"MKT_AgreementOrigin":"TEST_ORIGIN","MKT_AgreementIsCanceled":true,
"MKT_AgreementBundleOrigin":"","MKT_AgreementBundleExternalID":"","
"MKT_AgrmtBundleStartDateTime":null,
"MKT_AgreementExternalID":"20180410-165542-917",
"MKT_AgrmtCancellationReason":"","
"ContactID":"20180410-165542-277","ContactOrigin":"SAP_C4C_BUPA",
"MKT_AgreementIsBundle":false,
"MKT_AgreementIsBundleMember":true,
"MKT_AgreementType":"1",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/",
"MKT_AgreementStartDateTime":"/
Date(1325376000000+0000)/","MKT_AgreementEndDateTime":"/
Date(1640995200000+0000)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
AgreementTerms(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='2018
0410-165542-917',MKT_AgrmtTermsStartDateTime=datetimeoffset'2012-01-01T00%3A00
%3A00Z',MKT_AgrmtTermsEndDateTime=datetimeoffset'2013-12-31T00%3A00%3A00Z')
HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementStatus":"A",
"MKT_AgreementRenewalType":"1",
"ProductOrigin":"SAP_CRM_PRODUCT",
"ProductID":"CURRACCNTONLINE",
"MKT_AgrmtCanclnConditions":"1",
"MKT_AgrmtCanclnDcsnToDateTime":"2018-01-01T00:00:00Z",
"MKT_AgrmtProIngnDcsnFromDteTme":"2019-01-01T00:00:00Z",
"MKT_AgreementPaymentFrequency":"6",
"MKT_AgrmtPaymentIsInAdvance":true,
"MKT_AgrmtBundleProductOrigin":"TEST_ORIGIN",
"MKT_AgrmtBundleProductID":"TEST_BUNDLE",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
AgreementTerms(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='2018
0410-165542-917',MKT_AgrmtTermsStartDateTime=datetimeoffset'2014-01-01T00%3A00
%3A00Z',MKT_AgrmtTermsEndDateTime=datetimeoffset'2024-01-01T00%3A00%3A00Z')
HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementStatus":"S",
"MKT_AgreementRenewalType":"2",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621--
--batch-

```

## Change Agreement

### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_77162fcd-b8da-41ac-
a9f8-9357efbbd622
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd622
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
Agreements (MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='20180410-
165542-917') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgrmtCancellationReason":"0000001001",
"MKT_AgreementIsCanceled":false,
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)"}
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd622
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
AgreementTerms (MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='2018
0410-165542-917',MKT_AgrmtTermsStartDateTime=datetimetypeoffset'2012-01-01T00%3A00
%3A00Z',MKT_AgrmtTermsEndDateTime=datetimetypeoffset'2014-12-31T00%3A00%3A00Z')
HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementStatus":"S",
"MKT_AgreementRenewalType":"2",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)"}
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd622--
--batch--
```

## Create an Additional External Agreement ID

### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_77162fcd-b8da-41ac-
a9f8-9357efbbd621
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PUT
AgrmtAdditionalExtIDs (MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalI
```

```
D='20180410-165542-917',MKT_AgrmtAddlExternalOrigin='EXTERN_ORIGIN',MKT_Agreem
entAddlExternalID='AGR_ADDL_EXT_ID') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementOrigin":"EXTERN_ORIGIN",
"MKT_AgreementExternalID":"20180410-165542-917",
"MKT_AgrmtAddlExternalOrigin":"EXTERN_ORIGIN",
"MKT_AgreementAddlExternalID":"AGR_ADDL_EXT_ID"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621--
--batch--
```

## Delete an Additional External Agreement ID

### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_77162fcd-b8da-41ac-
a9f8-9357efbbd621
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
DELETE
AgrmtAdditionalExtIDs(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='
20180410-165542-917',MKT_AgrmtAddlExternalOrigin='EXTERN_ORIGIN',MKT_AgreementAdd
lExternalID='AGR_ADDL_EXT_ID') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20170508141617.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementOrigin":"EXTERN_ORIGIN",
"MKT_AgreementExternalID":"20180410-165542-917",
"MKT_AgrmtAddlExternalOrigin":"EXTERN_ORIGIN",
"MKT_AgreementAddlExternalID":"AGR_ADDL_EXT_ID"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621--
--batch--
```

## Create Agreement Bundle with 2 Bundle Members

```
--batch
Content-Type: multipart/mixed; boundary=changeset_77162fcd-b8da-41ac-
a9f8-9357efbbd621
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
```

```

PATCH
Agreements (MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='BUNDLE_EXAM
PLE') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20201217112001.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementOrigin":"TEST_ORIGIN",
"MKT_AgreementExternalID":"BUNDLE_EXAMPLE",
"MKT_AgreementIsCanceled":false,
"MKT_AgreementBundleOrigin":"","
"MKT_AgreementBundleExternalID":"","
"MKT_AgrmtBundleStartDateTime":null,
"ContactID":"20180410-165542-277",
"ContactOrigin":"SAP_C4C_BUPA",
"MKT_AgreementIsBundle":true,
"MKT_AgreementIsBundleMember":false,
"MKT_AgreementType":"1",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/",
"MKT_AgreementStartDateTime":"/Date(1590969600000+0000)/",
"MKT_AgreementEndDateTime":"/Date(1609372800000+0000)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
AgreementTerms (MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='BUNDLE_
EXAMPLE',MKT_AgrmtTermsStartDateTime=datetimeoffset'2020-01-01T12%3A00%3A00Z',MKT
_AgrmtTermsEndDateTime=datetimeoffset'2025-12-31T12%3A00%3A00Z') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20201217114000.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementStatus":"A",
"MKT_AgreementRenewalType":"1",
"ProductOrigin":"TEST_ORIGIN",
"ProductID":"TEST_BUNDLE",
"MKT_AgrmtCanclnConditions":"1",
"MKT_AgrmtCanclnDcsnToDateTime":"2020-06-30T12:00:00Z",
"MKT_AgrmtProlngnDcsnFromDteTme":"2020-06-01T00:00:00Z",
"MKT_AgreementPaymentFrequency":"6",
"MKT_AgrmtPaymentIsInAdvance":true,
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
Agreements (MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='BUNDLE_MEMB
ER01') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20201217112001.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementOrigin":"TEST_ORIGIN",
"MKT_AgreementExternalID":"BUNDLE_MEMBER01",
"MKT_AgreementIsCanceled":false,
"MKT_AgreementBundleOrigin":"TEST_ORIGIN",
"MKT_AgreementBundleExternalID":"BUNDLE_EXAMPLE",

```

```

"MKT_AgrmtBundleStartDateTime":"/Date(1590969600000+0000)/",
"ContactID":"20180410-165542-277",
"ContactOrigin":"SAP_C4C_BUPA",
"MKT_AgreementIsBundle":false,
"MKT_AgreementIsBundleMember":true,
"MKT_AgreementType":"1",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/",
"MKT_AgreementStartDateTime":"/Date(1590969600000+0000)/",
"MKT_AgreementEndDateTime":"/Date(1609372800000+0000)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
AgreementTerms(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='BUNDLE_
MEMBER01',MKT_AgrmtTermsStartDateTime=datetimeoffset'2020-06-01T12%3A00%3A00Z',MK
T_AgrmtTermsEndDateTime=datetimeoffset'2021-12-31T12%3A00%3A00Z') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20201217114001.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementStatus":"A",
"MKT_AgreementRenewalType":"1",
"ProductOrigin":"SAP_CRM_PRODUCT",
"ProductID":"CURRACCNTONLINE",
"MKT_AgrmtCanclnConditions":"1",
"MKT_AgrmtCanclnDcsnToDateTime":"2020-06-30T12:00:00Z",
"MKT_AgrmtProlnngnDcsnFromDteTme":"2020-06-01T00:00:00Z",
"MKT_AgreementPaymentFrequency":"6",
"MKT_AgrmtPaymentIsInAdvance":true,
"MKT_AgrmtBundleProductOrigin":"TEST_ORIGIN",
"MKT_AgrmtBundleProductID":"TEST_BUNDLE",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
Agreements(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='BUNDLE_MEMB
ER02') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20201217112001.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementOrigin":"TEST_ORIGIN",
"MKT_AgreementExternalID":"BUNDLE_MEMBER02",
"MKT_AgreementIsCanceled":false,
"MKT_AgreementBundleOrigin":"TEST_ORIGIN",
"MKT_AgreementBundleExternalID":"BUNDLE_EXAMPLE",
"MKT_AgrmtBundleStartDateTime":"/Date(1590969600000+0000)/",
"MKT_AgrmtCancellationReason":"","",
"ContactID":"20180410-165542-277",
"ContactOrigin":"SAP_C4C_BUPA",
"MKT_AgreementIsBundle":false,
"MKT_AgreementIsBundleMember":true,
"MKT_AgreementType":"1",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/",
"MKT_AgreementStartDateTime":"/Date(1590969600000+0000)/",
"MKT_AgreementEndDateTime":"/Date(1609372800000+0000)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621
Content-Type: application/http

```

```

Content-Transfer-Encoding: binary
PATCH
AgreementTerms(MKT_AgreementOrigin='TEST_ORIGIN',MKT_AgreementExternalID='BUNDLE_MEMBER02',MKT_AgrmtTermsStartDateTime=datetimeoffset'2020-12-01T12%3A00%3A00Z',MKT_AgrmtTermsEndDateTime=datetimeoffset'2021-12-31T12%3A00%3A00Z') HTTP/1.1
Content-Type: application/json
Content-Length: 1024
Sap-Cuan-SourceSystemId: XXXCLN123
Sap-Cuan-SourceSystemType:
Sap-Cuan-SequenceId: AGREEMENT_MASTER_DATA
Sap-Cuan-RequestTimestamp: 20201217114002.0000001
Sap-Cuan-ExternalReferenceId: XXXCLN12320170508141617_01
{"MKT_AgreementStatus":"A",
"MKT_AgreementRenewalType":"1",
"ProductOrigin":"SAP_CRM_PRODUCT",
"ProductID":"SP_CS_CASH",
"MKT_AgrmtCanclnConditions":"1",
"MKT_AgrmtCanclnDcsnToDateTime":"2020-06-30T12:00:00Z",
"MKT_AgrmtProlngnDcsnFromDteTme":"2020-06-01T00:00:00Z",
"MKT_AgreementPaymentFrequency":"6",
"MKT_AgrmtPaymentIsInAdvance":true,
"MKT_AgrmtBundleProductOrigin":"TEST_ORIGIN",
"MKT_AgrmtBundleProductID":"TEST_BUNDLE",
"OriginDataLastChgUTCDateTime":"/Date(1523372142917)/"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd621--
--batch--

```

## 5.2.12 Scores

Public OData API (API\_MKT\_SCORE\_SRV) for Scores

### Entity Data Model

**Service Metadata URI:** [https://<Server>:<Port>/sap/opu/odata/SAP/API\\_MKT\\_SCORE\\_SRV;/ \\$metadata](https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_SCORE_SRV;/ $metadata)

### Technical Data

OData Version	2.0
Root URI	<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_SCORE_SRV">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_SCORE_SRV</a>
Service Metadata URI:	<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_SCORE_SRV/\$metadata">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_SCORE_SRV/\$metadata</a>
Authorizations	The following business catalog role is required: SAP_CEC_BC_MKT_API_IC2_PC



Communication Scenario ID	SAP_COM_0307
Component for Incidents	CEC-MKT-ML-PRE
Field Extensibility	No

### i Note

Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

### i Note

You can convert the XML file to an XML table to make it easier to read.

Access Link	Remarks
<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_SCORE_SRV;v=0003/\$metadata?sap-documentation=all">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_SCORE_SRV;v=0003/\$metadata?sap-documentation=all</a>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Scores Details Page</a>	<p>General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>1. On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>2. Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Scores API</a>	General access to the <i>Scores</i> metadata file. One-time registration or logon is required.

## Related Information

<https://api.sap.com>

## 5.2.12.1 Basic Concepts

The public API for Scores `API_MKT_SCORE_SRV` supports operations on the `Scores Business Object`.

### i Note

Please note that score saving must be enabled if you want to use this API. For more information, see [Saving Scores for a Predictive Model](#).

## Score Versioning

The score values for a contact are stored in a version together with the timestamp of their calculation.

Every time scores or predictive models are calculated in the background, the calculated score values are saved under a new version. This version corresponds to the current UTC timestamp. A nightly report deletes all outdated versions. However, at least one version is kept, even if the version is outdated.

When you use scores or predictive models in *Segmentation* or *Customer Profile*, the latest version of the score is used.

- If no version exists for a score, the score values are calculated on the fly. If a large number of score values for contacts are calculated on the fly, this can lead to performance issues in the system.

In the *Scores* API, `MarketingScoreDateTime` corresponds to the timestamp which is used as version.


- If several score values are uploaded at once, all score values need to have the same timestamp.
- If different uploads use the same version, meaning the same timestamp, the upload will add score values for new contacts and it will overwrite score values of contacts which were already uploaded with this version.
- If a contact is included in an older version, but not in the latest version, their score value will be **0** or **No Valuation**, depending on the application in which the score value is displayed in.
- The timestamp must not be in the future. The timestamp must not be older than an already existing timestamp of this score.

## Processing Info

Batch requests are submitted as a single `HTTP POST` request to the `$batch` endpoint of a service as described in [\[OData-URI ↗\]](#). The batch request must contain a header parameter `content-type`, specifying the value **multipart/mixed** and **boundary=batch**. The operation header must include the `Sap-Cuan-SequenceId`.

A `PATCH (MERGE)` request updates only the properties indicated in the request body and leaves everything untouched that was not mentioned. All properties that are not to be changed can be omitted. The transmitted properties are merged with the data already stored in SAP Marketing Cloud. If the record to be updated is not found, then it is simply created.

## Best Practices

- The *ContactOrigin* cannot be shareable. If the origin is set to *Shareable*, this will trigger an error. For more information, see [Configuring Origins](#). You can view sample payloads and test the API at <https://api.sap.com> .
- The *ContactOrigin* cannot be updated. It's a key field together with the *Interaction Contact ID*.

## 5.2.12.2 Structure of OData Service API\_MKT\_SCORE\_SRV

This document describes the structure of the Public OData API service API\_MKT\_SCORE\_SRV. Make sure you read the Basic Concepts topic before you start.

### Entities

The Scores OData API provides the following entities:

Entity	Description	Path
Scores	This entity contains a persisted score scenario. A persisted score scenario is a scenario for which scores are saved.	/Scores (MarketingScore=<score id>)
ScoreModels	This entity contains the score model associated with a particular score scenario.	/ScoreModels (MarketingScore=<score id>, MarketingScoreModel=<model id>)
ScoreValues	This entity contains the score value for a score scenario, score model, and interaction contact.	/ScoreValues (MarketingScore=<score id>, MarketingScoreModel=<model id>, MarketingScoreDateTime=<date>, MarketingScoredObjectUUID=<contact uuid> )
ScoreTargetObjects	This entity contains the target object for a score scenario.	/ScoreTargetObjects (MarketingScore=<score id>, MarketingScoreModel=<model id>, MtkgScoreTargetObjectType=<target object>)

You can view sample payloads and test the API at <https://api.sap.com> .

## Entity Sets

The Scores OData API provides the following entity sets:

Entity Set	Description	Path
Scores	This entity contains the persisted type of score scenarios. A persisted score scenario is a scenario for which scores are saved.	/Scores
ScoreModels	This entity contains the score models associated with a particular score scenario.	/ScoreModels
ScoreValues	This entity contains the score value for a score scenario, score model, and interaction contact.	/ScoreValues

You can view sample payloads and test the API at <https://api.sap.com>.

### Scores

You can perform the following operations on the Scores entity set:

HTTP Method	Path	Comments
GET	/Scores	The options \$inlinecount and \$expand are supported.
	/Scores(<Marketing Score>)	

#### i Note

Only persisted scores can be summoned using the Scores API.

## ScoreModels

You can perform the following operations on the `ScoreModels` entity set:

HTTP Method	Path	Comments
<i>GET</i>	<code>/ScoreModels</code> <hr/> <code>/ScoreModels(MarketingScore={MarketingScore},MarketingScoreModel={MarketingScoreModel})</code>	The options <code>\$top</code> and <code>\$filter</code> on <code>MarketingScore</code> and <code>MarketingScoreModel</code> are supported.
<i>POST</i>	<code>/ScoreModels</code>	

## ScoreValues

You can perform the following operations on the `ScoreValues` entity set:

HTTP Method	Path	Comments
<i>GET</i>	<code>/ScoreValues</code>	<ul style="list-style-type: none"><li>The option <code>\$top</code> is mandatory with maximal 5000.</li><li>The option <code>\$inlinecount</code> is supported.</li><li>The option <code>\$filter</code> is valid only for the fields <code>MarketingScore</code>, <code>MarketingScoreModel</code>, <code>MarketingScoreValue</code>, <code>MarketingScoreDateTime</code> and <code>MarketingScoredObjectUUID</code> and <code>ID/Origin</code> combined. At least <code>MarketingScore</code> or <code>MarketingScoreModel</code> should be referenced in the filter option. If the <code>MarketingScoreModel</code> is not specified in the filter, the result will cover all the models related to this score.</li><li>The filtering on <code>MarketingScoreDateTime</code> gives only the latest version.</li><li>If the filter is used with <code>MarketingScoredObjectUUID</code> or <code>MarketingScoredID/</code></li></ul>

HTTP Method	Path	Comments
	/ ScoreValues (MarketingScore= '<Marketing Score>', MarketingScoreModel= '<Marketing Score Model>', MarketingScoredObjectUUID=guid '<UUID>', MarketingScoreDateTime=datetimeoffset '<Timestamp>')	Origin, the result is the latest score value assigned to this UUID or ID/Origin. • Only the operation <b>EQ</b> or <b>=</b> is supported when filtering on MarketingScore, MarketingScoreModel, MarketingScoreDateTime and MarketingScoredObjectUUID or MarketingScoredID/Origin.
<b>POST</b>	/ Scores (MarketingScore='<Marketing Score>')/ ScoreModels	You can only post score values from externally created scores.  For best performance, deep insert is supported as illustrated in the payload examples.

## ScoreTargetObjects

You can perform the following operations on the `ScoreTargetObjects` entity set:

HTTP Method	Path	Comments
<b>GET</b>	/ScoreTargetObjects	
	/ ScoreTargetObjects (MarketingScore='{MarketingScore}', MarketingScoreModel='{MarketingScoreModel}', MtkgScoreTargetObjectType='{MtkgScoreTargetObjectType}')	

### Note

Score values can only be imported in models which have an assigned implementation method dedicated for external score values. This is always the case for score models created via this OData.

## 5.2.12.3 Payload Examples for Scores

The following examples show how you can use the Scores API.

## Scores

### GET all scores

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/Scores
```

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/Scores?$inlinecount=allpages&
$expand=ScoreModels
```

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/Scores?$select=MarketingScore
```

### GET one score

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/Scores('CHURN_SCORE')
```

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/Scores('CHURN_SCORE')?$expand=ScoreModels
```

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/Scores('CHURN_SCORE')?
$select=MarketingScore,MarketingScoreName,MarketingScorePurpose
```

## Score Models

### GET all Score Models

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreModels?$top=10&
$filter=(MarketingScoreModel eq '555') and (MarketingScore eq 'CHURN_SCORE')
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreModels?$top=10&
$filter=(MarketingScoreModel eq '555')
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreModels?$top=10&
$filter=(MarketingScore eq 'CHURN_SCORE')
```

## GET one Score Model

### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/  
ScoreModels (MarketingScore='CHURN_SCORE,MarketingScoreModel='555')
```

### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/  
ScoreModels (MarketingScore='CHURN_SCORE,MarketingScoreModel='555')?  
$select=MarketingScoreModelUUID
```

## POST Score Model

### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreModels
```

### Sample Code

```
{  
  "MarketingScore": "ChurnScore",  
  "MarketingScoreModelName": "<model name>"  
}
```

## Score Values

### GET all Score Values

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreValues?$top=20&  
$filter=MarketingScore eq 'CHURN_SCORE
```

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreValues?$top=20&  
$filter=MarketingScore eq 'CHURN_SCORE and MarketingScoredObjectUUID eq  
guid'00163e34-bda6-1ed7-bf8e-ff79868c52ea'
```

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreValues?$top=3000&  
$filter=MarketingScore eq 'ChurnScore' and MarketingScoreDateTime gt  
datetimeoffset '2019-07-12T07%3A58%3A00.4030000Z'
```



### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreValues?$top=3000&
$filter=MarketingScore eq 'ChurnScore' and MarketingScoreDateTime eq
datetimeoffset'2019-07-12T07%3A58%3A00.4030000Z'
```

### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreValues?$top=3000&
$filter=MarketingScore eq 'ChurnScore' and MarketingScoreValue gt 0.33
```

### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreValues?$top=3000&
$filter=MarketingScore eq 'CHURN_SCORE' and MarketingScoreValue le 0.33
```

### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreValues?$skip=2&$top=3000&
$filter=MarketingScore eq 'CHURN_SCORE' and MarketingScoreValue le 0.33&
$orderby=MarketingScoreValue desc
```

## GET one Score Value

### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/
ScoreValues(MarketingScore='CHURN_SCORE',MarketingScoreModel='555',MarketingSc
oreDateTime=datetimeoffset'2019-07-12T07%3A58%3A00.4030000Z',MarketingScoreOb
jectUUID=guid'ffffffff-ffff-ffff-ffff-fffffffffffffff')
```

### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/
ScoreValues(MarketingScore='CHURN_SCORE',MarketingScoreModel='555',MarketingSc
oreDateTime=datetimeoffset'2019-07-12T07%3A58%3A00.4030000Z',MarketingScoreOb
jectUUID=guid'ffffffff-ffff-ffff-ffff-fffffffffffffff')?
$select=MarketingScoreValue
```

## POST Score Values

### Sample Code

```
sap/opu/odata/sap/API_MKT_SCORE_SRV/Scores(MarketingScore=<score id >)/
ScoreModels
```

### Sample Code

```
{ "MarketingScore": "ChurnScore", ; "MarketingScoreModel": "555", "ScoreValues":
[ { "MarketingScoreDateTime": "2017-12-02T14:26:00.9824060",
```

```
"MarketingScoredObjectUUID":"ffffffff-ffff-ffff-ffff-fffffffffffffff",  
"MarketingScoreValue":"0.60" } ] }
```

## Score Target Objects

### GET all Target Objects

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/ScoreTargetObjects?  
$filter=MarketingScore eq 'CHURN_SCORE'
```

### GET one Target Object

#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/  
ScoreTargetObjects(MarketingScore='CHURN_SCORE',MarketingScoreModel='555',Mtkg  
ScoreTargetObjectType='IP_TARGETPRODUCT')
```

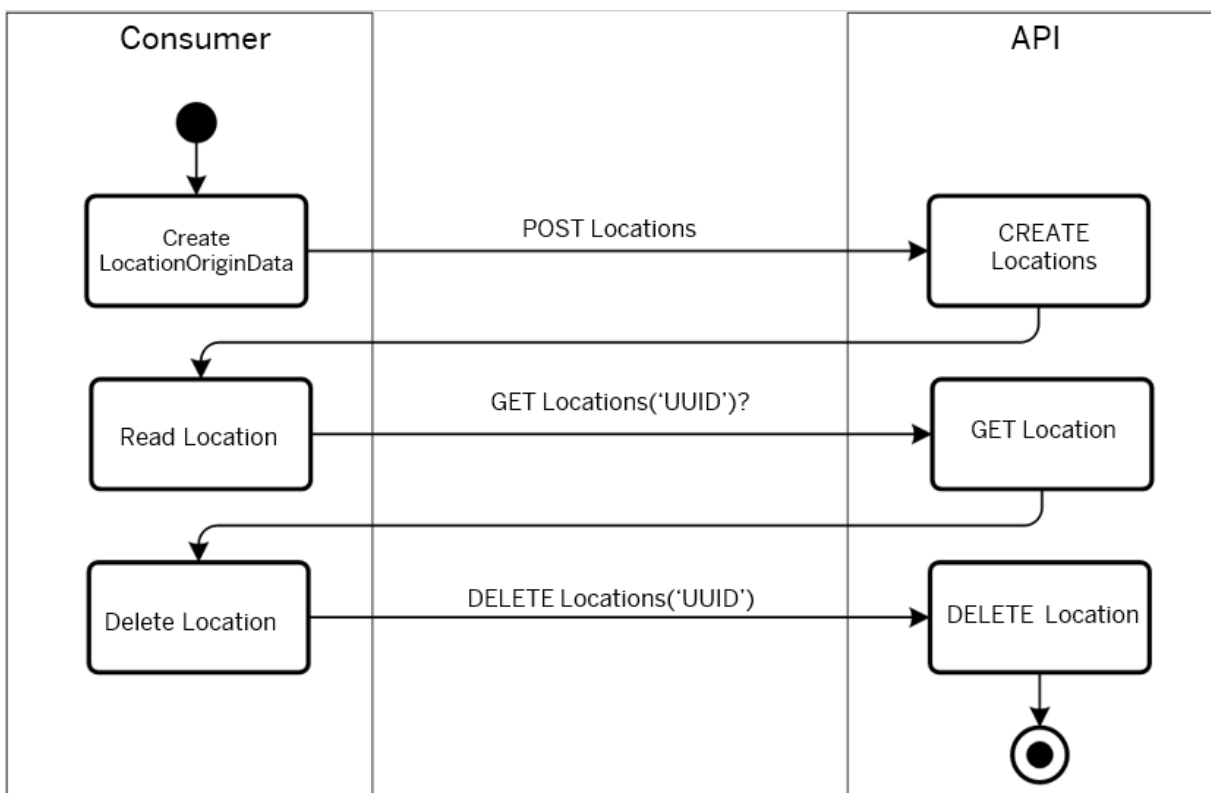
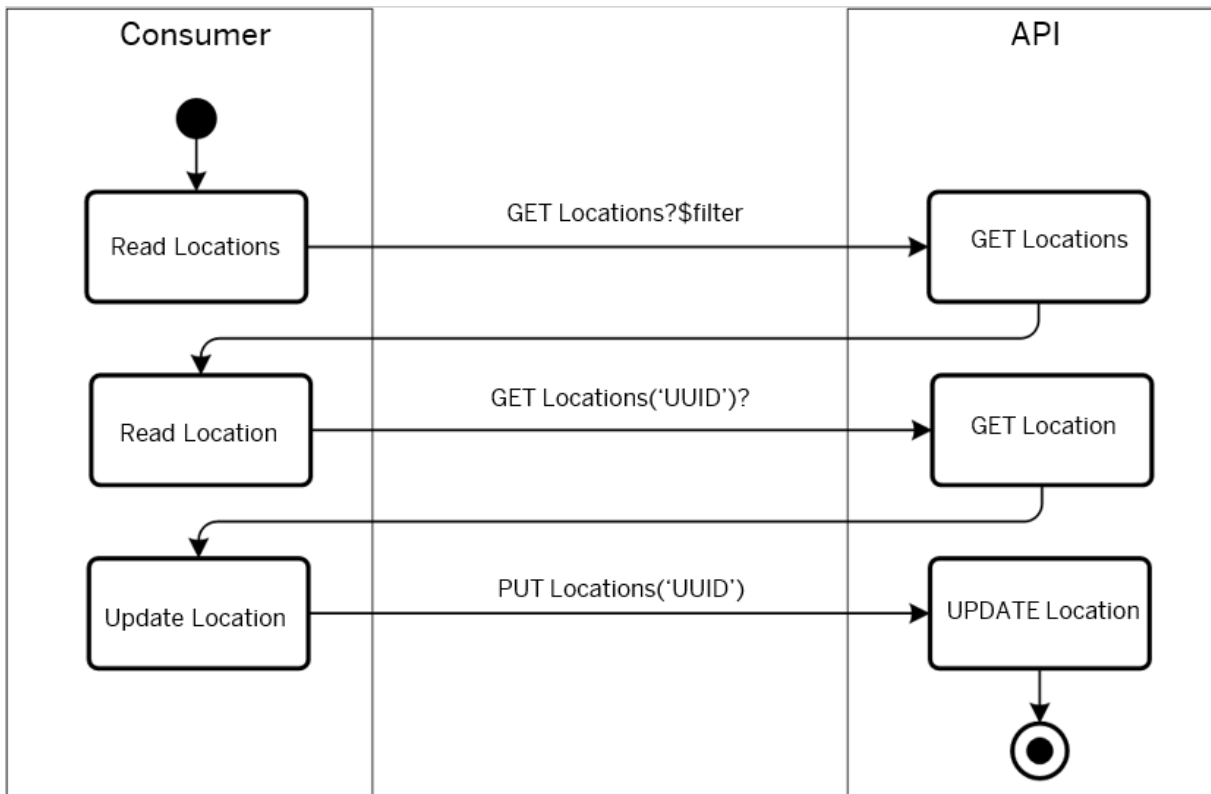
#### Sample Code

```
/sap/opu/odata/sap/API_MKT_SCORE_SRV/  
ScoreTargetObjects(MarketingScore='CHURN_SCORE',MarketingScoreModel='555',Mtkg  
ScoreTargetObjectType='IP_TARGETPRODUCT')?$select=MarketingScore
```

## 5.2.13 Marketing Locations

Public OData API (API\_MKT\_LOCATION) for Marketing Locations. A marketing location is any physical or virtual location where a marketing activity can be conducted.

The following diagrams illustrate the business process model for the marketing location API:



## Technical Data

Name of the Service	API_MKT_LOCATION
Underlying BO	BO_MARKETING_LOCATION
Package	CUAN_BO_MARKETING_LOCATION
OData Version	2.0
Root URI	https://<Server>:<Port>/sap/opu/odata/SAP/ API_MKT_LOCATION_SRV
Service Metadata URI	https://<Server>:<Port>/sap/opu/odata/SAP/ API_MKT_LOCATION_SRV/\$metadata
Authorizations	The following business catalog role is required: <ul style="list-style-type: none"> <li>SAP_CEC_BC_MKT_API_LOC_PC</li> </ul>
Communication Scenario ID	SAP_COM_0305
Component for Incidents	<ul style="list-style-type: none"> <li>CEC-MKT-DM-LOC (Marketing Location)</li> </ul>
<div style="background-color: #f0f0f0; padding: 5px;"> <p><b>i Note</b></p> <p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p> </div>	
Field Extensibility Supported	Yes

## Technical Field Documentation

You can access technical documentation for the API fields by downloading a metadata file in one of the following ways:

### i Note

You can convert the XML file to an XML table to make it easier to read.

Access Link	Remarks
https://<Server>:<Port>/sap/opu/odata/SAP/ API_MKT_LOCATION_SRV;v=0002/\$metadata?sap- documentation=all	Only for internal access. You need to provide the server and port names.

Access Link	Remarks
<a href="#">Marketing - Marketing Locations Details Page</a>	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Marketing Locations API</a>	General access to the <i>Marketing Locations</i> metadata file. One-time registration or login is required.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## 5.2.13.1 Basic Concepts

The Marketing Location API supports CRUD operations on the `Marketing Location Business Object`.

### Best Practices

You can view sample payloads and test the API at <https://api.sap.com>.

### Field Extensibility

In addition to the pre-delivered attributes, you can add customer-specific fields using the *Custom Fields* app. For more information, see [Custom Fields](#).

## 5.2.13.2 Structure of API\_MKT\_LOCATION

This document describes the structure of the Public OData API service `API_MKT_LOCATION`. Make sure you read the Basic Concepts topic before you start.

### Entity Sets

The Marketing Location OData API provides the following entity sets:

Entity	Description	Path
Location	This entity contains the list of marketing locations.	<code>/Locations</code>
LocationInfo	This entity contains the list of marketing locations information.	<code>/LocationsInfo</code>
LocationOriginData	This entity contains the list of origin data for marketing locations.	<code>/LocationsOriginData</code>
LocationOriginDataInfo	This entity contains the list of origin data information for marketing locations.	<code>/LocationsOriginDataInfo</code>

### Location

**Resource Path:** `/Location`

You can perform the following operations on the `Location` entity:

Operations on the `Location` entity

HTTP Method	Description	Path
<code>GET</code>	Get the list of marketing locations.	<code>GET /Locations</code>
	Get the details of a marketing location.	<code>GET /Locations('Marketing Location UUID')</code>

### LocationsInfo

**Resource Path:** `/LocationsInfo`

You can perform the following operations on the `LocationsInfo` entity:

Operations on the `LocationInfo` entity

HTTP Method	Description	Path
<i>GET</i>	Get the list of marketing locations information.	GET <code>/LocationsInfo</code>
	Get the details of a marketing location information.	GET <code>/LocationsInfo('Marketing Location UUID')</code>

## LocationsOriginData

**Resource Path:** `/LocationOriginData`

You can perform the following operations on the `LocationOriginData` entity:

Operations on the `LocationOriginData` entity

HTTP Method	Description	Path
<i>GET</i>	Get the list of marketing locations origin data.	GET <code>/LocationsOriginData</code>
	Get the details of a marketing location origin.	GET <code>/LocationsOriginData('Marketing Location ID','Marketing Location Origin')</code>
<i>POST</i>	Create a marketing location origin.	POST <code>/LocationsOriginData</code>
<i>PATCH</i>	Update a marketing location origin.	PATCH <code>/LocationsOriginData</code>
<i>PUT</i>	Update a marketing location origin.	PUT <code>/LocationsOriginData</code>
<i>DELETE</i>	Delete the marketing location.	DELETE <code>/LocationsOriginData('Marketing Location ID','Marketing Location Origin')</code>

## LocationsOriginDataInfo

**Resource Path:** `/LocationOriginDataInfo`

You can perform the following operations on the `LocationOriginDataInfo` entity:

Operations on the `LocationOriginDataInfo` entity

HTTP Method	Description	Path
<i>GET</i>	Get the details of a marketing location origin information.	GET / LocationsOriginDataInfo('Marketing Location ID','Marketing Location Origin')
<i>POST</i>	Create a marketing location origin information.	POST / LocationsOriginDataInfo
<i>PATCH</i>	Update a marketing location origin information.	PATCH / LocationsOriginDataInfo
<i>PUT</i>	Update a marketing location origin information	PUT / LocationsOriginDataInfo

## 5.2.13.3 Payload Examples

The following examples show how you can use the Marketing Locations API.

### Create Marketing Locations

#### POST

##### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
POST LocationsOriginData HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 10000
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingLocationID>RDLOC1309</d:MarketingLocationID>
<d:MarketingLocationOrigin>WECHAT_POI</d:MarketingLocationOrigin>
<d:MarketingLocationName>Location Name for LOC1409</d:MarketingLocationName>
<d:CompanyName>CompanyName for RDLOC1409</d:CompanyName>
<d:MarketingLocationMallName>MallName for LOC1309</
d:MarketingLocationMallName>
<d:MarketingArea>GERMANY</d:MarketingArea>
<d:MarketingAreaName>GERMANY</d:MarketingAreaName>
```



```

<d:MarketingLocationType>MALL</d:MarketingLocationType>
<d:MarketingLocationTypeName>MALL</d:MarketingLocationTypeName>
<d:Country>CA</d:Country>
<d:CountryName>Canada</d:CountryName>
<d:CityName>Montreal</d:CityName>
<d:AddressRegion>QC</d:AddressRegion>
<d:RegionName>QC</d:RegionName>
<d:PostalCode>H9X2R2</d:PostalCode>
<d:AddressStreetName>ROBERT BOURASSA</d:AddressStreetName>
<d:AddressHouseNumber>111</d:AddressHouseNumber>
<d:Building>Building1309</d:Building>
<d:Floor>FOURTH</d:Floor>
<d:RoomNumber>LAGRANGE</d:RoomNumber>
<d:PhoneNumber>5149999999</d:PhoneNumber>
<d:FaxNumber>5149999999</d:FaxNumber>
<d:EmailAddress>LOC1509@example.com</d:EmailAddress>
<d:WebsiteURL>example.com</d:WebsiteURL>
<d:ImageURL>example.com</d:ImageURL>
<d:Longitude>1010</d:Longitude>
<d:Latitude>1020</d:Latitude>
<d:SpatialReferenceSystem>Ref070903</d:SpatialReferenceSystem>
<d:ValidityStartDate>2016-09-13T08:01</d:ValidityStartDate>
<d:ValidityEndDate>2016-09-13T08:01</d:ValidityEndDate>
</m:properties>
</atom:content>
</atom:entry>
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
POST LocationsOriginData HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 10000
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingLocationID>RDLOC1509</d:MarketingLocationID>
<d:MarketingLocationOrigin>WECHAT_POI</d:MarketingLocationOrigin>
<d:MarketingLocationName>Location_Name for LOC1509</d:MarketingLocationName>
<d:CompanyName>CompanyName for RDLOC1309</d:CompanyName>
<d:MarketingLocationMallName>MallName for LOC1309</
d:MarketingLocationMallName>
<d:MarketingArea>GERMANY</d:MarketingArea>
<d:MarketingAreaName>GERMANY</d:MarketingAreaName>
<d:MarketingLocationType>MALL</d:MarketingLocationType>
<d:MarketingLocationTypeName>MALL</d:MarketingLocationTypeName>
<d:Country>CA</d:Country>
<d:CountryName>Canada</d:CountryName>
<d:CityName>Montreal</d:CityName>
<d:AddressRegion>QC</d:AddressRegion>
<d:RegionName>QC</d:RegionName>
<d:PostalCode>H9X2R2</d:PostalCode>
<d:AddressStreetName>DUKE</d:AddressStreetName>
<d:AddressHouseNumber>111</d:AddressHouseNumber>
<d:Building>Building1309</d:Building>
<d:Floor>FOURTH</d:Floor>
<d:RoomNumber>LAGRANGE</d:RoomNumber>
<d:PhoneNumber>5149999999</d:PhoneNumber>
<d:FaxNumber>5149999999</d:FaxNumber>
<d:EmailAddress>LOC1509@example.com</d:EmailAddress>
<d:WebsiteURL>example.com</d:WebsiteURL>
<d:ImageURL>example.com</d:ImageURL>
<d:Longitude>1010</d:Longitude>
<d:Latitude>1020</d:Latitude>
<d:SpatialReferenceSystem>Ref070903</d:SpatialReferenceSystem>
<d:ValidityStartDate>2016-09-13T08:01</d:ValidityStartDate>

```

```

<d:ValidityEndDate>2016-09-13T08:01</d:ValidityEndDate>
</m:properties>
</atom:content>
</atom:entry>
--changeset--
--batch--

```

## Create a Marketing Location

### Sample Code

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingLocationID>RDLOC13095</d:MarketingLocationID>
<d:MarketingLocationOrigin>WECHAT_POI</d:MarketingLocationOrigin>
<d:MarketingLocationName>Location Name for LOC1409</d:MarketingLocationName>
<d:CompanyName>CompanyName for RDLOC1409</d:CompanyName>
<d:MarketingLocationMallName>MallName for LOC1309</
d:MarketingLocationMallName>
<d:MarketingArea>GERMANY</d:MarketingArea>
<d:MarketingAreaName>GERMANY</d:MarketingAreaName>
<d:MarketingLocationType>MALL</d:MarketingLocationType>
<d:MarketingLocationTypeName>MALL</d:MarketingLocationTypeName>
<d:Country>CA</d:Country>
<d:CountryName>Canada</d:CountryName>
<d:CityName>Montreal</d:CityName>
<d:AddressRegion>QC</d:AddressRegion>
<d:RegionName>QC</d:RegionName>
<d:PostalCode>H9X2R2</d:PostalCode>
<d:AddressStreetName>ROBERT BOURASSA</d:AddressStreetName>
<d:AddressHouseNumber>111</d:AddressHouseNumber>
<d:Building>Building1309</d:Building>
<d:Floor>FOURTH</d:Floor>
<d:RoomNumber>LAGRANGE</d:RoomNumber>
<d:PhoneNumber>5149999999</d:PhoneNumber>
<d:FaxNumber>5149999999</d:FaxNumber>
<d:EmailAddress>LOC1509@example.com</d:EmailAddress>
<d:WebsiteURL>example.com</d:WebsiteURL>
<d:ImageURL>example.com</d:ImageURL>
<d:Longitude>1010</d:Longitude>
<d:Latitude>1020</d:Latitude>
<d:SpatialReferenceSystem>Ref070903</d:SpatialReferenceSystem>
<d:ValidityStartDate>2016-09-13T08:01</d:ValidityStartDate>
<d:ValidityEndDate>2016-09-13T08:01</d:ValidityEndDate>
</m:properties>
</atom:content>
</atom:entry>

```

## Delete a Marketing Location

### DELETE

#### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
DELETE
LocationsOriginData (MarketingLocationID='RDLOC1309',MarketingLocationOrigin='W
ECHAT_POI') HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 10000
--changeset--
--batch--
```

## Get a List of all Marketing Locations

### GET

#### Sample Code

```
--batch
Content-Type: application/http
Content-Transfer-Encoding: binary
GET LocationsOriginData HTTP/1.1
--batch--
```

## Get a List of One Marketing Location

### GET

#### Sample Code

```
--batch
Content-Type: application/http
Content-Transfer-Encoding: binary
GET
LocationsOriginData (MarketingLocationID='RDLOC1309',MarketingLocationOrigin='W
ECHAT_POI') HTTP/1.1
--batch
Content-Type: application/http
Content-Transfer-Encoding: binary
GET
LocationsOriginData (MarketingLocationID='RDLOC1509',MarketingLocationOrigin='W
ECHAT_POI') HTTP/1.1
--batch--
```

## Update Several Marketing Locations

### PATCH

#### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
LocationsOriginData (MarketingLocationID='RDLOC1309',MarketingLocationOrigin='WECHAT_POI') HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 10000
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingLocationID>RDLOC1309</d:MarketingLocationID>
<d:MarketingLocationOrigin>WECHAT_POI</d:MarketingLocationOrigin>
<d:MarketingLocationName>Location Name for LOC1409 Updated</d:MarketingLocationName>
<d:CompanyName>CompanyName for RDLOC1409 Updated</d:CompanyName>
<d:MarketingLocationMallName>MallName for LOC1309 Updated</d:MarketingLocationMallName>
<d:MarketingArea>GERMANY</d:MarketingArea>
<d:MarketingAreaName>GERMANY</d:MarketingAreaName>
<d:MarketingLocationType>MALL</d:MarketingLocationType>
<d:MarketingLocationTypeName>MALL</d:MarketingLocationTypeName>
<d:Country>CA</d:Country>
<d:CountryName>Canada</d:CountryName>
<d:CityName>Montreal</d:CityName>
<d:AddressRegion>QC</d:AddressRegion>
<d:RegionName>QC</d:RegionName>
<d:PostalCode>H9X2R2</d:PostalCode>
<d:AddressStreetName>ROBERT BOURASSA</d:AddressStreetName>
<d:AddressHouseNumber>111</d:AddressHouseNumber>
<d:Building>Building1309</d:Building>
<d:Floor>FOURTH</d:Floor>
<d:RoomNumber>LAGRANGE</d:RoomNumber>
<d:PhoneNumber>5149999999</d:PhoneNumber>
<d:FaxNumber>5149999999</d:FaxNumber>
<d:EmailAddress>LOC1509@example.com</d:EmailAddress>
<d:WebsiteURL>example.com</d:WebsiteURL>
<d:ImageURL>example.com</d:ImageURL>
<d:Longitude>1010</d:Longitude>
<d:Latitude>1020</d:Latitude>
<d:SpatialReferenceSystem>Ref070903</d:SpatialReferenceSystem>
<d:ValidityStartDate>2016-09-13T08:01</d:ValidityStartDate>
<d:ValidityEndDate>2016-09-13T08:01</d:ValidityEndDate>
</m:properties>
</atom:content>
</atom:entry>
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
LocationsOriginData (MarketingLocationID='RDLOC1509',MarketingLocationOrigin='WECHAT_POI') HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 10000
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
```

```

<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
--batch
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
LocationsOriginData (MarketingLocationID='RDLOC1309',MarketingLocationOrigin='WECHAT_POI') HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 10000
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingLocationID>RDLOC1309</d:MarketingLocationID>
<d:MarketingLocationOrigin>WECHAT_POI</d:MarketingLocationOrigin>
<d:MarketingLocationName>Location Name for LOC1409 Updated</d:MarketingLocationName>
<d:MarketingLocationName>
<d:CompanyName>Company Name for RDLOC1409 Updated</d:CompanyName>
<d:MarketingLocationMallName>Mall Name for LOC1309 Updated</d:MarketingLocationMallName>
<d:MarketingArea>GERMANY</d:MarketingArea>
<d:MarketingAreaName>GERMANY</d:MarketingAreaName>
<d:MarketingLocationType>MALL</d:MarketingLocationType>
<d:MarketingLocationTypeName>MALL</d:MarketingLocationTypeName>
<d:Country>CA</d:Country>
<d:CountryName>Canada</d:CountryName>
<d:CityName>Montreal</d:CityName>
<d:AddressRegion>QC</d:AddressRegion>
<d:RegionName>QC</d:RegionName>
<d:PostalCode>H9X2R2</d:PostalCode>
<d:AddressStreetName>ROBERT BOURASSA</d:AddressStreetName>
<d:AddressHouseNumber>111</d:AddressHouseNumber>
<d:Building>Building1309</d:Building>
<d:Floor>FOURTH</d:Floor>
<d:RoomNumber>LAGRANGE</d:RoomNumber>
<d:PhoneNumber>5149999999</d:PhoneNumber>
<d:FaxNumber>5149999999</d:FaxNumber>
<d:EmailAddress>LOC1509@example.com</d:EmailAddress>
<d:WebsiteURL>example.com</d:WebsiteURL>
<d:ImageURL>example.com</d:ImageURL>
<d:Longitude>1010</d:Longitude>
<d:Latitude>1020</d:Latitude>
<d:SpatialReferenceSystem>Ref070903</d:SpatialReferenceSystem>
<d:ValidityStartDate>2016-09-13T08:01</d:ValidityStartDate>
<d:ValidityEndDate>2016-09-13T08:01</d:ValidityEndDate>
</m:properties>
</atom:content>
</atom:entry>
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
LocationsOriginData (MarketingLocationID='RDLOC1509',MarketingLocationOrigin='WECHAT_POI') HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 10000
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingLocationID>RDLOC1509</d:MarketingLocationID>

```

```

<d:MarketingLocationOrigin>WECHAT_POI</d:MarketingLocationOrigin>
<d:MarketingLocationName>Location Name for LOC1509 Updated</
d:MarketingLocationName>
<d:CompanyName>CompanyName for RDLOC1309 Updated3</d:CompanyName>
<d:MarketingLocationMallName>MallName for LOC1309 Updated</
d:MarketingLocationMallName>
<d:MarketingArea>GERMANY</d:MarketingArea>
<d:MarketingAreaName>GERMANY</d:MarketingAreaName>
<d:MarketingLocationType>MALL</d:MarketingLocationType>
<d:MarketingLocationTypeName>MALL</d:MarketingLocationTypeName>
<d:Country>CA</d:Country>
<d:CountryName>Canada</d:CountryName>
<d:CityName>Montreal</d:CityName>
<d:AddressRegion>QC</d:AddressRegion>
<d:RegionName>QC</d:RegionName>
<d:PostalCode>H9X2R2</d:PostalCode>
<d:AddressStreetName>DUKE</d:AddressStreetName>
<d:AddressHouseNumber>111</d:AddressHouseNumber>
<d:Building>Building1309</d:Building>
<d:Floor>FOURTH</d:Floor>
<d:RoomNumber>LAGRANGE</d:RoomNumber>
<d:PhoneNumber>5149999999</d:PhoneNumber>
<d:FaxNumber>5149999999</d:FaxNumber>
<d:EmailAddress>LOC1509@example.com</d:EmailAddress>
<d:WebsiteURL>example.com</d:WebsiteURL>
<d:ImageURL>example.com</d:ImageURL>
<d:Longitude>1010</d:Longitude>
<d:Latitude>1020</d:Latitude>
<d:SpatialReferenceSystem>Ref070903</d:SpatialReferenceSystem>
<d:ValidityStartDate>2016-09-13T08:01</d:ValidityStartDate>
<d:ValidityEndDate>2016-09-13T08:01</d:ValidityEndDate>
</m:properties>
</atom:content>
</atom:entry>
--changeset--
--batch--

```

## 5.2.14 Classifications (Deprecated)

Public OData API (`API_MKT_ML_CLASSIFICATION`, deprecated) for reading and writing data about classifications. A classification is the truth about whether a certain event in the past or not. You define this event yourself.

### Entity Data Model

**Service Metadata URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV;v=0002/$metadata`

### Technical Data

OData Version	2.0
Root URI	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV;v=0002
Service Metadata URI:	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV;v=0002/\$metadata
Authorizations	The following business catalog role is required: SAP_CEC_BC_MKT_API_IC2_PC
Communication Scenario ID	SAP_COM_0245
Component for Incidents	CEC-MKT-ML-CLS
Field Extensibility	No

### i Note



Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

### i Note


You can convert the XML file to an XML table to make it easier to read.

Access Link	Remarks
https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV;v=0003/\$metadata?sap-documentation=all	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Classifications Page</a> 	General access to the <a href="#">Details</a> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"> <li>On the <a href="#">Details</a> page, click <a href="#">Download Specification</a> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Classifications API</a> 	General access to the <a href="#">Contacts</a> metadata file. One-time registration or logon is required.

## 5.2.14.1 Basic Concepts (Deprecated)


The public API for Classifications `API_MKT_ML_CLASSIFICATION_SRV` supports operations on the `Classification Business Object`.

### Processing Info

Batch requests are submitted as a single `HTTP POST` request to the `$batch` endpoint of a service as described in [OData-URI ]. The batch request must contain a header parameter `content-type`, specifying the value `multipart/mixed` and `boundary=batch`. The operation header must include the `Sap-Cuan-SequenceId`.

A `PATCH (MERGE)` request updates only the properties indicated in the request body and leaves everything untouched that was not mentioned. All properties that are not to be changed can be omitted. The transmitted properties are merged with the data already stored in SAP Marketing Cloud. If the record to be updated is not found, then it is simply created.

### Best Practices

- For classifications, batch requests are only supported for `GET` and `PATCH` operations. We recommend to use a batch request to update classification values for one classification only and we recommend to restrict a batch request to one predictive scenario only. The batch request is limited to 100.000 classification values.  
For `GET`, the batch request is limited to 5000 classifications.  
For deep create, the request is limited to 50.000 classification values.
- The `ContactOrigin` cannot be shareable. If the origin is set to `Shareable`, this will trigger an error. For more information, see [Configuring Origins](#). You can view sample payloads and test the API at <https://api.sap.com> .
- The `ContactOrigin` cannot be updated. It's a key field together with the `Interaction Contact ID`.

### Error Messages

If the OData service is not accessible, for example due to missing authorization, or because the system is not available, a corresponding HTTP status code is returned.

If the OData service is accepted by the gateway component in the SAP Marketing Cloud system, the HTTP status code 201 or 204 is returned. Potential processing errors are recorded in the SAP Marketing Cloud system in the [Import Monitor](#) app, where they can be monitored, restarted and discarded.

Data processing for classifications is mostly synchronous. In most cases an OK response, such as a receipt notification, is returned almost immediately. An exception to this would be data uploads that might contain severe errors, such as parse or format errors, and so would not return an OK response but an error message.



### i Note

Data processing for *Deep Create* and *Bacth* requests is asynchronous. The data you upload lands in a staging area, where it is then further processed. In the response you will get a *Reference Message ID*.

To view the processing status and to check for errors or success messages, you must launch the *Import Monitor* app and search for the *Reference Message ID*. Messages for classifications in this app are displayed under the API for classifications. In the event of errors, you can restart or discard the import in the *Import Monitor*.

## 5.2.14.2 Structure of OData Service API\_MKT\_ML\_CLASSIFICATION\_SVR (Deprecated)

This document describes the structure of the Public OData API service API\_MKT\_ML\_CLASSIFICATION. Make sure you read the Basic Concepts topic before you start.

### Entity Sets

The Classification OData API provides the following entity sets:

Entity Set	Description	Path
ClassificationsType	This entity contains information about the version, target object and predictive scenario of a classification.	/ClassificationsType
ClassificationsValuesType	This entity contains information about a contact and their classification value.	/ClassificationsValuesType
MchnLrngScenarios	This entity contains information about the custom predictive scenarios of the type <i>External Score</i> .	/MchnLrngScenarios

### ClassificationsType

The table below describes the properties for the entity `ClassificationsType`.

## ClassificationsType Property Names and Descriptions

Property Name	Property Description	Usage
MchnLrngClassificationVersion	Contains the version of the classification.  <b>i Note</b> A classification can only have one version. Different classifications can have different versions.	Mandatory
MchnLrngClassificationVersText	Contains the freetext name, which you can give to the classification version.	
MchnLrngTargetObject	Contains the target object for which the classification is calculated.	Mandatory
PredictiveScenario	Contains the custom predictive scenario that is used for storing the classification.  <b>i Note</b> The predictive scenario must exist in the <i>Predictive Scenarios</i> app.	Mandatory

You can perform the following operations on the `ClassificationsType` entity set:

HTTP Method	Description	Path
<i>GET</i>	Read header information for a classification, which includes classification key, predictive scenario, classification version and optionally classification version text.	<code>/Classifications('&lt;PredictiveScenario&gt;,&lt;MchnLrngTargetObject&gt;,&lt;MchnLrngClassificationVersion&gt;')</code>
	Read all classifications	<code>/Classifications</code>
<i>POST</i>	Create a new classification with header information about the classification key, predictive scenario, target object and classification version.  The classification version text is optional.	<code>/Classifications</code>
	Update an existing classification with additional information, namely the classification version text.  The <i>PUT</i> request replaces the existing entry, so all property values in the entry either take the values indicated in the request body, or are reset to their default value if not mentioned in the request.	<code>/Classifications('&lt;PredictiveScenario&gt;,&lt;MchnLrngTargetObject&gt;,&lt;MchnLrngClassificationVersion&gt;')</code>

HTTP Method	Description	Path
<i>PATCH</i>	<p>Update an existing classification with additional information, namely the classification version text.</p> <p>The <i>PATCH</i> request updates only the properties indicated in the request body, and leaves untouched anything not mentioned in its current state.</p>	<p>/</p> <p>Classifications('&lt;PredictiveScenario&gt;,&lt;MchnLrngTargetObject&gt;,&lt;MchnLrngClassificationVersion&gt;')</p>
<i>PATCH</i> (Batch)	<p>In your batch request, you can update both different classifications and different classification values.</p> <p>For batch requests, you can only update classification values for one classification. The batch request is limited to 100,000 classification values.</p> <p>Add to the batch request header: <b>Content-Type: multipart/mixed; boundary=batch;</b></p> <p>Add to the operation header: <b>Sap-Cuan-SequenceId: CLASSIFICATION_BATCH_SINGLE</b></p>	<p>https://</p> <p>&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CONTACT_SRV;v=0002/\$batch</p>
<i>DELETE</i>	<p>Delete an existing classification and all its classification values.</p>	<p>/</p> <p>Classifications('&lt;PredictiveScenario&gt;,&lt;MchnLrngTargetObject&gt;,&lt;MchnLrngClassificationVersion&gt;')</p>

## ClassificationsValueType

The table below describes the properties for the entity `ClassificationsValueType`.

ClassificationsValueType Property Names and Descriptions

Property Name	Property Description	Usage
InteractionContactId	<p>Contains the ID of the interaction contact.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>The ID for the interaction contact must exist in the system.</p> </div>	
InteractionContactOrigin	<p>Contains the origin of the contact.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <p>The origin cannot be shareable.</p> </div>	

Property Name	Property Description	Usage
InteractionContactUUID	Contains the internal ID of an interaction contact in SAP Marketing Cloud. The ID must exist in the system.	Mandatory
MchnLrngClassificationValue	Contains the classification value for the contact. The classification value can only be <b>0</b> or <b>1</b> .	
MchnLrngClassificationVersion	Contains the version of the classification.  <div style="border-left: 2px solid #0070C0; padding-left: 10px; background-color: #F0F0F0;"> <p><b>i Note</b></p> <p>A classification can only have one version.</p> <p>Different classifications can have different versions.</p> </div>	Mandatory
MchnLrngClfnEndTime	Contains the end date of the analysis period.  <div style="border-left: 2px solid #0070C0; padding-left: 10px; background-color: #F0F0F0;"> <p><b>i Note</b></p> <p>The end date time of a classification value must be later than the start date time of the same classification value.</p> </div>	
MchnLrngClfnStartTime	Contains the start data of the analysis period.  <div style="border-left: 2px solid #0070C0; padding-left: 10px; background-color: #F0F0F0;"> <p><b>i Note</b></p> <p>The start date time of a classification value must be earlier than the end date time of the same classification value.</p> </div>	Mandatory
MchnLrngTargetObject	Contains the target object for which the classification is calculated.	Mandatory
PredictiveScenario	Contains the custom predictive scenario that is used for storing the classifications.  The predictive scenario must exist in the <a href="#">Predictive Scenarios</a> app.	Mandatory

You can perform the following operations on the `ClassificationValuesType` entity set:

HTTP Method	Description	Path
GET	Read information for a specific classification value, which includes the interaction contact UUID, predictive scenario, target object, start date time and the optional attributes contact ID, ID origin, end date time and a classification value.	/ClassificationValues ('<PredictiveScenario>,<MchnLrngTargetObject>,<MchnLrngClassificationVersion>,<InteractionContactUUID=guid>,<MchnLrngClfnStartDateTime>')
	Get a list of classification values. You can read the classification values for one classification or for several different classifications. This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, and \$orderby	/ClassificationValues
	<p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• A maximum of 5,000 machine learning scenarios can be fetched in a single request.</li> <li>• Specification of TOP is mandatory.</li> <li>• Batch requests are also supported for GET.</li> </ul>	
POST	<p>Create a single classification value, give its five keys, interaction contact UUID, predictive scenario, start date time, Season Ticket, and classification version.</p> <p>Additional attributes are contact ID and ID origin, end date time and a classification value.</p> <p>Instead of the contact UUID, you can also give the contact ID and ID origin.</p>	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_ML_CLASSIFICATION_S RV/ClassificationValues</pre>
POST (Deep Create)	<p>Upload multiple classification values belonging to only one classification.</p> <p><b>i Note</b></p> <p>If the classification for which you want to upload values does not exist yet, it is also created with the deep create. A deep create is only possible for the initial load.</p> <p>A maximum of 50,000 classification values can be uploaded in one request.</p>	<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_ML_CLASSIFICATION_S RV/Classifications</pre>

HTTP Method	Description	Path
<i>PUT</i>	<p>Update an existing classification value with additional information about end date time and a classification value.</p> <p>The <i>PUT</i> request replaces the existing entry, so all property values in the entry either take the values indicated in the request body, or are reset to their default value if not mentioned in the request.</p>	<p>/</p> <pre>ClassificationValues ('&lt;PredictiveScenario&gt;, &lt;MchnLrngTargetObject&gt;, &lt;MchnLrngClassificationVersion&gt;, &lt;InteractionContactUUID=guid&gt;, &lt;MchnLrngClfnStartDateTime&gt;')</pre>
<i>PATCH</i>	<p>Update an existing classification value with additional information about end date time and a classification value.</p> <p>The <i>PATCH</i> request updates only the properties indicated in the request body, and leaves untouched anything not mentioned in its current state.</p>	<p>/</p> <pre>ClassificationValues ('&lt;PredictiveScenario&gt;, &lt;MchnLrngTargetObject&gt;, &lt;MchnLrngClassificationVersion&gt;, &lt;InteractionContactUUID=guid&gt;, &lt;MchnLrngClfnStartDateTime&gt;')</pre>
<i>PATCH</i> (Batch)	<p>In your batch request, you can update both different classifications and different classification values.</p> <p>We recommend to use a batch request to update classification values for one classification only and we recommend to restrict a batch request to one predictive scenario only. The batch request is limited to 100,000 classification values.</p> <p>Add to the batch request header: <b>Content-Type: multipart/mixed; boundary=batch;</b></p> <p>Add to the operation header: <b>Sap-Cuan-SequenceId: CLASSIFICATION_BATCH_SINGLE</b></p>	<p>https://</p> <pre>&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV/\$batch</pre>
<i>DELETE</i>	<p>Delete an existing classification value.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>Note</b></p> <p>The corresponding classification entity is not deleted, only the specified classification value.</p> </div>	<p>/</p> <pre>ClassificationValues ('&lt;PredictiveScenario&gt;, &lt;MchnLrngTargetObject&gt;, &lt;MchnLrngClassificationVersion&gt;, &lt;InteractionContactUUID=guid&gt;, &lt;MchnLrngClfnStartDateTime&gt;')</pre>

## MchnLrngScenarios Property Names and Descriptions

The table below describes the properties for the entity `MchnLrngScenarios`.

Property Name	Property Description	Usage
MachineLearningScenario	Contains the ID of the machine learning scenario.	

Property Name	Property Description	Usage
MachineLearningScenarioText	Contains the name of the machine learning scenario as it was defined in <a href="#">Predictive Studio</a> .	

You can perform the following operations on the `MchnLrngScenarios` entity set:

HTTP Method	Description	Path
<a href="#">GET</a>	Get a list of machine learning scenarios. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> .	<code>/MchnLrngScenarios</code>

### 5.2.14.3 Payload Examples (Deprecated)

The following examples show how you can use the Classifications API.

## Classifications

### GET All Classifications

#### Sample Code

```
/sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV/Classifications
```

### GET one Classification

#### Sample Code

```
/Classifications(PredictiveScenario='CHURN_RENEWAL_SEASON_TICKET',MchnLrngTargetObject='Season%20Ticket',MchnLrngClassificationVersion=1)
```

### POST one Classification

#### Sample Code

```
{
  "PredictiveScenario": "CHURN_RENEWAL_SEASON_TICKET",
  "MchnLrngTargetObject": "Season Ticket",
  "MchnLrngClassificationVersion": 1,
  "MchnLrngClassificationVersText": "Version 1.0"
}
```

## PUT **one Classification**

### Sample Code

```
{
  "MchnLrngClassificationVersText": "Version 2.0"
}
```

## PATCH **one Classification**

### Sample Code

```
{
  "MchnLrngClassificationVersText": "Version 3.0"
}
```

## PATCH **Several Classifications in a Batch Request**

### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
Classifications(PredictiveScenario='CHURN_RENEWAL_SEASON_TICKET',MchnLrngTargetObject='Cloud',MchnLrngClassificationVersion=1) HTTP/1.1
Content-Type: application/json
Content-Length: ###
Sap-Cuan-SequenceId: CLASSIFICATION_BATCH_SINGLE
{
  "MchnLrngClassificationVersText" : "Version 4.0"
}
--changeset--
--batch--
```

## DELETE **one Classification**

### Sample Code

```
/
Classifications(PredictiveScenario='CHURN_RENEWAL_SEASON_TICKET',MchnLrngTargetObject='Season%20Ticket',MchnLrngClassificationVersion=1)
```

## Classification Values

### GET **All Classification Values**

### Sample Code

```
sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV/ClassificationValues?$top=50
```



Or:

### Sample Code

```
--batch
Content-Type: application/http
Content-Transfer-Encoding: binary
GET ClassificationValues?$top=50 HTTP/1.1
--batch--
```

## GET **one Classification Value**

### Sample Code

```
/sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV/
ClassificationValues(PredictiveScenario='CHURN_RENEWAL_SEASON_TICKET',MchnLrng
TargetObject='Season
%20Ticket',MchnLrngClassificationVersion=1,InteractionContactUUID=guid'0000c9e
9-49b6-1ed3-b3dd-
ffffffffffff',MchnLrngClfnStartDateTime=datetimeoffset'2017-04-01T00%3A00%3A00
Z')
```

Or:

```
--batch
Content-Type: application/http
Content-Transfer-Encoding: binary
GET
ClassificationValues(PredictiveScenario='CHURN_RENEWAL_SEASON_TICKET',MchnLrng
TargetObject='Season
%20Ticket',MchnLrngClassificationVersion=1,InteractionContactUUID=guid'0000c9e
9-49b6-1ed3-b3dd-
ffffffffffff',MchnLrngClfnStartDateTime=datetimeoffset'2017-04-01T00%3A00%3A00
Z') HTTP/1.1
--batch--
```

### Sample Code

## POST **Several Classification Values with Deep Create**

### Sample Code

```
{
  "PredictiveScenario": "CHURN_RENEWAL_SEASON_TICKET",
  "MchnLrngTargetObject": "Season Ticket",
  "MchnLrngClassificationVersion": 1,
  "MchnLrngClassificationVersText": "Version 1"
  "to_ClassificationValue":
  [
    {
      "InteractionContactUUID": "00000000-49b6-1ed3-b48c-ffffffffffff",
      "PredictiveScenario": "CHURN_RENEWAL_SEASON_TICKET",
      "MchnLrngClfnStartDateTime": "2016-04-01T00:00:00",
      "MchnLrngTargetObject": "Season Ticket",
      "MchnLrngClassificationVersion": 1,
      "MchnLrngClfnEndDateTime": "2017-03-31T23:59:59",
      "MchnLrngClassificationValue": "1"
    },
  ],
}
```

```
{
  "InteractionContactUUID": "0000c9e9-49b6-1ed3-b48c-ffffffffffff",
  "PredictiveScenario": "CHURN_RENEWAL_SEASON_TICKET",
  "MchnLrngClfnStartDateTime": "2016-04-01T00:00:00",
  "MchnLrngTargetObject": "Season Ticket",
  "MchnLrngClassificationVersion": 1,
  "MchnLrngClfnEndDateTime": "2017-03-31T23:59:59",
  "MchnLrngClassificationValue": "0"
}
```

#### POST **one Classification Value**

##### Sample Code

```
{
  "InteractionContactUUID": "00000000-49b6-1ed3-cccc-ffffffffffff",
  "PredictiveScenario": "CHURN_RENEWAL_SEASON_TICKET",
  "MchnLrngClfnStartDateTime": "2018-04-01T00:00:00",
  "MchnLrngTargetObject": "Season Ticket",
  "MchnLrngClassificationVersion": 1,
  "MchnLrngClfnEndDateTime": "2019-03-31T23:59:59",
  "MchnLrngClassificationValue": "0"
}
```

#### PUT **one Classification Value**

##### Sample Code

```
{
  "MchnLrngClfnEndDateTime": "2019-03-31T00:00:00",
  "MchnLrngClassificationValue": "1"
}
```

#### PATCH **one Classification Value**

##### Sample Code

```
{
  "MchnLrngClfnEndDateTime": "2019-03-31T00:00:00",
  "MchnLrngClassificationValue": "1"
}
```

#### PATCH **Several Classification Values in a Batch Request**

You can use a batch request to update classification values that are related to one classification only.

##### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
PATCH
ClassificationValues(PredictiveScenario='CHURN_RENEWAL_SEASON_TICKET',MchnLrng
TargetObject='Season
```

```
%20Ticket',MchnLrngClassificationVersion=1,InteractionContactUUID=guid'0000c9e
9-49b6-1ed3-b48c-
ffffffffffff',MchnLrngClfnStartDateTime=datetimeoffset'2016-04-01T00%3A00%3A00
Z') HTTP/1.1
Content-Type: application/json
Content-Length: ###
Sap-Cuan-SequenceId: CLASSIFICATION_BATCH_SINGLE
{
  "MchnLrngClassificationValue" : "1"
}
--changeset--
--batch--
```

### DELETE **one Classification Value**

#### Sample Code

```
/
ClassificationValues(PredictiveScenario='CHURN_RENEWAL_SEASON_TICKET',MchnLrng
TargetObject='Season
%20Ticket',MchnLrngClassificationVersion=1,InteractionContactUUID=guid'0000c9e
9-49b6-1ed3-b48c-
ffffffffffff',MchnLrngClfnStartDateTime=datetimeoffset'2016-04-01T00%3A00%3A00
Z')
```

## Machine Learning Scenarios

### GET **All Machine Learning Scenarios**

#### Sample Code

```
/sap/opu/odata/SAP/API_MKT_ML_CLASSIFICATION_SRV/MchnLrngScenarios
```

## 5.2.15 Marketing Attribute Categories

OData API (API\_MKT\_ATTRIBUTE\_CATEGORY) for writing master data about marketing attribute categories. Marketing attribute categories are freely-definable classifications of information that can be assigned to customers, for instance, to store their hobbies or education history.

### Technical Data

Name of the Service	API_MKT_ATTRIBUTE_CATEGORY
Communication Scenario IDs	SAP_COM_0207 and SAP_COM_0017

OData Version	1.0
Root URI	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_ATTRIBUTE_CATEGORY_SRV
Service Metadata URI:	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_ATTRIBUTE_CATEGORY_SRV/\$metadata
Field Extensibility Supported	No

## 5.2.15.1 Basic Concepts

### Creating and Updating Marketing Attribute Categories

1. Use the `API_MKT_ATTRIBUTE_CATEGORY` service to initially create the categories you require. With this service you can also load multiple language translations of the categories.
2. When you want to load individual marketing attribute values for your contacts, for example, to store their hobbies or the languages they speak, use the `MarketingAttribute` entity in the `API_MKT_CONTACT` service.

#### i Note

The entity `MarketingAttributeCategoryName` is an alternative key for a marketing attribute category that can replace the entity `MarketingAttributeCategory`. In other words, we accept either `MarketingAttributeCategory` or `MarketingAttributeCategoryName`. If you send both, the `MarketingAttributeCategoryName` is ignored. This means:

- If you send only the `MarketingAttributeCategoryName`, a marketing attribute category is created with a technical key and a description is provided in `MarketingAttributeCategoryName`.
- If you send the `MarketingAttributeCategory`, a marketing attribute category is created with a generated description (a timestamp is added to the name). You can change this name in the app `Marketing Attribute Categories`.

3. Use the app `Marketing Attribute Categories` to translate existing marketing attribute categories into different languages. With the app, you can also delete categories.

#### i Note

If multiple origins provide the same marketing attribute categories, these cannot be merged. Separate categories are created for each origin.

## Full Update

Blank entries overwrite existing entries. For example, if a marketing attribute category in the marketing system is stored with descriptions in the languages `EN`, `DE`, `IT`, and `ES` and a subsequent import only contains descriptions in the languages `EN`, `DE`, and `IT`, but not `ES`, the descriptions in language `ES` will be deleted.

## Consistency Checks

The ODATA Service performs the following consistency checks:

- Unknown language codes
- Description with language code missing
- Language code sent without a description
- No description sent at all
- Category sent more than once with different timestamps - the data set with the most recent timestamp is taken.
- Entry without an ID
- Entry without an ID Origin

## Error Message Handling

If the OData service is not accessible, for example due to missing authorization, or because the system is not available, a corresponding HTTP status code is returned.

If the OData service is accepted by the gateway component in the SAP Marketing Cloud system, the HTTP status code 201 or 204 is returned. Potential processing errors are recorded in the SAP Marketing Cloud system in the [Import Monitor](#) app, where they can be monitored, restarted and discarded.

By default, data processing is asynchronous. In most cases an OK response, such as a receipt notification, is returned almost immediately. An exception to this would be data uploads that might contain severe errors, such as parse or format errors, and so would not return an OK response but an error message. The data you upload lands in a staging area, where it is then further processed.

To view the processing status and to check for errors or success messages, you must launch the [Import Monitor](#) app. In the event of errors, you can restart or discard the import in the [Import Monitor](#).

## 5.2.15.2 Structure of API\_MKT\_ATTRIBUTE\_CATEGORY

This document describes the structure of the OData API service API\_MKT\_ATTRIBUTE\_CATEGORY. Make sure you read the Basic Concepts topic before you start.

### Request Header

The request header contains the following additional header fields:

Property	Example	Description	Edm Core Type	Max.Length	Mandatory
Sap-Cuan-RequestTimestamp	'2017-09-28T12:13:14'	Timestamp of the import run in the format: number of milliseconds since midnight Jan 1, 1970. For example: "/Date(1406014140922)/"	Edm.DateTime		X
Sap-Cuan-ReferenceMessage	345g67980907	External reference of the inbound message	Edm.String	32	

### Entity Sets

The MarketingAttributeCategory OData API provides the following entity sets:

Entity Set	Description	Path
MarketingAttributeCategories	This entity contains the ID and the Origin of ID. <div data-bbox="603 1608 991 1794"><b>i Note</b> If the same ID comes from multiple different origins, separate IDs are created.</div>	/MarketingAttributeCategories
MarketingAttributeCategoryNames	This entity contains the semantic name of the marketing attribute category in the relevant language.	/MarketingAttributeCategoryNames

## MarketingAttributeCategories

**POST:** https://<Server>:<Port> /sap/opu/odata/SAP/API\_MKT\_ATTRIBUTE\_CATEGORY\_SRV/\$batch.

You can perform the following operation on the MarketingAttributeCategories entity set:

HTTP Method	Description	Path
<i>POST</i>	Post a list of marketing attribute categories.	/ MarketingAttributeCategories?\$top=1

## MarketingAttributeCategoryNames

**POST:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_ATTRIBUTE\_CATEGORY\_SRV;v=0002/MarketingAttributeCategoryNames

You can perform the following operation on the MarketingAttributeCategoryName entity set:

HTTP Method	Description	Path
<i>POST</i>	Post a list of marketing attribute category names.	/ MarketingAttributeCategoryNames?\$top=1

### 5.2.15.3 Payload Examples

#### POST Marketing Attribute Categories

##### Sample Code

```
--batch
content-type:multipart/mixed;
boundary=changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a
content-type: application/http
content-transfer-encoding: binary
POST MarketingAttributeCategories HTTP/1.1
Content-Length: 1035
Accept: application/json
Content-Type: application/json
Sap-Cuan-RequestTimestamp: '2018-08-14T12:13:14'
sap-Cuan-ReferenceMessage: '12345678'
{
  "Id":"SH_20181029_001",
  "IdOrigin":"SAP_C4C_BUPA",
```

```

"MarketingAttributeCategoryNames": [
  { "LanguageCode": "E",
    "Name": "FavouriteBook" }
]
}
--changeset_761e49b6-3146-4a57-8d10-15816fb9c75a--
--batch--

```

## 5.2.16 Import Monitoring

Public OData API (`API_MKT_IMPORT_MONITORING`) for reading messages output for a specific data import using the import header ID. This service can be used by all API services whose imports are processed via the staging area.

### Technical Data

Name of the Service	API_MKT_IMPORT_MONITORING
Authorizations	The following business catalog is required: SAP_CEC_BC_MKT_API_IC2_PC
Communication Scenario IDs	SAP_COM_0003, SAP_COM_0206, SAP_COM_0207, SAP_COM_0264.  These are just some of the communication scenarios that implement this service. There may be others.
OData Version	2.0
Root URI	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_IMPORT_MONITORING_SRV
Service Metadata URI	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_IMPORT_MONITORING_SRV/\$metadata
Field Extensibility Supported	No

### Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:



Access Link	Remarks
<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_IMPORT_MONITORING_SRV/ \$metadata?sap- documentation=all</pre>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Import Monitoring Details Page</a>	<p>General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>1. On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>2. Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Import Monitoring API</a>	<p>General access link takes you directly to the <i>Import Monitoring</i> metadata file. One-time registration or logon is required.</p>

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## 5.2.16.1 Basic Concepts

Use this API Service to read the specific messages that are triggered in the staging area when you call other API services. This service returns the notifications that are also output in the *Import Monitor* app.

### Use

When you send data to the marketing system using a public API service, the import data can sometimes be processed in the staging area. The staging area returns a success message that the imported data is being processed. Using this service, you can query the status of the import, that is, whether import processing has been completed, as well as the status messages that are output, so that you can take prompt action where necessary.

This is a read-only service. You can only perform GET operations with it. With authorization for this service, you have access to all import header messages.

## 5.2.16.2 Structure of OData Service API\_MKT\_IMPORT\_MONITORING

This document describes the structure of the Public OData API service `API_MKT_IMPORT_MONITORING`.

### Entities

- **GET: Entity Path:** `/ImportHeaders`

You can perform the following operations on the `ImportHeader` entity set:

HTTP Method	Description	Path
<a href="#">GET</a>	Get a list of messages output for import headers. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> .	<code>/ImportHeaders?\$top=1</code>
	<b>i Note</b> <ul style="list-style-type: none"><li>◦ A maximum of 5000 import headers can be fetched in a single request.</li><li>◦ Specification of TOP is mandatory.</li></ul>	
	Get the details of a specific import header using the <code>ImportHeader</code> UUID.	<code>/ImportHeaders(guid '&lt;ImportHeader UUID&gt;')</code>

- **Entity Path:** `/ImportAggregatedMessage`

You **cannot** perform GET operations on the `ImportAggregatedMessage` entity, but you can expand from a given import header.

## 5.2.16.3 Payload Examples for Import Monitoring

The following examples show how you can use the Import Monitoring API service.

### GET Requests

#### Get all messages for a single import header (XXXX = enter the respective GUID)

```
/sap/opu/odata/SAP/API_MKT_IMPORT_MONITORING_SRV/ImportHeaders(guid'xxxxx')?
$expand=HeaderToMessage
```

### Get import headers with error messages

```
/sap/opu/odata/SAP/API_MKT_IMPORT_MONITORING_SRV/ImportHeaders?
$expand=HeaderToMessage&$top=2&$filter=Status eq '2'
```

### Get all messages for the top 2 import headers

```
/sap/opu/odata/SAP/API_MKT_IMPORT_MONITORING_SRV/ImportHeaders?
$expand=HeaderToMessage&$top=2
```

## 5.3 Landing Pages

The following integration APIs are available for landing pages:

- [External Landing Pages \[page 743\]](#)
- [External Landing Page Value Help \[page 749\]](#)

### 5.3.1 External Landing Pages

Public OData API (`API_MKT_LANDING_PAGE`) for writing external landing pages to the SAP Marketing Cloud system.

The API service is part of communication scenario `SAP_COM_0342`.

#### ⚠ Caution

It's possible to maintain two different target objects in the SAP Marketing Cloud system - *Forms* and *Landing Pages*. Depending on the target, you have the following options:

- To provide your external landing pages as **forms**, use `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_LANDING_PAGE_SRV/`. This is the default.
- To provide your external landing pages as **landing pages**, use `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_LANDING_PAGE_SRV?v=0002/`.

## Processing Information

Requests can be submitted in batch mode or in non-batch mode. Batch requests are submitted as a single `HTTP POST` request to the **\$batch** endpoint of a service as described in <https://www.odata.org/documentation/odata-version-2-0/uri-conventions/>. The batch request must contain a header parameter **content-type**, specifying the value **multipart/mixed** and **boundary=batch**.

**Root URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_LANDING_PAGE_SRV/`

## Technical Data

Technical Data of Service

<b>Name of Service</b>	API_MKT_LANDING_PAGE
<b>OData Version</b>	2.0
<b>Service Metadata URI</b>	https://<Server>:<Port>/sap/opu/odata/SAP/ API_MKT_LANDING_PAGE_SRV/\$metadata
<b>Service Metadata UI for Documentation of Properties</b>	https://<Server>:<Port>/sap/opu/odata/SAP/ API_MKT_LANDING_PAGE_SRV/\$metadata?sap-documen- tation=all
<b>Communication Scenario ID</b>	SAP_COM_0342
<b>Component for Incidents</b>	CEC-MKT-LPC
<b>Field Extensibility Supported</b>	No

### i Note

Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<b>Target Object: Forms</b>  <code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_LANDING_PAGE_SRV/ \$metadata?sap- documentation=all</code>	Only for internal access. You need to provide the server and port names.
<b>Target Object: Landing Pages</b>  <code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_LANDING_PAGE_SRV;v=000 2/\$metadata?sap- documentation=all</code>	

Access Link	Remarks
<ul style="list-style-type: none"> <li><a href="#">External Landing Page Metadata (Forms)</a></li> <li><a href="#">External Landing Page Metadata (Landing Pages) Version 2</a></li> </ul>	<p>General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<ul style="list-style-type: none"> <li><a href="#">External Landing Page Metadata (Forms) API</a></li> <li><a href="#">External Landing Page Metadata (Landing Pages) Version 2 API</a></li> </ul>	<p>General access link takes you directly to the <i>External Landing Pages</i> metadata files. One-time registration or logon is required.</p>

### Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Related Information

[Structure of OData Service API\\_MKT\\_LANDING\\_PAGE \[page 746\]](#)

[Payload Examples \[page 747\]](#)

## 5.3.1.1 Structure of OData Service API\_MKT\_LANDING\_PAGE

Complete list of entity sets for API\_MKT\_LANDING\_PAGE.

### Structure of OData Service API\_MKT\_LANDING\_PAGE

OData Service Structure

Entity	Description	Path
LandingPage	This entity contains the list of landing pages.	/LandingPages
Publication	This entity contains the list of publications for a landing page.	/Publications

#### LandingPages

LandingPages Entity

HTTP Method	Description	Path
GET	Get the list of landing pages	GET /LandingPages
	Get the details for a landing page	GET /LandingPages('Origin';External ID')
POST	Create a landing page	POST /LandingPages
PATCH	Create or delta update of a landing page. This creates a landing page if it does not exist.	PATCH /LandingPages('Origin';External ID')
PUT	Update a landing page	PUT /LandingPages('Origin';External ID')

#### Publications

Publications Entity

HTTP Method	Description	Path
GET	Get the list of publications	GET /Publications
	Get the details for a publication	GET /Publications ('Key')
POST	Create a publication	POST /Publications

## 5.3.1.2 Payload Examples

The following examples show how you can use the External Landing Pages API.

### GET: Get a Landing Page and its Publications

```
/sap/opu/odata/SAP/API_MKT_LANDING_PAGE_SRV/LandingPages?$filter=LandingPageOrigin
eq 'origin' and LandingPageExternalId eq 'id'&$top=100&$expand=Publications&
$select=Publications/LandingPagePublishedURL,Publications/
LandingPagePublicationUUID,Publications/LandingPageUUID
```

### POST: Landing Page

#### Example 1

##### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_1
--changeset_1
content-type: application/http
content-transfer-encoding: binary
POST LandingPages HTTP/1.1
Accept: application/json
Content-Type: application/json
{
    "LandingPageName" : "Landing Page Name",
    "LandingPageOrigin" : "Z_ORIGIN",
    "LandingPageExternalId" : "12345"
}
--changeset_1
content-type: application/http
content-transfer-encoding: binary
POST Publications HTTP/1.1
Accept: application/json
Content-Type: application/json
{
    "LandingPageOrigin" : "origin",
    "LandingPageExternalId" : "id",
    "LandingPagePublishedURL" : "http://www.<yourdomain>.com"
}
--changeset_1--
--batch--
```

#### Example 2

##### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_1
--changeset_1
content-type: application/http
```

```

content-transfer-encoding: binary
POST LandingPages HTTP/1.1
Accept: application/json
Content-Type: application/json
{
    "LandingPageName" : "Landing Page Name",
    "LandingPageOrigin" : "origin",
    "LandingPageExternalId" : "id",
    "Publications" :
    [
        {
            "LandingPageOrigin" : "origin",
            "LandingPageExternalId" : "id",
            "LandingPagePublishedURL" : "http://www.<yourdomain>.com"
        }
    ]
}
--changeset_1--
--batch--

```

## PATCH

### Sample Code

```

--batch
Content-Type: multipart/mixed; boundary=changeset_1
--changeset_1
content-type: application/http
content-transfer-encoding: binary
PATCH LandingPages (LandingPageOrigin='origin',LandingPageExternalId='id')
HTTP/1.1
Accept: application/json
Content-Type: application/json
{
    "LandingPageName" : "Landing Page Name",
    "LandingPageOrigin" : "origin",
    "LandingPageExternalId" : "id"
}
--changeset_1
content-type: application/http
content-transfer-encoding: binary
POST Publications HTTP/1.1
Accept: application/json
Content-Type: application/json
{
    "LandingPageOrigin" : "origin",
    "LandingPageExternalId" : "id",
    "LandingPagePublishedURL" : "http://www.<yourdomain>.com"
}
--changeset_1--
--batch--

```



## 5.3.2 External Landing Page Value Help

Public OData API (`API_MKT_LANDING_PAGE_VALUEHELP`) for retrieving attribute values used in landing pages. The API supports 14 attributes, such as Countries, Communication Categories, Marketing Areas, and Forms of Address.

The API service is part of communication scenario `SAP_COM_0342`.

### Processing Information

For a complete list of all entity sets, see [Structure of OData Service API\\_MKT\\_LANDING\\_PAGE\\_VALUEHELP \[page 752\]](#).

### Technical Data

Technical Data of Service

<b>Name of Service</b>	API_MKT_LANDING_PAGE_VALUEHELP
<b>OData Version</b>	2.0
<b>Service Metadata URI</b>	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_LANDING_PAGE_VALUEHELP_SRV/\$metadata
<b>Service Metadata UI for Documentation of Properties</b>	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_LANDING_PAGE_VALUEHELP_SRV/\$metadata?sap-documentation=all
<b>Communication Scenario ID</b>	SAP_COM_0342
<b>Component for Incidents</b>	CEC-MKT-LPC
	<b>i Note</b> Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a> .
<b>Field Extensibility Supported</b>	No

### Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_LANDING_PAGE_VALUEHELP _SRV/\$metadata?sap- documentation=all</pre>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Landing Page Value Help</a>	<p>General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">External Landing Page Value Help API</a>	General access link takes you directly to the <i>External Landing Page Value Help</i> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Overview

11 of the 14 attributes consist of 3 properties. They follow the pattern: **Code**, **Name/Description**, and **Language**.

Key fields are `Code` and `Language`. For example, the attribute `Industries` has a two-digit code for **Industry** and a name/description for **IndustryName**. The language in which an entry is returned is always called `Language`. The following are examples for the `Industries` attribute:

- Industry = 02
- IndustryName = Financial Services

### i Note

Name properties always end with **Name**.

- Language = EN

Or an entry as follows:

- Industry = 41

- IndustryName = Iron and Steel
- Language = EN

You can use a GET request with a filter in order to get a list of all Industry codes in English:

```
https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_LANDING_PAGE_VALUEHELP_SRV/Industries?$filter=Language eq 'EN'
```

3 of the 14 attributes supported consist of 4 properties. The following table gives you an overview:

#### Attributes and Properties

Attribute	Properties	Description
CommunicationCategory	CommunicationCategory CommunicationCategoryName IsNewsletter Marketing Area	The attribute does not have a Language property. The only key field is the code field CommunicationCategory. To retrieve only communication categories that can be subscribed to, you must set the filter IsNewsletter eq X. Marketing Area is also one of the properties, and there is also an attribute for Marketing Areas. The attribute is used to retrieve all active marketing areas.
Interaction Types	InteractionType InteractionTypeName CommunicationMedium Language	Interaction Types is an attribute that has three key fields. In addition to the code of the interaction and the language, the CommunicationMedium is needed to retrieve a unique interaction type. The communication medium must be set to WEB.
Regions	Region RegionName Country Language	The attribute for Regions has an additional key field for the country code (two-letter).

**Root URI:** https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_LANDING\_PAGE\_VALUEHELP\_SRV.

## Related Information

[Structure of OData Service API\\_MKT\\_LANDING\\_PAGE\\_VALUEHELP \[page 752\]](#)

[Payload Examples \[page 754\]](#)

## 5.3.2.1 Structure of OData Service API\_MKT\_LANDING\_PAGE\_VALUEHELP

Complete list of entity sets for API\_MKT\_LANDING\_PAGE\_VALUEHELP.

### OData Service Structure

Complete List of Entity Sets

Entity	Description	Path
Communication Category	Contains all communication categories. Newsletters can be retrieved with filter <code>IsNewsletter</code> equal to 'X'.	/CommunicationCategories
Country	Contains all countries in all available languages. Uses two-digit country code.	/Countries
Department	Contains company departments, for example Human Resources.	/Departments
Function	Contains professional functions, for example Marketing Manager.	/Functions
Gender	Contains the values Gender not known, Female, and Male.	/Genders
Industry	Contains lines of business, for example Financial Services.	/Industries
Interest	Contains all interests that can be found in the system.	/ItemsOfInterest
Language	Contains ISO Code, descriptions in all available languages, and the language of the description.	/Languages
Marital Status	A marital status can be Single or Married, for example.	/MaritalStatuses
Marketing Area	Delivers all active marketing areas.	/MarketingAreas
Region	Contains all regions of countries in all available languages. Due to the large response body, it is recommended to use a filter, for example <code>Language equals EN</code> .	/Regions

Entity	Description	Path
Form of Address	Contains the salutation, for example Mr. and Mrs. or Company.	/FormsOfAddress
Origin of ID	Contains the ID origin, for example, Phone Number or Email Address.	/Origins
Interaction Type	Contains interaction types such as <code>OptOut for Marketing Permission</code> . It must be filtered by <code>CommunicationMedium equals WEB</code> .	/InteractionTypes

Only the HTTP method `GET` is supported. All properties of all entity types can be filtered. The filter for `Language` works with every attribute except `Communication Categories` as it is not translated. You should use the two-letter ISO code for the [Language](#) field.

## Key Fields

List of Key Fields

Entity Set	Key Properties
Communication Category	CommunicationCategory
Country	Country, Language
Department	Department, Language
Function	InteractionContactFunction, Language
Gender	GenderCode, Language
Industry	Industry, Language
Interest	ItemOfInterest, Language
Language	LanguageISOCODE, Language
Marital Status	MaritalStatus, Language
Marketing Area	MarketingArea, Language
Region	Region, Country, Language
Form of Address	FormOfAddress, Language
Origin of ID	InteractionContactOrigin, Language

Entity Set	Key Properties
Interaction Type	InteractionType, CommunicationMedium, Language

## 5.3.2.2 Payload Examples

The following examples show how you can use the External Landing Page Value Help API.

### Example 1

```
https://<Server>:<Port>/sap/opu/odata/sap/API_MKT_LANDING_PAGE_VALUEHELP_SRV/Countries(Country='FR',Language='EN')
```

### Example 2

```
https://<Server>:<Port>/sap/opu/odata/sap/API_MKT_LANDING_PAGE_VALUEHELP_SRV/Functions?$filter=InteractionContactFunctionName eq 'Marketing Manager'
```

### Example 3

```
https://<Server>:<Port>/sap/opu/odata/sap/API_MKT_LANDING_PAGE_VALUEHELP_SRV/Regions(Region='BE',Country='DE',Language='EN')
```

### Example 4

```
https://<Server>:<Port>/sap/opu/odata/sap/api_mkt_landing_page_valuehelp_srv/Departments?$filter=Language eq 'EN'
```

## 5.4 Segmentation

## 5.4.1 Target Groups

Public OData API (API\_MKT\_TARGET\_GROUP\_SRV) for Target Groups

[Overview \[page 755\]](#)

### Overview

The public API for Target Groups supports operations on the `Target Group` Business Object.

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/\$metadata</code>
<b>Authorizations</b>	The following authorizations are required: <ul style="list-style-type: none"><li>• Business Role: <code>SAP_BCR_CEC_MKT_API_TGP_PC</code></li><li>• Business Catalog: <code>SAP_CEC_BC_MKT_API_TGP_PC</code></li></ul>
Component for Incidents	<ul style="list-style-type: none"><li>• CEC-MKT-TG</li></ul>
<b>i Note</b> Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a> .	
<b>Communication Scenario ID</b>	<code>SAP_COM_0205</code>
Field Extensibility Supported	No

### Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/\$metadata?sap-documentation=all</code>	Only for internal access. You need to provide the server and port names.

Access Link	Remarks
<a href="#">Marketing - Target Group Details Page</a>	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Target Groups API</a>	General access link takes you directly to the <i>Target Groups</i> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Related Information

<https://api.sap.com>

### 5.4.1.1 Structure of OData Service API\_MKT\_TARGET\_GROUP\_SRV

This document describes the Public OData API service `API_MKT_TARGET_GROUP_SRV` for Target Groups.

## Entity Sets

The Target Groups OData API provides the following entities:

Entity Set	Description	Path
TargetGroups	This entity contains target group data	/TargetGroups



Entity Set	Description	Path
TargetGroupInteractionContacts	This entity contains the interaction contacts of the target group	/TargetGroups(guid`<Target Group UUID>`)/TargetGroupInteractionContacts

You can view sample payloads and test the API at <https://api.sap.com>.

## TargetGroups

**Resource Path:** /TargetGroups

You can perform the following operations on the TargetGroups entity set:

Operations on TargetGroups entity set

HTTP Method	Description	Path
<i>GET</i>	Get a list of <i>Static</i> , <i>Dynamic</i> , and <i>Live</i> target groups. This method supports standard OData parameters such as <i>\$filter</i> , <i>\$select</i> , <i>\$top</i> , <i>\$orderby</i> and <i>\$skip</i> .	/TargetGroups?\$top=<Number of target groups>
<div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>The <i>\$top</i> parameter is mandatory.</li> <li>The <i>\$expand</i> parameter is not supported.</li> <li>You can get only 100 target groups with each request.</li> <li>You can get only the <i>Member Type 03 (Contacts)</i> from a Target Group.</li> </ul> </div>		
	Get the details of a specific target group using the Target Group UUID	/TargetGroups(guid`<Target Group UUID>`)
<i>POST</i>	Create a <i>Static</i> target group	/TargetGroups

Custom operations on TargetGroups entity set

HTTP Method	Description	Path
<i>POST</i>	Rebuild a <i>Dynamic</i> target group using the Target Group UUID	/RebuildTargetGroup?TargetGroupUUID=guid`<Target Group UUID>`

## TargetGroupInteractionContacts

**Resource Path:** /TargetGroups(guid`<Target Group UUID>`)/TargetGroupInteractionContacts

You can perform the following operations on the TargetGroupInteractionContacts entity set:

Operations on TargetGroupInteractionContacts entity set

HTTP Method	Description	URI
GET	<p>Get Target Group Interaction Contacts of a specific Interaction Contact Origin from the required target group.</p> <div data-bbox="357 465 900 1335" style="background-color: #f0f0f0; padding: 10px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>You must pass the <i>\$stop</i> parameter to get the following properties:                             <ul style="list-style-type: none"> <li>Target Group Interaction Contacts</li> <li>Interaction Contact Origin</li> </ul> </li> <li>You can get only 1000 target group interaction contacts with each request.</li> <li>The <i>\$expand</i> parameter is not supported.</li> <li>You must pass the <i>\$stop</i> and <i>\$filter</i> parameters to get the following properties:                             <ul style="list-style-type: none"> <li>Interaction Contact ID</li> <li>Interaction Contact Origin</li> </ul> </li> <li>The <i>\$filter</i> parameter is not mandatory in the following scenarios:                             <ul style="list-style-type: none"> <li>You use the <i>\$select</i> parameter to get the TargetGroupMemberUUID</li> <li>You use the <i>\$select</i> parameter to get the InteractionContactUUID</li> <li>You use the <i>\$select</i> parameter to get the TargetGroupUUID</li> </ul> </li> </ul> <p>To use the <i>\$filter</i> parameter with Interaction Contact Origin, the Interaction Contact Origin must not be defined as <i>Shareable</i> in the system.</p> </div>	<p>URI for <i>\$stop</i> and <i>\$filter</i> parameters:</p> <pre>/TargetGroups(guid'&lt;Target Group UUID&gt;')/TargetGroupInteractionContacts?\$stop=&lt;Number of interaction contacts&gt; &amp; \$filter=InteractionContactOrigin eq '&lt;Channel or Origin of the Interaction Contact&gt;'</pre> <p>URI for <i>\$stop</i> and <i>\$select</i> parameters:</p> <pre>/TargetGroups(guid'&lt;Target Group UUID&gt;')/TargetGroupInteractionContacts?\$stop=&lt;Number of interaction contacts&gt;&amp;\$format=json&amp;\$select=TargetGroupMemberUUID,InteractionContactUUID,TargetGroupUUID</pre>
POST	<p>Assign an Interaction Contact to a specific target group by using the Interaction Contact UUID.</p> <div data-bbox="357 1435 900 1581" style="background-color: #f0f0f0; padding: 10px;"> <p><b>i Note</b></p> <p>You can assign only the Member Type 03 (Contacts) to a Target Group.</p> </div>	<pre>/TargetGroups(guid'&lt;Target Group UUID&gt;')/TargetGroupInteractionContacts</pre>
	<p>Assign an Interaction Contact to a specific target group by using the Interaction Contact ID and Interaction Contact Origin.</p> <div data-bbox="357 1715 900 1888" style="background-color: #f0f0f0; padding: 10px;"> <p><b>i Note</b></p> <p>To assign a contact, the Interaction Contact Origin must not be defined as <i>Shareable</i> in your system.</p> </div>	<pre>/TargetGroups(guid'&lt;Target Group UUID&gt;')/TargetGroupInteractionContacts</pre>

## 5.4.1.2 Payload Examples

The following examples show how you can use the Target Groups API.

### Note

Delete request is not supported on Target Groups API.

## GET Requests - Examples

### Get the first 100 Target Groups

```
/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/TargetGroups?$format=json&$top=100
```

### Get the first 100 Target Groups filtered by Target Group Type and LifeCycle Status

```
/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/TargetGroups?
$filter=TargetGroupLifeCycleStatus eq '1' and TargetGroupCategoryName eq 'Static'&
$top=100&$format=json
```

### Get the first 100 Target Groups filtered by Marketing Area and Segmentation Object

```
/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/TargetGroups?$filter=MarketingArea eq
'CXXGLOBAL' and SegmentationObject eq 'SAP_CONTACT_ENGAGEMENT_SIN'&$top=100&
$format=json
```

### Get the first 1000 Contacts (UUID, Contact ID, and Contact Origin) belonging to a particular Target Group

```
/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/TargetGroups(guid'9CD400-0C70-1ED6-
BF9C-0C6E0BB242E9')/TargetGroupInteractionContacts?$filter=InteractionContactOrigin
eq 'SAP_CRM_BUPA'&
$select=InteractionContactUUID,TargetGroupUUID,InteractionContactId&$top=1000&
$format=json
```

## POST Requests - Examples

### Create a Static Target Group

```
/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/TargetGroups
```

#### Sample Code

```
{
  "TargetGroupName" : "DR-TG-2018-04-20T12:36:04.0000000",
  "MarketingArea" : "GLOBAL",
  "TargetGroupDescription" : "This API supports only Static TG
creation-2018-04-20T12:36:04.0000000",
  "TargetGroupMainResponsible" : "hsghds",
  "TargetGroupMemberType" : "03"
```

```
}
```

### Add a Contact, by its UUID to an existing Static Target Group

```
/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/TargetGroups(guid'9CDCD400-0C70-1ED6-BF9C-0C6E0BB242E9')/TargetGroupInteractionContacts
```

#### Sample Code

```
{
  "InteractionContactUUID": "8CDCD400-0C70-1ED6-BF9C-0C6E0BB242E9",
  "TargetGroupUUID": "9CDCD400-0C70-1ED6-BF9C-0C6E0BB242E9"
}
```

### Add a Contact, by its Contact ID and Contact Origin to an existing Static Target Group

```
/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/TargetGroups(guid'9CDCD400-0C70-1ED6-BF9C-0C6E0BB242E9')/TargetGroupInteractionContacts
```

#### Sample Code

```
{
  "TargetGroupUUID": "9CDCD400-0C70-1ED6-BF9C-0C6E0BB242E9",
  "InteractionContactId": "321981",
  "InteractionContactOrigin": "SAP_CRM_BUPA"
}
```

## OData Batch Requests - Examples

### POST a Target Group and Assign a Contact Using ICID and ICOriigin\_ \$batch

```
/sap/opu/odata/SAP/API_MKT_TARGET_GROUP_SRV/$batch
```

#### Sample Code

```
--batch_01869434-0005
Content-Type: multipart/mixed; boundary=changeset_01869434-0005-0001
--changeset_01869434-0005-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST TargetGroups HTTP/1.1
Content-Type: application/json
Content-Length: 1021
Content-ID: 1
{"TargetGroupName": "API TG_BATCH1
_2018-04-20T12:36:04.0000000", "MarketingArea": "GLOBAL", "TargetGroupMemberType"
:"03"}
--changeset_01869434-0005-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST $1/TargetGroupInteractionContacts HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{"InteractionContactId": "321981", "InteractionContactOrigin": "SAP_CRM_BUPA"}
```

```
--changeset_01869434-0005-0001--  
--batch_01869434-0005--
```

## 5.4.2 Export Target Groups and Target Group Member Data

With this integration you can export target groups to an external system.

For more information on how to export target group data, see:

- [Create Export File](#)
- [Export Definitions](#)

## 5.5 Campaign Management

### 5.5.1 Campaign and Target Group Data

With the OData Service `CUAN_INITIATIVE_SRV` you can retrieve certain attributes of campaigns and target groups, for example, for the recommendation scenario.

#### Overview

With the OData Service `CUAN_INITIATIVE_SRV` you can to retrieve certain attributes of campaigns and target groups, for example, for the recommendation scenario.

#### Details of the Service Entity

- **URLs:**  
`https://<Server>:<Port>/sap/opu/odata/sap/CUAN_INITIATIVE_SRV/Initiatives`  
`https://<Server>:<Port>/sap/opu/odata/sap/CUAN_INITIATIVE_SRV/TargetGroups`
- **Request Mode:** GET
- **Entity Data Model:** *CUAN Initiative* (`CUAN_INITIATIVE`)

**Support of OData Features:** See the following chapters for implementation details and search behavior of the OData services.

## Entity Type

### Initiative Entity Type

Property	Description	Edm Core Type	Max Length	Key
Name	The name of the campaign	Edm.String	40	No
Description	The description of the campaign	Edm.String	n.a.	No
InitiativeId	The identifier of the campaign	Edm.String	10	Yes
InitiativeExt	The external identifier of the campaign	Edm.String	10	No
LifeCycleStatus- StatusCode	The life cycle status code of the campaign	Edm.String	1	No
LifeCycleStatus- StatusDescription	The life cycle status description of the campaign	Edm.String	60	No
Search-SearchTerm	The search term of the campaign	Edm.String	n.a.	No
Search- TileFilterCategory	The tile filter category of the campaign	Edm.String	2	No
Category- CategoryCode	The category code of the campaign	Edm.String	3	No
Category- CategoryDescription	The category description of the campaign	Edm.String	60	No
Category- CategoryType	The category type of the campaign	Edm.Int16	n.a.	No
Filter- InteractionContactId	The interaction contact identifier of the campaign	Edm.String	n.a.	No
Filter- InteractionContactIdOrigin	The interaction contact identifier origin of the campaign	Edm.String	20	No

## Target Group Entity Type

Property	Description	Edm Core Type	Max Length	Key
TargetGroupId	The identifier of the target group	Edm.String	10	Yes
CustomerMemberCount	The customer member count of the target group	Edm.Int32	n.a.	No

## OData Service Call (GET) Examples Used in the Recommendation Scenario

Searching campaigns with a search term and additional filters:

### Sample Code

```
https://<Server>:<Port>/sap/opu/odata/sap/CUAN_INITIATIVE_SRV/Initiatives/?
$expand=TargetGroup&
$select=Name,Description,InitiativeId,InitiativeIdExt,LifecycleStatus,TargetGroup/
CustomerMemberCount&$filter=Search/SearchTerm eq 'tes' and Category/
CategoryCode eq '' and (Search/TileFilterCategory eq '1' or Search/
TileFilterCategory eq '2')
```

Selecting a specific campaign to create a customer segment:

### Sample Code

```
https://<Server>:<Port>/sap/opu/odata/sap/CUAN_INITIATIVE_SRV/
Initiatives('0000009108')/?$expand=TargetGroup&
$select=Name,Description,InitiativeId,InitiativeIdExt,LifecycleStatus,TargetGroup/
CustomerMemberCount
```

Getting campaigns for the current user with the specified filters:

### Sample Code

```
https://<Server>:<Port>/sap/opu/odata/sap/CUAN_INITIATIVE_SRV/Initiatives/?
$select=Name,Description,InitiativeId,InitiativeIdExt&$filter=Category/
CategoryCode eq '' and Search/TileFilterCategory eq '1' and (Filter/
InteractionContactId eq '1d998c85cc3d5205' or Filter/InteractionContactId eq
'john.dempsey@hana.com') and (Filter/InteractionContactIdOrigin eq 'EMAIL' or
Filter/InteractionContactIdOrigin eq 'COOKIE_ID')
```

## 5.5.2 Campaign Execution Plans

Campaign execution plans can be imported from other systems using a public OData application programming interface (API).

You can use the public CUAN\_MPO\_IMPORT\_SRV OData service to upload (import) campaign execution plans. The upload of campaign execution plans is always started through the `ImportHeaders` entity and a deep insert on the `ExecPlanItem` entity.

## Entity Data Model

The following tables list the details of the Campaign Execution Plan import service entities.

OData Version	2.0
Root URI	https://<Server>:<Port>/sap/opu/odata/SAP/ CUAN_MPO_IMPORT_SRV/
Authorizations	The following role is required: SAP_CEI_MPO_EXEC_PLAN_IMPORT
Support of OData Features	See the following chapters for implementation details and search behavior of the OData services.

### Entity Data Model: ImportHeader

Name	Is Key	Edm Core Type	Max Length	Creatable	Updatable	Sortable	Nullable	Filterable	Complex Type Name
Id	TRUE	Edm.String	--	TRUE	FALSE	FALSE	FALSE	FALSE	n.a.
Timestamp	FALSE	Edm.String	--	TRUE	FALSE	FALSE	TRUE	FALSE	n.a.
UserName	FALSE	Edm.String	12	TRUE	FALSE	FALSE	TRUE	FALSE	n.a.
SourceSystemType	FALSE	Edm.String	3	TRUE	FALSE	FALSE	FALSE	FALSE	n.a.
SourceSystemId	FALSE	Edm.String	20	TRUE	FALSE	FALSE	FALSE	FALSE	n.a.
ProcessingOrNot	FALSE	Edm.Boolean	--	TRUE	FALSE	FALSE	TRUE	FALSE	n.a.

### Entity Data Model: ExecPlanItem

Name	Is Key	Edm Core Type	Max Length	Creatable	Updatable	Sortable	Nullable	Filterable	Complex Type Name
Id	TRUE	Edm.String	10	FALSE	FALSE	FALSE	FALSE	FALSE	n.a.



Name	Is Key	Edm Core Type	Max Length	Creatable	Updatable	Sortable	Nullable	Filterable	Complex Type Name
OptimizationScenarioId	TRUE	Edm.String	20	TRUE	FALSE	FALSE	FALSE	FALSE	n.a.
CampaignId	FALSE	Edm.String	10	TRUE	FALSE	FALSE	TRUE	FALSE	n.a.
Timestamp	FALSE	Edm.DateTime	--	TRUE	FALSE	FALSE	TRUE	FALSE	n.a.
InteractionContactId	FALSE	Edm.String	255	TRUE	FALSE	FALSE	TRUE	FALSE	n.a.

The OData service provides the basic CRUD services as follows:

The upload of data is always started through the `ImportHeaders` entity and, in order to provide bulk processing, a deep insert on the `ExecPlanItem` entity (`CREATE_DEEP_ENTITY`).

The fields of the OData entities have the following meaning:

- `ImportHeader`
  - `Id`: A technical ID of one import service execution. In case no value is provided by the caller, an ID is generated by system.
  - `Timestamp`: Timestamp of the import run. In case no value is provided by the caller, a timestamp is generated by the system.
  - `UserName`: Name of the user who started the import. In case no value is provided by the caller, the system uses system name.
  - `SourceSystemType`: The type of the source system and can be freely defined, for example, **CRM** or **ERP**.
  - `SourceSystemId`: The ID of the source system and can be freely defined.
  - `ProcessAllOrNothing`: In case an error occurs, this flag defines if all imported offers are discarded or only the faulty ones. Default is **true**.
- `ExecPlanItem`
  - `Id`
  - `OptimizationScenarioId`
  - `CampaignId`
  - `Timestamp`
  - `InteractionContactId`

## Calling the OData Service Operation

### Create Request

The upload of campaign execution plans is started as a post request through the `ImportHeaders` entity and a deep insert on the `ExecPlanItem` entity (`CREATE_DEEP_ENTITY`). The following example shows the coding for creating a campaign execution plan.

URL (POST): `/sap/opu/odata/sap/CUAN_MPO_IMPORT_SRV/ImportHeaders`

#### Sample Code

```
POST data:
{
  "Id" : "",
  "Timestamp" : "2016-07-01T08:10:12",
  "SourceSystemType" : "EXT",
  "SourceSystemId" : "JMeter_Auto",
  "ExecPlanItems" : [
    {
      "Id" : "",
      "OptimizationScenarioId" : "PHONE",
      "CampaignId" : "234",
      "Timestamp" : "2016-06-16T13:10:12",
      "InteractionContactId" : "3440B5B11ACE1EE693DCDDFBB3B211B5"
    },
    {
      "Id" : "",
      "OptimizationScenarioId" : "PHONE",
      "CampaignId" : "321",
      "Timestamp" : "2016-07-16T17:11:03",
      "InteractionContactId" : "3440B5B11ACE1EE69E934168A6E95CEE"
    }
  ]
}
```

### Response

Example for response in case of successful creation:

#### Sample Code

```
{
  "d":
  {
    :   "__metadata":
    :   {
    :     :   "id": "<system>:<port>/sap/opu/odata/sap/CUAN_MPO_IMPORT_SRV/
ImportHeaders('E41D2DE534A01ED6A2F92AC2DD49165E')",
    :     :   "uri": "<system>:<port>/sap/opu/odata/sap/CUAN_MPO_IMPORT_SRV/
ImportHeaders('E41D2DE534A01ED6A2F92AC2DD49165E')",
    :     :   "type": "CUAN_MPO_IMPORT_SRV.ImportHeader"
    :   },
    :   "Id": "E41D2DE534A01ED6A2F92AC2DD49165E",
    :   "Timestamp": "\/Date(1467360612000)\/",
    :   "SourceSystemType": "EXT",
    :   "SourceSystemId": "JMeter_Auto",
    :   "Username": "AUT_TESTER",
    :   "ProcessAllOrNothing": false,
    :   "ExecPlanItems": null
    : }
  }
```

## 5.5.3 Campaigns

Public OData API (API\_MKT\_CAMPAIN\_SRV) for Campaigns.

### Technical Data

The public API for Campaigns supports operations on the Campaigns business object.

#### i Note

We recommend that you use the current version 0003 of this service. Don't revert to using version 0002 once you start using version 0003 since this may result in data inconsistencies. However, if you want to continue using the previous version, you'll find the help link here:

- Campaigns API, Version 0002: [Campaigns API, Version 0002](#)

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIN_SRV;V=3</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIN_SRV;V=3/\$metadata</code>
<b>Authorizations</b>	The following business catalog role is required: <ul style="list-style-type: none"><li>• SAP_CEC_BC_MKT_API_CPG3_PC</li></ul>
<b>Communication Scenario ID</b>	SAP_COM_0204
<b>Component for Incidents</b>	<ul style="list-style-type: none"><li>• CEC-MKT-CPG</li></ul>
	<h4>i Note</h4> <p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p>
<b>Field Extensibility Supported</b>	Yes

### Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CAMPAGN_SRV;v=0002/\$m etadata?sap-documentation=all</pre>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Campaign Details Page</a>	<p>General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Campaigns API</a>	General access link takes you directly to the <i>Campaign</i> metadata file. One-time registration or logon is required.

### Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## 5.5.3.1 Structure of API\_MKT\_CAMPAGN\_SRV

### Entities

The Campaign OData API provides the following entities:

Entity	Description	Path
Campaigns	This entity contains campaign data.	/Campaigns
AssignedTargetGroups	This entity contains target groups that are assigned to a campaign.	/CampaignAssignedTargetGroups
TeamMembers	This entity contains team members that are assigned to a campaign.	/CampaignAssignedTeamMembers

Entity	Description	Path
ActualSpends	This entity provides information on the actual spend amount and the committed amount of a campaign.	/ActualSpends
SpendHeaders	This entity provides generic information about spend period, status and whether <code>SpendItems</code> exists or not. <code>SpendHeaders</code> contains data as soon as spend information for a campaign is maintained.	/SpendHeaders
ExternalCampaignReferences	This entity contains external campaigns that are assigned to a campaign.	/ExternalCampaignReferences
Interests	This entity contains interests that are assigned to a campaign.	/CampaignAssignedInterests
SpendHeaderTimeSplits	This entity exists for each <code>SpendHeaders</code> and provides information on how the planned spend header amount is distributed over the period.	/SpendHeaderTimeSplits
SpendItemTimeSplits	This entity exists for each <code>SpendItem</code> and provides information on how the planned spend item amount is distributed to the spend periods.	/SpendItemTimeSplits
SpendItems	This entity contains data as soon as a campaign is planned in detail. It provides generic information about the spend item itself and its spend period.	/SpendItems

You can view sample payloads and test the API at <https://api.sap.com>.

## Campaigns

**Resource Path:** /Campaigns

You can perform the following operations on the Campaigns entity set:

Operations on Campaigns entity set

HTTP Method	Description	Path
GET	Get a list of campaigns. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> .	/Campaigns?\$top=1
<div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>The Campaigns OData API does not support Lead Transfer Campaigns (process type = '05').</li> <li>A maximum of 100 campaigns can be fetched in a single request.</li> </ul> </div>		
	Get the details of a specific campaign using the Campaign UUID.	/Campaigns(guid '<Campaign UUID>')

HTTP Method	Description	Path
	Get a list of campaigns.	/Campaigns
	Get a specific campaign using the Campaign UUID.	/ Campaigns({Campaign UUID})
	Get a list of target groups assigned to a specific campaign.	/ Campaigns({Campaign UUID})/ CampaignAssignedTar getGroups
	Get a list of assigned team members to a campaign using the Campaign UUID.	/ Campaigns({Campaign UUID})/ CampaignAssignedTea mMembers
	Get the actual spend of a specific campaign.	/ Campaigns({Campaign UUID})/ CampaignActualSpend
	Get a specific campaign's spend header information.	/ Campaigns({Campaign UUID})/ CampaignSpendHeader
	Get a list of external campaign references from a specific campaign.	/ Campaigns({Campaign UUID})/ ExternalCampaignRef erences
	Get a list of a campaign's assigned interests.	/ Campaigns({Campaign UUID})/ CampaignAssignedInt erests
<b>POST</b>	Create a campaign.	/Campaigns
	Create a campaign and assign a target group to the campaign by using the <a href="#">\$batch</a> parameter.	/Campaigns/\$batch
	Add a target group to a specific campaign.	/ Campaigns({Campaign UUID})/ CampaignAssignedTar getGroups

HTTP Method	Description	Path
	Assign team members to a specific campaign.	/ Campaigns ({Campaign UUID}) / CampaignAssignedTeamMembers
	Add spend headers to a specific campaign.	/ Campaigns ({Campaign UUID}) / CampaignSpendHeader
	Assign interests to a specific campaign.	/ Campaigns ({Campaign UUID}) / CampaignAssignedInterests
	Create a campaign from a template.	/ CreateCampaignFromTemplate
<i>PUT</i>	Update the data for an <i>In Preparation</i> campaign.	/ Campaigns (guid '<Campaign UUID>')
<i>PATCH</i>	Update the data for an <i>In Preparation</i> campaign. For example, you can update the Life Cycle Status of a campaign.	/ Campaigns ({Campaign UUID})

## AssignedTargetGroups

**Resource Path:** /AssignedTargetGroups

You can perform the following operations on the AssignedTargetGroups entity set:

Operations on AssignedTargetGroups entity set

HTTP Method	Description	Path
<i>GET</i>	Get a list of target groups assigned to the campaign.	/Campaigns ({CampaignUUID}) / CampaignAssignedTargetGroups
	Get the details of an assigned target group.	/ AssignedTargetGroups (TargetGroupUUID={TargetGroupUUID}, CampaignUUID={CampaignUUID})
	Get the details of all target groups.	/AssignedTargetGroups

HTTP Method	Description	Path
<i>POST</i>	Create an assigned target group for a specific campaign by using the Campaign UUID.	/Campaigns ({CampaignUUID}) / CampaignAssignedTargetGroups
	Assign target groups to a campaign.	/AssignedTargetGroups
<i>DELETE</i>	Delete an assigned target group from a campaign.	/ AssignedTargetGroups (TargetGroupUUID={TargetGroupUUID}, CampaignUUID={CampaignUUID})

## TeamMembers

**Resource Path:** /TeamMembers

You can perform the following operations on the TeamMembers entity set:

Operations on TeamMembers entity set

HTTP Method	Description	Path
<i>GET</i>	Get a list of team members assigned to a specific campaign.	/Campaigns ({CampaignUUID}) / CampaignAssignedTeamMembers
	Get the details of an assigned team member.	/ TeamMembers (CampaignUUID={CampaignUUID}, TeamMemberName= '{TeamMemberName}')
	Get the details of all team members.	/TeamMembers
<i>POST</i>	Create an assigned interest for a specific campaign by using the Campaign UUID.	/Campaigns ({CampaignUUID}) / CampaignAssignedTeamMembers
	Add new team members.	/TeamMembers
<i>PATCH</i>	Update the data of a team member. For example, you can change the owner of a campaign.	/ TeamMembers (CampaignUUID={CampaignUUID}, TeamMemberName= '{TeamMemberName}')
<i>DELETE</i>	Delete an assigned team member from a campaign.	/ TeamMembers (CampaignUUID={CampaignUUID}, TeamMemberName= '{TeamMemberName}')

## ActualSpends

**Resource Path:** /ActualSpends



You can perform the following operations on the `ActualSpends` entity set:

Operations on `ActualSpends` entity set

HTTP Method	Description	Path
<code>GET</code>	Get a list of actual spends.	<code>/ActualSpends</code>
	Get a list of actual spends for a specific campaign.	<code>/Campaigns({CampaignUUID})/CampaignActualSpend</code>
	Get actual spend information from a specific campaign.	<code>/ActualSpends({CampaignUUID})</code>

## SpendHeaders

**Resource Path:** `/SpendHeaders`

You can perform the following operations on the `SpendHeaders` entity set:

Operations on `SpendHeaders` entity set

HTTP Method	Description	Path
<code>GET</code>	Get specific spend header information. <div data-bbox="603 1196 991 1368"><b>Note</b> Do not use <code>GET SpendHeader</code> if <code>IsPlannedInDetail == true</code>.</div>	<code>/SpendHeaders({MarketingSpendHeaderUUID})</code>
	Get a list of spend headers.	<code>/SpendHeaders</code>
	Get a spend header for a campaign.	<code>/Campaigns({CampaignUUID})/CampaignSpendHeader</code>
	Get the header time split from a specific spend header.	<code>/SpendHeaders({MarketingSpendHeaderUUID})/HeaderTimeSplit</code>
	Get a list of items from a specific spend header.	<code>/SpendHeaders({MarketingSpendHeaderUUID})/Item</code>

HTTP Method	Description	Path
<i>POST</i>	Create a spend header for an existing campaign.	/SpendHeaders
	Deep create: Create a spend header including its spend items and spend item time splits for an existing campaign.	
	Add a campaign spend header to a specific campaign.	/ Campaigns ({CampaignUUID}) / CampaignSpendHeader
<i>PATCH</i>	Add a list of items to a specific spend header.	/ SpendHeaders ({MarketingSpendHeaderUUID}) /Item
	Update a specific spend header.	/ SpendHeaders ({MarketingSpendHeaderUUID})

## ExternalCampaignReferences

**Resource Path:** /ExternalCampaignReferences

You can perform the following operations on the ExternalCampaignReferences entity set:

Operations on ExternalCampaignReferences entity set

HTTP Method	Description	Path
<i>GET</i>	Get a list of external campaign references for a specific campaign.	/ Campaigns ({CampaignUUID}) / ExternalCampaignReferences
	Get details of a specific external campaign reference.	/ ExternalCampaignReferences ({ExternalCampaignUUID})
	Get a list of external campaign references.	/ ExternalCampaignReferences

## Interests

**Resource Path:** /Interests

You can perform the following operations on the Interests entity set:

Operations on Interests entity set

HTTP Method	Description	Path
<i>GET</i>	Get a list of interests assigned to a specific campaign.	/Campaigns ({CampaignUUID}) / CampaignAssignedInterests
	Get the details of an assigned interest.	/ Interests (CampaignUUID={CampaignUUID} , InterestItem=' {InterestItem} ')
	Get the details of all interests.	/Interests
<i>POST</i>	Create an assigned interest for a specific campaign by using the Campaign UUID.	/Campaigns ({CampaignUUID}) / CampaignAssignedInterests
	Add new interests.	/Interests
<i>DELETE</i>	Delete an assigned interest from a campaign.	/ Interests (CampaignUUID={CampaignUUID} , InterestItem=' {InterestItem} ')

## SpendHeaderTimeSplits

**Resource Path:** /SpendHeaderTimeSplits

You can perform the following operations on the SpendHeaderTimeSplits entity set:

Operations on SpendHeaderTimeSplits entity set

HTTP Method	Description	Path
<i>GET</i>	Get specific spend header time split information.	/ SpendHeaderTimeSplits ({MarketingSpendHdrTimeSplitUUID})
	Get a list of spend header time splits.	/SpendHeaderTimeSplits
	Get spend header time splits for a specific spend header.	/ SpendHeaders ({MarketingSpendHeaderUUID}) / HeaderTimeSplit

## SpendItemTimeSplits

**Resource Path:** /SpendItemTimeSplits

You can perform the following operations on the `SpendItemTimeSplits` entity set:

Operations on `SpendItemTimeSplits` entity set

HTTP Method	Description	Path
<i>GET</i>	Get specific spend item time split information.	<code>/SpendItemTimeSplits({MarketingSpendItemTimeSplitUUID})</code>
	Get a list of spend item time splits.	<code>/SpendItemTimeSplits</code>
	Get spend item time splits for a specific spend item.	<code>/SpendItems({MarketingSpendItemUUID})/ItemTimeSplit</code>
<i>POST</i>	Create spend item time splits.	<code>/SpendItemTimeSplits</code>
	Create a spend item time split and add them to a specific spend item.	<code>/SpendItems({MarketingSpendItemUUID})/ItemTimeSplit</code>
<i>PATCH</i>	Update a specific spend item time split.	<code>/SpendItemTimeSplits({MarketingSpendItemTimeSplitUUID})</code>
<i>DELETE</i>	Delete a specific spend item time split.	<code>/SpendItemTimeSplits({MarketingSpendItemTimeSplitUUID})</code>

## SpendItems

**Resource Path:** `/SpendItems`

You can perform the following operations on the `SpendItems` entity set:

Operations on `SpendItems` entity set

HTTP Method	Description	Path
<i>GET</i>	Get specific spend item information.	<code>/SpendItems({MarketingSpendItemUUID})</code>
	Get a list of spend items.	<code>/SpendItems</code>
	Get spend items for a specific spend header.	<code>/SpendHeaders({MarketingSpendHeaderUUID})/Item</code>

HTTP Method	Description	Path
	Get item time splits for a specific spend item.	/SpendItems ({MarketingSpendItemUUID}) /ItemTimeSplit
<i>POST</i>	Create a spend item.  Deep create: Create a spend item including its spend item time splits for an existing spend header.	/SpendItems
	Add item time splits to a specific spend item.	/SpendItems ({MarketingSpendItemUUID}) /ItemTimeSplit
	Add items to a specific spend header.	/SpendHeaders ({MarketingSpendHeaderUUID}) /Item
<i>PATCH</i>	Update a specific spend item.	/SpendItems ({MarketingSpendItemUUID})
<i>DELETE</i>	Delete a specific spend item and all related spend item time splits.	/SpendItems ({MarketingSpendItemUUID})

## 5.5.3.2 Payload Examples

The following examples show how you can use the Campaigns API.

### GET Requests - Examples

#### Get the First 100 Campaigns

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/Campaigns?$top=100&$format=json
```

#### Get the First 100 Campaigns Filtered by Campaign Category and LifeCycle Status

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/Campaigns?
$filter=CampaignLifecycleStatus eq '1' and CampaignCategory eq 'CME'&$top=100&
$format=json
```

#### Get the First 100 Campaigns Filtered by Marketing Area and Media Type

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/Campaigns?$filter=MediaType eq 'PRINT'
and MarketingArea eq 'CXXGLOBAL'&$top=100&$format=json
```

## Get a Campaign's Assigned Target Group, Interests, Assigned Team Members

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')?  
$expand=CampaignAssignedTargetGroups,CampaignAssignedInterests,CampaignAssignedTeam  
Members&$format=json
```

## POST Requests - Examples

### Post a Campaign

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/Campaigns
```

#### Sample Code

```
{  
  
  "CampaignName": "Email Campaign 1SY 2018-04-20T12:36:04.0000000",  
  "CampaignDescription" : "V2 New Description 2018-04-20T12:36:04.0000000",  
  "MarketingArea" : "Global",  
  "CampaignCategory" : "CME",  
  "CommunicationCategoryUUID" : "1C98EC18-1855-1EE7-A8BF-713D0AF485F8",  
  "MarketingProgramUUID": "0050569F-4A52-1ED7-8481-8A95A404CF53",  
  "MediaType": "PRINT",  
  "CampaignPriority" : "3",  
  "CampaignExecutionFrqcyUnit" : "2"  
}
```

### Assign a Target Group to a Created Campaign

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/CampaignAssignedTargetGroups
```

#### Sample Code

```
{  
  "TargetGroupUUID": "9CDCD400-0C70-1ED6-BF9C-0C6E0BB242E9"  
}
```

### Assign a Team Member to a Campaign

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/CampaignAssignedTeamMembers
```

### Assign an Interest to a Created Campaign

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/CampaignAssignedInterests
```

#### Sample Code

```
{  
  "CampaignUUID": "94188283-1c7d-1ed9-82d1-59d7230c7110",  
  "InterestItem": "CAMERA"
```

```
}
```

## Update Requests - Examples

### PUT a Campaign

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')
```

#### Sample Code

```
{  
  "CampaignName" : "Email Campaign 1SY 2018-04-20T12:36:04.0000000",  
  "CampaignDescription" : "DR_PUT-DESC Change",  
  "CampaignScheduleDateTime" : "2018-07-30T11:36:00",  
  "CampaignExecutionFrqcyInterval" : "001",  
  "CampaignExecutionFrqcyUnit" : "3",  
  "MediaType": "PRINT",  
  "MarketingProgramUUID" : "0050569F-4A52-1ED7-8481-8A95A404CF53",  
  "CommunicationCategoryUUID" : "1C98EC18-1855-1EE7-A8BF-713D0AF485F8",  
  "CampaignPriority" : "1",  
  "CampaignStartDate" : "2018-06-22T00:00:00",  
  "CampaignEndDate" : "2018-08-22T00:00:00",  
  "CampaignOwner" : "CB9980000130"  
}
```

### PUT a Campaign's Team Member

```
sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
TeamMembers(CampaignUUID=guid'94188283-1c7d-1ed9-82d1-59d7230c7110', TeamMemberName=  
'CB9980000130')
```

#### Sample Code

```
{  
  "TeamMemberName": "ERIKA",  
  "IsOwner" : true  
}
```

## Delete Requests - Examples

### Unassign a Target Group from a Campaign

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
AssignedTargetGroups(TargetGroupUUID=guid'9CD400-0C70-1ED6-  
BF9C-0C6E0BB242E9', CampaignUUID=guid'94188283-1c7d-1ed9-82d1-59d7230c7110')
```

## Unassign a Team Member from a Campaign

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
TeamMembers(CampaignUUID=guid'94188283-1c7d-1ed9-82d1-59d7230c7110',TeamMemberName=  
'CB9980000130')
```

## Unassign an Interest from a Campaign

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/  
Interests(CampaignUUID=guid'94188283-1c7d-1ed9-82d1-59d7230c7110',InterestItem='CAM  
ERA')
```

## Batch OData Requests - Examples

### Create Campaigns in a Batch

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/$batch
```

#### Sample Code

```
--batch_020c-a527-decc  
Content-Type: multipart/mixed; boundary=changeset_9970-5898-d67d  
--changeset_9970-5898-d67d  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
POST Campaigns?sap-client=100 HTTP/1.1  
sap-contextid-accept: header  
Accept: application/json  
Accept-Language: en  
DataServiceVersion: 2.0  
MaxDataServiceVersion: 2.0  
sap-cancel-on-close: true  
Content-Type: application/json  
Content-Length: 588  
{  
  "CampaignName": "DR CPG -1  
2018-04-20T12:36:04.0000000", "CampaignDescription": "My CPG  
2018-04-20T12:36:04.0000000", "MarketingArea": "GLOBAL", "CampaignCategory": "CME"  
}  
--changeset_9970-5898-d67d  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
POST Campaigns?sap-client=100 HTTP/1.1  
sap-contextid-accept: header  
Accept: application/json  
Accept-Language: en  
DataServiceVersion: 2.0  
MaxDataServiceVersion: 2.0  
sap-cancel-on-close: true  
Content-Type: application/json  
Content-Length: 588  
{  
  "CampaignName": "DR CPG -2  
2018-04-20T12:36:04.0000000", "CampaignDescription": "My  
CPG2018-04-20T12:36:04.0000000", "MarketingArea": "GLOBAL", "CampaignCategory": "C  
ME"  
}  
--changeset_9970-5898-d67d  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
POST Campaigns?sap-client=100 HTTP/1.1  
sap-contextid-accept: header  
Accept: application/json  
Accept-Language: en
```



```

DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
Content-Type: application/json
Content-Length: 588
{"CampaignName":"DR CPG -3
2018-04-20T12:36:04.0000000","CampaignDescription":"My
CPG2018-04-20T12:36:04.0000000","MarketingArea":"GLOBAL","CampaignCategory":"C
ME"}
--changeset_9970-5898-d67d
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns?sap-client=100 HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
Content-Type: application/json
Content-Length: 588
{"CampaignName":"DR CPG -4
2018-04-20T12:36:04.0000000","CampaignDescription":"My
CPG2018-04-20T12:36:04.0000000","MarketingArea":"2018-04-20T12:36:04.0000000",
"CampaignCategory":"CME"}
--changeset_9970-5898-d67d
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns?sap-client=100 HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
Content-Type: application/json
Content-Length: 588
{"CampaignName":"DR CPG -5
2018-04-20T12:36:04.0000000","CampaignDescription":"My
CPG2018-04-20T12:36:04.0000000","MarketingArea":"GLOBAL","CampaignCategory":"C
ME"}
--changeset_9970-5898-d67d--
--batch_020c-a527-decc--

```

## Assign Target Group to a Campaign in a Batch

/sap/opu/odata/SAP/API\_MKT\_CAMPaign\_SRV;v=3/\$batch

### Sample Code

```

--batchtest01
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/
CampaignAssignedTargetGroups HTTP/1.1
Content-Type: application/json
Content-Length: 588
{
  "CampaignUUID": "94188283-1c7d-1ed9-82d1-59d7230c7110",
  "TargetGroupUUID": "9CDCD400-0C70-1ED6-BF9C-0C6E0BB242E9"
}
--changeset

```

```

Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/
CampaignAssignedTargetGroups HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 588
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:CampaignUUID>94188283-1c7d-1ed9-82d1-59d7230c7110</d:CampaignUUID>
<d:TargetGroupUUID>9CDCD400-0C70-1ED6-BF9C-0C6E0BB242E9</d:TargetGroupUUID>
</m:properties>
</atom:content>
</atom:entry>
--changeset--
--batchtest01--

```

## Assign Team Members to a Campaign in a Batch

/sap/opu/odata/SAP/API\_MKT\_CAMPAIGN\_SRV;v=3/\$batch

### Sample Code

```

--batchtest
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/
CampaignAssignedTeamMembers HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 588
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:CampaignUUID>94188283-1c7d-1ed9-82d1-59d7230c7110</d:CampaignUUID>
<d:TeamMemberName>CB9980000130</d:TeamMemberName>
</m:properties>
</atom:content>
</atom:entry>
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/
CampaignAssignedTeamMembers HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 588
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:CampaignUUID>94188283-1c7d-1ed9-82d1-59d7230c7110</d:CampaignUUID>
<d:TeamMemberName>CB9980000130</d:TeamMemberName>
</m:properties>
</atom:content>
</atom:entry>
--changeset--

```

```
--batchtest--
```

## Assign Interests to a Campaign in a Batch

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_SRV;v=3/$batch
```

### Sample Code

```
--batchtest
Content-Type: multipart/mixed; boundary=changeset
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/
CampaignAssignedInterests HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 588
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:CampaignUUID>94188283-1c7d-1ed9-82d1-59d7230c7110</d:CampaignUUID>
<d:InterestItem>CAMERA</d:InterestItem>
</m:properties>
</atom:content>
</atom:entry>
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/
CampaignAssignedInterests HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 588
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:CampaignUUID>94188283-1c7d-1ed9-82d1-59d7230c7110</d:CampaignUUID>
<d:InterestItem>CAMERA</d:InterestItem>
</m:properties>
</atom:content>
</atom:entry>
--changeset
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns(guid'94188283-1c7d-1ed9-82d1-59d7230c7110')/
CampaignAssignedInterests HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 588
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:CampaignUUID>94188283-1c7d-1ed9-82d1-59d7230c7110</d:CampaignUUID>
<d:InterestItem>CAMERA</d:InterestItem>
</m:properties>
</atom:content>
</atom:entry>
--changeset--
--batchtest--
```

## Spend Header, Spend Item, and Spend Item Time Splits

### Create One Spend Header, One Spend Item, and Three Spend Item Time Splits

The following is a deep create example using content-id and one batch request:

#### Sample Code

```
--batchtest_085421042017-0001
Content-Type: multipart/mixed; boundary=changeset_085421042017-0001-0001
--changeset_085421042017-0001-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Campaigns HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 999
Content-ID: 1
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:CampaignName>Email Campaign Batch 1</d:CampaignName>
<d:CampaignDescription>Email Campaign</d:CampaignDescription>
<d:MarketingArea>GERMANY</d:MarketingArea>
</m:properties>
</atom:content>
</atom:entry>
--changeset_085421042017-0001-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST $1/CampaignSpendHeader HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 999
Content-ID: 2
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingSpendPeriodStartYear>2018</d:MarketingSpendPeriodStartYear>
<d:MarketingSpendPeriodStartMonth>01</d:MarketingSpendPeriodStartMonth>
<d:MarketingSpendPeriodEndYear>2018</d:MarketingSpendPeriodEndYear>
<d:MarketingSpendPeriodEndMonth>04</d:MarketingSpendPeriodEndMonth>
<d:PlannedMktgSpendHeaderAmt>5000.00</d:PlannedMktgSpendHeaderAmt>
<d:PlannedMktgSpendHeaderCrcy>USD</d:PlannedMktgSpendHeaderCrcy>
</m:properties>
</atom:content>
</atom:entry>
--changeset_085421042017-0001-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST $2/Item HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 999
Content-ID: 10
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
```

```

xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingSpendType>DIGITAL</d:MarketingSpendType>
<d:MarketingSpendItemName>Digi Costs</d:MarketingSpendItemName>
<d:MarketingSpendPeriodStartYear>2018</d:MarketingSpendPeriodStartYear>
<d:MarketingSpendPeriodStartMonth>02</d:MarketingSpendPeriodStartMonth>
<d:MarketingSpendPeriodEndYear>2018</d:MarketingSpendPeriodEndYear>
<d:MarketingSpendPeriodEndMonth>04</d:MarketingSpendPeriodEndMonth>
</m:properties>
</atom:content>
</atom:entry>
--changeset_085421042017-0001-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST $10/ItemTimeSplit HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 999
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingSpendCalendarYear>2018</d:MarketingSpendCalendarYear>
<d:MarketingSpendCalendarQuarter>1</d:MarketingSpendCalendarQuarter>
<d:MarketingSpendCalendarMonth>02</d:MarketingSpendCalendarMonth>
<d:PlannedMktgSpendAmt>2000.00</d:PlannedMktgSpendAmt>
<d:PlannedMktgSpendCrcy>USD</d:PlannedMktgSpendCrcy>
</m:properties>
</atom:content>
</atom:entry>
--changeset_085421042017-0001-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST $10/ItemTimeSplit HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 999
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingSpendCalendarYear>2018</d:MarketingSpendCalendarYear>
<d:MarketingSpendCalendarQuarter>1</d:MarketingSpendCalendarQuarter>
<d:MarketingSpendCalendarMonth>03</d:MarketingSpendCalendarMonth>
<d:PlannedMktgSpendAmt>3000.00</d:PlannedMktgSpendAmt>
<d:PlannedMktgSpendCrcy>USD</d:PlannedMktgSpendCrcy>
</m:properties>
</atom:content>
</atom:entry>
--changeset_085421042017-0001-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST $10/ItemTimeSplit HTTP/1.1
Content-Type: application/atom+xml
Content-Length: 999
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<atom:content type="application/xml">
<m:properties>
<d:MarketingSpendCalendarYear>2018</d:MarketingSpendCalendarYear>
<d:MarketingSpendCalendarQuarter>1</d:MarketingSpendCalendarQuarter>
<d:MarketingSpendCalendarMonth>04</d:MarketingSpendCalendarMonth>
<d:PlannedMktgSpendAmt>4000.00</d:PlannedMktgSpendAmt>

```

```

<d:PlannedMktgSpendCrcy>USD</d:PlannedMktgSpendCrcy>
</m:properties>
</atom:content>
</atom:entry>
--changeset_085421042017-0001-0001--
--batchtest_085421042017-0001--

```

## Spend Header

### Create Spend Header

#### Sample Code

```

{
  "MarketingSpendStatus" : "00",
  "CampaignUUID" : "6C0B84B7-5523-1ED7-BBF7-F8A14CC949B6",
  "MarketingSpendPeriodStartYear" : "2018",
  "MarketingSpendPeriodStartMonth" : "03",
  "MarketingSpendPeriodEndYear" : "2018",
  "MarketingSpendPeriodEndMonth" : "04",
  "PlannedMktgSpendHeaderAmt" : "3000.00",
  "PlannedMktgSpendHeaderCrcy" : "USD"
}

```

### Update Spend Header Period and Planned Spend

#### Sample Code

```

{
  "MarketingSpendPeriodStartYear" : "2018",
  "MarketingSpendPeriodStartMonth" : "04",
  "MarketingSpendPeriodEndYear" : "2018",
  "MarketingSpendPeriodEndMonth" : "05",
  "PlannedMktgSpendHeaderAmt" : "10000",
  "PlannedMktgSpendHeaderCrcy" : "USD"
}

```

## Spend Item

### Create Spend Item

#### Sample Code

```

{
  "MarketingSpendHeaderUUID" : "6C0B84B7-5523-1EE7-BED2-B47CFBED5A07",
  "MarketingSpendType" : "DIGITAL",
  "MarketingSpendItemName" : "Print Costs",
  "MarketingSpendPeriodStartYear" : "2018",
  "MarketingSpendPeriodStartMonth" : "04",
  "MarketingSpendPeriodEndYear" : "2018",
  "MarketingSpendPeriodEndMonth" : "04"
}

```

## Create One Spend Item and 2 Spend Item Time Splits

### Sample Code

```
{
  "MarketingSpendHeaderUUID" : "6C0B84B7-5523-1ED7-BED4-79CB3454407E",
  "MarketingSpendType" : "Print",
  "MarketingSpendItemName" : "Paper",
  "MarketingSpendPeriodStartYear" : "2018",
  "MarketingSpendPeriodStartMonth" : "04",
  "MarketingSpendPeriodEndYear" : "2018",
  "MarketingSpendPeriodEndMonth" : "05",
  "ItemTimeSplit" : [
    {
      "MarketingSpendCalendarYear" : "2018",
      "MarketingSpendCalendarQuarter" : "2",
      "MarketingSpendCalendarMonth" : "04",
      "PlannedMktgSpendAmt" : "4000.00",
      "PlannedMktgSpendCrcy" : "USD"
    },
    {
      "MarketingSpendCalendarYear" : "2018",
      "MarketingSpendCalendarQuarter" : "2",
      "MarketingSpendCalendarMonth" : "05",
      "PlannedMktgSpendAmt" : "5000.00",
      "PlannedMktgSpendCrcy" : "USD"
    }
  ]
}
```

## Spend Item Time Splits

### Create Spend Item Time Splits

### Sample Code

```
{
  "MarketingSpendItemUUID" : "6C0B84B7-5523-1ED8-80D0-9AAE7F2C3528",
  "MarketingSpendCalendarYear" : "2018",
  "MarketingSpendCalendarQuarter" : "1",
  "MarketingSpendCalendarMonth" : "03",
  "PlannedMktgSpendAmt" : "5999.00",
  "PlannedMktgSpendCrcy" : "USD"
}
```

## 5.5.3.3 Function Imports

Function imports are used to perform custom operations on an entity, in addition to typical OData operations.

### Create Campaign Using Campaign Template ID and Description

HTTP Method	Description	Path
<i>POST</i>	Create a campaign using the campaign template ID and description	<pre>/ CreateCampaignFromTemplate? TemplateID='&lt;Campaign Template ID&gt;' &amp;CampaignName=' &lt;Description&gt;'</pre>

## 5.5.4 Campaign Templates

Public OData API (API\_MKT\_CAMPAIGN\_TEMPLATE\_SRV) for Campaign Templates.

### Technical Data

The public API for Campaign Templates supports operations on the Campaign Templates business object.

OData Version	2.0
Root URI	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIGN_TEMPLATE_SRV</code>
Service Metadata URI	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIGN_TEMPLATE_SRV/\$metadata</code>
Authorizations	The following business catalog role is required: <ul style="list-style-type: none"><li>SAP_CEC_BC_MKT_API_CPG_PC</li></ul>
Communication Scenario ID	SAP_COM_0204



Component for Incidents

- CEC-MKT-CPG

### i Note



Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

Field Extensibility Supported

No

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CAMPAIGN_TEMPLATE_SRV/ \$metadata?sap- documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Campaign Templates Page</a> 	General access to the <a href="#">Details</a> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>1. On the <a href="#">Details</a> page, click <a href="#">Download Specification</a> and download as EDMX.</li><li>2. Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Campaign Templates API</a> 	General access link takes you directly to the <a href="#">Campaign Templates</a> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Related Information

<https://api.sap.com>

### 5.5.4.1 Structure of API\_MKT\_CAMPAIGN\_TEMPLATE\_SRV

#### Entity Data Model

The Campaign Template OData API provides the following entities:

Entity	Description	Path
CampaignTemplates	This entity contains campaign template data.	/CampaignTemplates
AssignedInterests	This entity contains interests that are assigned to a campaign template.	/AssignedInterests
AssignedTargetGroups	This entity contains target groups that are assigned to a campaign template.	/CampaignAssignedTargetGroups
TeamMembers	This entity contains team members that are assigned to a campaign template.	/TeamMembers

You can view sample payloads and test the API at <https://api.sap.com>.

#### CampaignTemplates

**Resource Path:** /CampaignTemplates

You can perform the following operations on the CampaignTemplates entity set:

Operations on CampaignTemplates entity set

HTTP Method	Description	Path
<a href="#">GET</a>	Get a list of campaigns. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , and <code>\$orderby</code> .	/CampaignTemplates
	Get the details of a specific campaign template using the Campaign Template ID.	/CampaignTemplates ('{CampaignTemplate}')

## AssignedInterests

**Resource Path:** /AssignedInterests

You can perform the following operations on the AssignedInterests entity set:

Operations on AssignedInterests entity set

HTTP Method	Description	Path
<i>GET</i>	Get a list of interests assigned to a specific campaign template.	/ AssignedInterests (ItemOfInterest='{ItemOfInterest}', CampaignTemplate='{CampaignTemplate}')
	Get a list of assigned interests.	/AssignedInterests

## AssignedTargetGroups

**Resource Path:** /AssignedTargetGroups

You can perform the following operations on the AssignedTargetGroups entity set:

Operations on AssignedTargetGroups entity set

HTTP Method	Description	Path
<i>GET</i>	Get a list of assigned target groups.	/ AssignedTargetGroups
	Get the details of an assigned target group.	/ AssignedTargetGroups (TargetGroupUUID={TargetGroupUUID}, CampaignTemplate='{CampaignTemplate}')

## TeamMembers

**Resource Path:** /TeamMembers

You can perform the following operations on the `TeamMembers` entity set:

Operations on `TeamMembers` entity set

HTTP Method	Description	Path
<i>GET</i>	Get a specific team member assigned to a campaign template.	<code>/TeamMembers (CampaignTemplate='{CampaignTemplate}', CampaignTemplateTeamMemberName='{CampaignTemplateTeamMemberName}')</code>
	Get a list of team members.	<code>/TeamMembers</code>

## 5.5.4.2 Payload Examples

The following examples show how you can use the Campaign Templates API.

### GET Requests - Examples

#### Get the First 100 Campaign Templates

```
/sap/opu/odata/sap/API_MKT_CAMPAIGN_TEMPLATE_SRV/CampaignTemplates?$top=100&$format=json
```

#### Get the First 100 Campaign Templates Filtered by Campaign Category Type and Marketing Area

```
/sap/opu/odata/sap/API_MKT_CAMPAIGN_TEMPLATE_SRV/CampaignTemplates?$filter=MarketingArea eq 'GLOBAL' and CampaignCategoryType eq 'FB'&$top=100&$format=json
```

#### Get a Campaign Template's Assigned Target Group

```
sap/opu/odata/sap/API_MKT_CAMPAIGN_TEMPLATE_SRV/CampaignTemplates('C000000BDA')/to_AssignedTargetGroups?$format=json
```

#### Get a Campaign Template's Assigned Team Member

```
/sap/opu/odata/sap/API_MKT_CAMPAIGN_TEMPLATE_SRV/CampaignTemplates('erika.mustermann@privat.de')/to_AssignedTeamMembers?top=5&$format=json
```

#### Get a Campaign Template's Assigned Interest

```
/sap/opu/odata/sap/API_MKT_CAMPAIGN_TEMPLATE_SRV/CampaignTemplates('erika.mustermann@privat.de')/to_AssignedInterests?$$$format=json
```

## 5.5.5 Campaign Message Content and Personalized Email Content

Public OData API (`API_MKT_CAMPAIGN_MESSAGE_SRV`) for exporting and importing message content in multiple languages.

### Overview

The public API for Campaign Message Content supports the following operations for the `Marketing Engagement Business Object`:

#### i Note

This function is not available for the new editor, that is message type *Email Lite* and *Email Template Lite*.

- Creating a message. For example, create an email or a text message using the `MessageEntityType` entity.
- Exporting the message content for defined languages
  - Export messages as an HTML stream for defined languages. For example, you can use the exported `HTML Message Content` in a third-party HTML testing tool.
  - Export the `Message Block Content` (HTML body and subject) and its condition assignments for defined languages. For example, you can use the exported `HTML Message Block Content` in a third-party translation tool.
- Importing the message content for defined languages
  - Import the `Message Block Content` for defined languages.
  - Import the `Message Block Content`. For example, importing the HTML content of an email message for a new message created using the entity `MessageContentEntityType`.
  - Create the `Message Block Content` of the block and subject for new languages.
  - Update the `Message Block Content` of the block and subject for existing languages.
- Assigning a marketing agency to a message.
- Fetching marketing agencies, assigned to a message.
- Querying all installed languages using the `ValueHelps` entity set.
- Querying for all marketing areas using the `ValueHelps` entity set.
- Updating the HTML message content of an existing message. The following restrictions apply:
  - You cannot update the name of the message.
  - You cannot update a reusable block.
  - You cannot update the Confirmation (CON) content type.
  - If the SUBJECT block type is not available in the file, the SUBJECT block is removed during the update.
  - You can only update messages that have the status *In Preparation*.
  - You can update the existing message if no other language other than the default language exists.
  - You can update text messages, line messages, and mobile push notifications with a single block that only contains plain text.

## Technical Data

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/\$metadata</code> <code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/\$metadata?sap-documentation=all</code>
<b>Authorizations</b>	The following business catalog role is required: <code>SAP_CEC_BC_MKT_API_CMC_PC</code>
<b>Communication Scenario ID</b>	<code>SAP_COM_0208</code>
Component for Incidents	CEC-MKT-MEM
	<b>i Note</b> Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a> .
<b>Field Extensibility Supported</b>	No

## Support of OData Features

<b>Feature</b>	<b>Support</b>
Function Import	The Function Import <code>GetPersonalizedMessages</code> is called using REST GET and supports the following import parameters: <ul style="list-style-type: none"><li>• <code>CampaignOutbound='&lt;the outbound id&gt;'</code></li><li>• <code>LinkTrackingIsDisabled=true/false</code></li></ul>
Exception Handling	Exceptions are caught and logged on top level of the service.

## Use Case

You can use the Campaign Message Content (`API_MKT_CAMPAIGN_MESSAGE_SRV`) ODATA API for the following use cases:

- Use the API as a basis for collaborating with marketing agencies to create messages, and to map messages to agency users. An application that is built specifically for agencies or a third-party agency application can use this API and integrate this with the SAP Marketing Cloud system.
- Use the API to build a translation application for using translation as a service. The translation-as-a-service model enables translators to download messages and following translation, upload the translated content to the SAP Marketing Cloud system.
- Use the API to update the HTML message content of an existing message, if no other language other than the default language exists. You can use this functionality if an agency has uploaded an incorrect file, and the file has to be replaced instead of creating a message.

- Use this API in a customer care scenario where a customer care executive can access the personalized user content for a particular user. For example, during a call with a certain customer, the customer care executive gets access to the personalized content. For this use case, link tracking can be disabled using the parameter LINKTRACKING\_DISABLED = true.

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CAMPAIGN_MESSAGE_SRV;v =0002/\$metadata?sap- documentation=all</pre>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Campaign Message Content Details Page</a>	<p>General access to the <i>Details</i> page of the service on the SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>1. On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>2. Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Campaign Message Content API</a>	General access link takes you directly to the <i>Campaign Message Content</i> metadata file. One-time registration or logon is required.

### Note

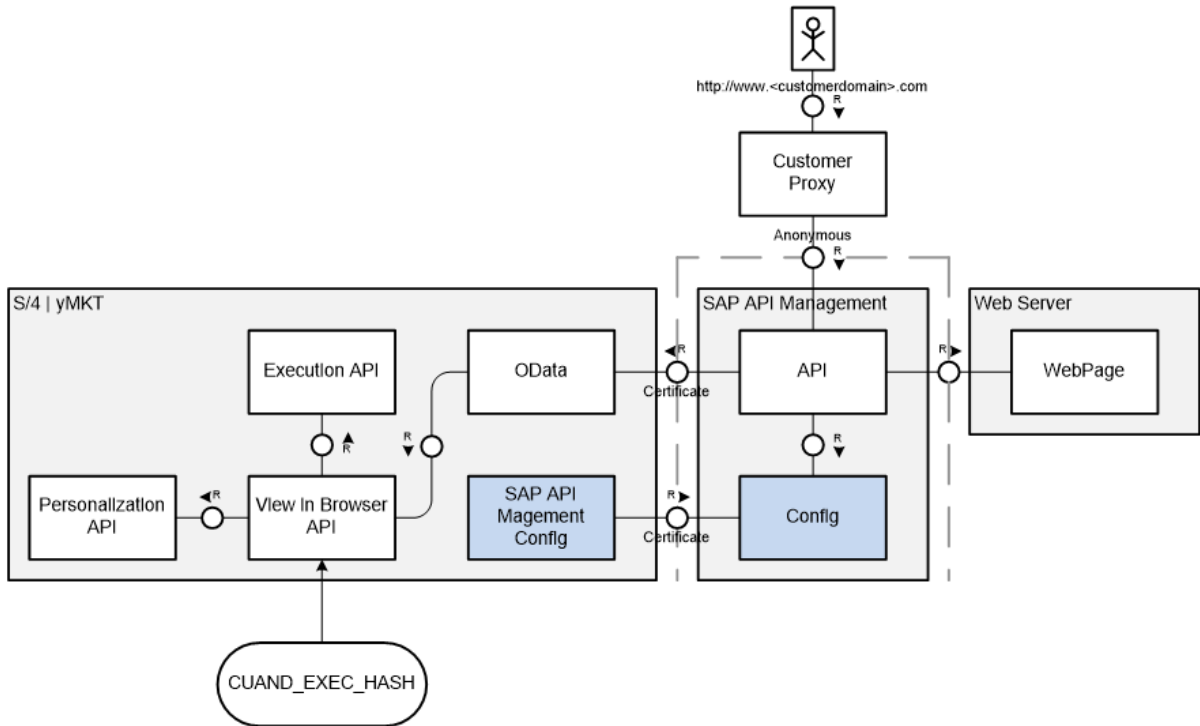
You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Entity Data Model for Personalized Email Content

The following figure shows the entity data model (EDM) for this OData Service.



## Permissions

- This feature can be enabled using the Business Role `SAP_COM_CSR_0094`.
- This feature must be assigned to a generic user for email recipients using a profile.
- The feature can be assigned to a specific user using a profile for call agents using this service.
- Add the following objects to the business role:
  - Metadata: `R3TR IWSG API_MKT_CAMPAGN_MESSAGE_SRV_0001`
  - Service: `R3TR IWSV API_MKT_CAMPAGN_MESSAGE_SRV 0001`
- Maintain the following authorization data for the business role:

Authorization Object	Authorization Field	Activity	Object Name
HPA_MKT_AR	MKTAREA_ID	Display	*
	HPA_OBJ	Display	<ul style="list-style-type: none"> <li>◦ CUAN_COMMUNICATION_CATEGORY</li> <li>◦ CUAN_MARKETING_ENGAGEMENT</li> <li>◦ CUAN_OFFER</li> <li>◦ CUAN_PRODUCT</li> <li>◦ CUAN_SENDER_PROFILE</li> <li>◦ HPA_USER</li> <li>◦ PRECO_SCENARIO</li> </ul>



Authorization Object	Authorization Field	Activity	Object Name
		<ul style="list-style-type: none"> <li>○ Change</li> <li>○ Display</li> </ul>	<ul style="list-style-type: none"> <li>○ CUAN_MARKETING_ENGAGEMENT</li> <li>○ HPA_USER</li> </ul>
HPA_ME_TMP	HPA_ME_TMP	<ul style="list-style-type: none"> <li>○ Change</li> <li>○ Display</li> </ul>	<ul style="list-style-type: none"> <li>○ Messages</li> <li>○ Template</li> </ul>

## Entity Sets

The Campaign Message Content OData API provides the following entity sets:

- [Messages \[page 798\]](#)
- [MessageContents \[page 800\]](#)
- [Blocks \[page 801\]](#)
- [BlockContents \[page 802\]](#)
- [MarketingAgencies \[page 806\]](#)
- [ValueHelps \[page 807\]](#)
- [GetPersonalizedMessage \[page 808\]](#)

Entity Set	Entity Type	Description	Path
Messages	MessageEntityType	This entity set contains all messages	/Messages
MessageContents	MessageContentEntityType	This entity set contains all message content.	/MessageContents
Blocks	BlockEntityType	This entity set contains all blocks in a message.	/Blocks
BlockContents	BlockContentEntityType	This entity set contains the contents of a block.	/BlockContents
MarketingAgencies	MarketingAgencyEntityType	The entity represents a marketing agency with the assigned campaign message.	/MarketingAgencies
ValueHelps	ValueHelpEntityType	The entity represents a generic value help.	/ValueHelps

Entity Set	Entity Type	Description	Path
GetPersonalizedMessages	GetPersonalizedMessageEntityType	This entity set returns the personalized campaign message content for a given campaign execution outbound id as text.	/GetPersonalizedMessages

You can view sample payloads and test the API at <https://api.sap.com>.

## Messages

**Resource Path:** /Messages

You can perform the following operations on the Messages entity set:

Operations on Messages entity set

HTTP Method	Description	Path
<a href="#">GET</a>	Get the list of all messages. This method supports standard OData parameters such as \$filter, \$select, \$top, and { "MessageName": "API New Message", "MessageType": "EM", "MarketingArea": "GLOBAL", "DefaultLanguage": "DE", "IsTemplate": "true" } \$skip.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/Messages?\$top=<Number of messages>
	<div style="background-color: #f0f0f0; padding: 5px;"> <p><b>i Note</b></p> <p>The \$top parameter is mandatory. You can get only 100 messages with each request.</p> </div>	
	Get a specific message using the Message UUID.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/Messages(guid`<Message UUID>`)
<a href="#">POST</a>	Create a message.	https://<Server>:<Port>/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/Messages
	<div style="background-color: #f0f0f0; padding: 5px;"> <p><b>i Note</b></p> <p>MessageType, MessageName, MarketingArea, and DefaultLanguage are the mandatory parameters for creating a message.</p> </div>	

### Sample Payload

```
{
  "MessageName": "API New Message",
  "MessageType": "EM",
  "MarketingArea": "GLOBAL",
  "DefaultLanguage": "DE",
```

```

    "IsTemplate": "true"
  }

```

The following table describes the properties for the `Messages` entity.

#### Messages Property Names and Descriptions

Property Name	Property Description
MessageUUID	Message database key.
Message	Identifier of a message in SAP Marketing Cloud.
MessageType	<p>The type of the message. Whether the message is an email or a text message.</p> <p>The possible values for the MessageType are:</p> <ul style="list-style-type: none"> <li>• EM - email (additionally, if the IsTemplate flag is set, the message is an email template)</li> <li>• LIN - LINE Message</li> <li>• LP - Landing Page</li> <li>• MPN - Mobile Push Notification</li> <li>• SMS - Text Message</li> </ul>
MessageTypeName	The description of the message type.
MessageName	The name of the message.
MessageStatus	The status of the message.
MessageStatusName	The description of the message status.
CreationDateTime	The creation date and time.
LastChangeDateTime	The last changed date and time.
CreatedByUser	The user who created the message.
CreatedByUserName	The user name of the user who created the message.
LastChangeByUser	The user who last modified the message.
LastChangeByUserName	The user name of the user who last modified the message.
DefaultLanguage	Identifier of the default language. For example: EN.
DefaultLanguageName	Description of the default language. For example: English.

Property Name	Property Description
IsTemplate	Defines if the message is a template.
	<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>i Note</b></p> <p><b>IsTemplate = true</b> is a valid combination only if <b>MessageType</b> is email.</p> </div>
MarketingArea	ID of the marketing area.
MarketingAreaName	Name of the marketing area.

## MessageContents

**Resource Path:** /MessageContents

You can perform the following operations on the MessageContents entity set:

Operations on MessageContents entity set

HTTP Method	Description	Path
<a href="#">GET</a>	Get all the message contents of a message.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/Messages(guid'<MessageUUID>')/MessageContents
	Get the specific message content of a message.	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/MessageContents(MessageUID=guid'<MessageUUID>', LanguageCode=<Language Code>)
<a href="#">POST</a>	Create message content.	https://<Server>:<Port>/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MessageContents

*Payload Sample:*

```
{
  "MessageUUID": "6c0b84b7-5523-1ed7-bcbe-2fdf35bd42b3",
  "LanguageCode": "EN",
  "LanguageName": "English",
```

```
"MessageContentHTMLString": "<!DOCTYPE html><html><head><meta charset=
\"UTF-8\"></head> <body></body></html>"
}
```

The following table describes the properties for `MessageContents` entity.

Blocks Property Names and Descriptions

Property Name	Property Description
MessageUUID	Message database key.
LanguageCode	ISO code of the language.
	<div style="border-left: 2px solid #0070C0; padding-left: 10px;"> <p><b>i Note</b></p> <p>The value of <code>LanguageCode</code> must be the same as the <code>DefaultLanguage</code> of the corresponding message (<code>MessageUUID</code>).</p> </div>
LanguageName	The name of the language.
MessageContentHTMLString	The HTML content of the message.

## Blocks

**Resource Path:** `/Blocks`

You can perform the following operations on the `Blocks` entity set:

Operations on `Blocks` entity set

HTTP Method	Description	Path
<a href="#">GET</a>	Get all blocks in a message in a specific language.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIN_MESSAGE_SRV/MessageContents(MessageUUID=guid'&lt;MessageUUID&gt;', LanguageCode=&lt;Language Code&gt;)/MessageBlocks</code>
	Get a specific block using the <code>Block</code> entity.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIN_MESSAGE_SRV/Blocks(guid'&lt;Block UUID&gt;')</code>

HTTP Method	Description	Path
<i>POST</i>	Create a block using the <code>Block</code> entity.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/\$batch https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/Blocks</code>

### i Note

This method supports creation of deep entity. You can create a deep entity using the `Block` to create a `BlockContents` entity within a `Block` entity.

The following table describes the properties for the `Blocks` entity.

Blocks Property Names and Descriptions

Property Name	Property Description
<code>BlockUUID</code>	Unique identifier of a block in SAP Marketing Cloud.
<code>ParentBlockUUID</code>	Unique identifier of a parent block in SAP Marketing Cloud.
<code>MessageUUID</code>	Unique identifier of a message in SAP Marketing Cloud.
<code>LanguageCode</code>	ISO code of the language.
<code>Block</code>	Identifier of the block.
<code>BlockType</code>	The following block types exist: TEXT, OFFER, PRODUCT, PROD_RECO, OFFER_RECO, ASC, ASC_PROD, and SUBJECT.
	<h3>i Note</h3> <p>The <code>BlockType</code> field has subblocks. The <code>BlockType=ASC</code> can have subblocks. The values for subblocks can be <code>BlockType=ASC_PROD</code> (Products) or <code>BlockType=TEXT</code> (Header or Footer). The <code>ParentBlockUUID</code> property refers to an ASC parent-block within the subblock of an ASC parent-block.</p>
<code>BlockReuseType</code>	Whether the <code>BlockType</code> is reusable. If the status is <b>R</b> , it is a reusable block with reference.
	<h3>i Note</h3> <p>The header and footer <code>BlockType</code> can be reused.</p>

## BlockContents

**Resource Path:** `/BlockContents`

You can perform the following operations on the `BlockContents` entity set:

## Operations on BlockContents entity set

HTTP Method	Description	Path
<i>GET</i>	Get all Block Contents of a Block.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIN_MESSAGE_SRV/Blocks(guid'&lt;Block UUID&gt;')/MessageBlockContents</code>
	Get the specific Block Content of a Block.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIN_MESSAGE_SRV/BlockContents(guid'&lt;Block Contents UUID&gt;')</code>
<i>POST</i>	Create block content using the Block Content entity.	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIN_MESSAGE_SRV/\$batch https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_CAMPAIN_MESSAGE_SRV/BlockContents</code>

The following table describes the properties for BlockContents entity.

### BlockContents Property Names and Descriptions

Property Name	Property Description
BlockContentUUID	Unique identifier of block content in SAP Marketing Cloud.
BlockUUID	Unique identifier of a block in SAP Marketing Cloud.
MessageUUID	Unique identifier of a message in SAP Marketing Cloud.
LanguageCode	ISO code of the language.
BlockContentConditionID	Identifier of the condition link with block content.
BlockContentConditionName	Name of the condition link with block content.
BlockContentHTMLString	Block content in the form of HTML string.
BlockContentType	Type of the block content like subject, text, and so on.
BlockPosition	Block position in a message. For example: <ul style="list-style-type: none"> <li>The position of the SUBJECT block has a fixed value of -2.</li> <li>The position of the BODY block has a fixed value of -1.</li> <li>All other blocks start with position 0.</li> </ul>
BlockContentStyleClasses	Style classes used in block content.

Property Name

Property Description

BlockControl

Block Control Identifier.

**i Note**

The defined JSON objects and their possible properties are as follows:

- If the *Block Type: ASC*, the possible property values are:

```
"BlockControl" : "{ \"SELECTION\n\": { \"LAYOUT\n\": { \"SHOW_HEADER\n\": true, \"SHOW_PRODUCTS\n\": true, \"SHOW_FOOTER\n\": true}, \"NO_MESSAGE_SEND_ON_ISSUES\n\": true}}"\n\n"BlockControlName" : ""
```

```
"BlockControl" : "{ \"SELECTION\n\": { \"LAYOUT\n\": { \"SHOW_HEADER\n\": true, \"SHOW_PRODUCTS\n\": true, \"SHOW_FOOTER\n\": false}, \"NO_MESSAGE_SEND_ON_ISSUES\n\": false}}"\n\n"BlockControlName" : ""
```

- If the *Block Type: ASC\_PROD*, the possible property values are:

```
"BlockControl" : "{ \"SELECTION\n\": { \"MAX_ITEMS\n\": 8}}"\n\n"BlockControlName" : ""
```

```
"BlockControl" : "{ \"SELECTION\n\": { \"MAX_ITEMS\n\": 9999}}"\n\n"BlockControlName" : ""
```

- If the *Block Type: OFFER*, the possible property values are:

```
"BlockControl" : "{ \"ASSIGNMENT\n\": \"static\", \"SELECTION\n\": { \"ID\n\": \"000004419\", \"CI_NAME\n\": \"\", \"COMM_MEDIUM\n\": \"EMAIL\n\", \"CONT_MEDIUM_TYPE\n\": \"01\n\", \"LANGUAGE\n\": \"EN\n\", \"NO_MESSAGE_SEND_ON_ISSUES\n\": true}}"\n\n"BlockControlName" : "4419 - Image"
```

```
"BlockControl" : "{ \"ASSIGNMENT\n\": \"static\", \"SELECTION\n\": { \"NO_MESSAGE_SEND_ON_ISSUES\n\": true}}",\n\n"BlockControlName" : ""
```



Property Name

Property Description

- If the *BlockType: OFFER\_RECO*, the possible property values are:

```
"BlockControl" : "{ \"ASSIGNMENT
\": \"offer_reco\", \"SELECTION\":
{ \"LEADING_ITEMS\": [ { \"ITEMS\":
[ { \"DB_KEY\":
\"FF91813198160B001600236CE9B411
D4\", \"NAME\": \"R-T215\" },
{ \"DB_KEY\":
\"FF93813198160B001600236CE9B411
D4\", \"NAME\": \"T-T109\" } ] },
\"ITEM_TYPE\": \"CUAN_PRODUCT
\" } ], \"LEADING_CATEGORIES\":
[ { \"ITEMS\": [ { \"DB_KEY\":
\"JMAT_ProdCatHier_API_20170512-
205027_Cat_1\", \"NAME\":
\"JMAT_ProdCatHier_API_20170512-
205027_Cat_1\" } ] }, \"ITEM_TYPE\":
\"CUAN_PRODUCT_CATEGORY\" } ] },
\"POSITION\": \"TOP\", \"CNT_TYPE
\": \"01\", \"MAX_ITEMS\":
5, \"MAX_RESULT\":
10, \"SCENARIO_ID\":
\"LD_TEST_COUPON_USAGE\",
\"TARGET\": \"\",
\"NO_MESSAGE_SEND_ON_ISSUES
\": true } }"
"BlockControlName" :
"LD_TEST_COUPON_USAGE"
```

- If the *BlockType: PRODUCT*, the possible property values are:

```
"BlockControl" : ""
"BlockControlName" : ""
```

```
"BlockControl" : "{ \"ASSIGNMENT
\": \"static\", \"SELECTION\":
{ \"PRODUCT_ID\": \"R-T129\",
\"PRODUCT_ORIGIN\":
\"SAP_ERP_MATNR\",
\"NO_MESSAGE_SEND_ON_ISSUES
\": true } }"
"BlockControlName" : "R-T129"
```

- If the *BlockType: PROD\_RECO*, the possible property values are:

```
"BlockControl" : ""
"BlockControlName" : ""
```

```
"BlockControl" : "{ \"ASSIGNMENT
\": \"prod_reco\", \"SELECTION\":
{ \"LEADING_ITEMS\": [ { \"ITEMS\":
[ ], \"ITEM_TYPE\": \"CUAN_PRODUCT
\" } ], \"MAX_ITEMS\":
7, \"MAX_RESULT\":
```

Property Name	Property Description
BlockControlName	<pre> 20,\"SCENARIO_ID\": \"SAP_MOST_VIEWED_EMAIL_CAMPAIGN \", \"TARGET\": \"\", \"NO_MESSAGE_SEND_ON_ISSUES \":false}}\" \"BlockControlName\" : \"SAP_MOST_VIEWED_EMAIL_CAMPAIGN\" </pre>
BlockControlName	Block Control Name.

## MarketingAgencies

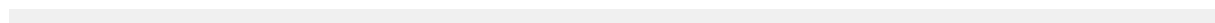
**Resource Path:** /MarketingAgencies

You can perform the following operations on the MarketingAgencyEntityType entity set:

Operations on MarketingAgencyEntityType entity set

HTTP Method	Description	Path
<i>GET</i>	Get a marketing agency to assign to a campaign message.	<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MarketingAgencies(MessageUUID=guid'6c0b84b7-5523-1ed8-8bce-01d61d137b6f',MarketingAgencyUUID=guid'6c0b84b7-5523-1ed7-bdf5-5aab5e1f5e21')">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MarketingAgencies(MessageUUID=guid'6c0b84b7-5523-1ed8-8bce-01d61d137b6f',MarketingAgencyUUID=guid'6c0b84b7-5523-1ed7-bdf5-5aab5e1f5e21')</a>
	Get all marketing agencies.	<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MarketingAgencies">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MarketingAgencies</a>
<i>POST</i>	Create a marketing agency to assign to a campaign message.	<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MarketingAgencies">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MarketingAgencies</a>
<i>DELETE</i>	Delete a marketing agency assigned to a campaign message.	<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MarketingAgencies(MessageUUID=guid'6c0b84b7-5523-1ed8-8bce-01d61d137b6f',MarketingAgencyUUID=guid'6c0b84b7-5523-1ed7-bdf5-5aab5e1f5e21')">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/sap/API_MKT_CAMPAIGN_MESSAGE_SRV/MarketingAgencies(MessageUUID=guid'6c0b84b7-5523-1ed8-8bce-01d61d137b6f',MarketingAgencyUUID=guid'6c0b84b7-5523-1ed7-bdf5-5aab5e1f5e21')</a>

*Sample Payload:*



```

{
  "MessageUUID": "6c0b84b7-5523-1ed8-8bce-01d61d137b6f",
  "MarketingAgencyUUID": "6c0b84b7-5523-1ed7-bdf5-5aab5e1f5e21",
  "MarketingAgency": "TLGG",
  "MarketingAgencyName": "TLGG"
}

```

The following table describes the properties for the `MarketingAgencyEntityType` entity.

MarketingAgencyEntityType Property Names and Descriptions

Property Name	Property Description
MessageUUID	Unique identifier of the campaign message.
MarketingAgencyUUID	Unique identifier of the agency.
MarketingAgency	Identifier of the agency.
MarketingAgencyName	Name of the agency.
EmailAddress	Email-Address of the agency.

## ValueHelps

**Resource Path:** /ValueHelps

You can perform the following operations on the `ValueHelps` entity set:

Operations on `ValueHelps` entity set

HTTP Method	Description	Path
<a href="#">GET</a>	Get all installed languages.	https://<Server>:<Port>/sap/opu/odata/sap/API_MKT_CAMPAIN_MESSAGE_SRV/ValueHelps?\$filter=ObjectType eq 'language'
	Get all marketing areas.	https://<Server>:<Port>/sap/opu/odata/sap/API_MKT_CAMPAIN_MESSAGE_SRV/ValueHelps?\$filter=ObjectType eq 'marketing_area'
	Get all marketing areas for an agency.	https://<Server>:<Port>/sap/opu/odata/sap/API_MKT_CAMPAIN_MESSAGE_SRV/ValueHelps?\$filter=ObjectType eq 'marketing_area ' and Filter/Context1 eq '6C0B84B7-5523-1EE8-8B85-084154378FB6'

The following table describes the properties for the `ValueHelps` entity.

## ValueHelps Property Names and Descriptions

Property Name	Property Description						
ObjectType	Type of value help. The possible values are either <i>MARKETING_AREA</i> or <i>LANGUAGE</i> .  <b>i Note</b> The value of valuehelps entity set is not case-sensitive.						
Code	Identifier of the value. For example, marketing area ID.						
Description	Description of the value. For example, marking area description.						
Filter	Generic filter structure of ValueHelpFilterCT type. The content is dependent on the value of ObjectType property.  <table border="1"> <thead> <tr> <th>ObjectType</th> <th>Filter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>MARKETING_AREA</td> <td>Context1</td> <td>Unique identifier of the agency.</td> </tr> </tbody> </table> <b>i Note</b> ValueHelpFilterCT is the generic type for the value help.	ObjectType	Filter	Description	MARKETING_AREA	Context1	Unique identifier of the agency.
ObjectType	Filter	Description					
MARKETING_AREA	Context1	Unique identifier of the agency.					

## GetPersonalizedMessage

### Operations

#### CRUD Operations

Do not exist for this OData service scenario.

#### Custom or Service Operations

HTTP Method	Operation Type	URI
<i>GET</i>	GetPersonalizedMessage	/GetPersonalizedMessage

### Operations

#### Request

**URI:** /API\_MKT\_CAMPAIGN\_MESSAGE\_SRV/GetPersonalizedMessage

**Operation Type:** Function Import

**HTTP Method:** *GET*

**Permissions:** SAP Business Role SAP\_COM\_CSR\_0094

## Request Parameters

Parameter	Required	Data Type	Description	Parameter Type
CampaignOutbound	Yes	Edm.String	The campaign outbound id hash key to identify the campaign, execution of the campaign and the recipient of the email.	ID
LinkTrackingIsDisabled	Yes	Edm.Boolean	If set to 'true', a link tracking interaction record is written to the database, if set to 'false', no tracking record is written.	Flag

## Request Example

```
API_MKT_CAMPAIGN_MESSAGE_SRV/GetPersonalizedMessage?
CampaignOutbound='174A024EC5EAAC77D6BF115125D284E705C90F68'&LinkTrackingIsDisabled=false
```

## Response Status and Error Codes

Code	Reason	Description
200	OK	Content found and returned
400	Bad Request	Code = CUAN_ME/812 (Page not found.) means: No content could be determined using the CampaignOutbound (Id) specified.

## Related Information

<https://api.sap.com>

### 5.5.5.1 Payload Examples

The following examples show how you can use the Campaign Message Content API.

#### GET Requests - Examples

##### Get the First 100 Messages

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/Messages?$top=100&$format=json
```

## Get the First 100 Messages Filtered by MessageType and MessageStatus

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/Messages?$filter=MessageStatus eq '10' and MessageType eq 'EM'&$top=100&$format=json
```

## Get a Message Content

```
sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/MessageContents(MessageUUID=guid'0050569F-4A52-1ED7-8481-8A95A404CF53',LanguageCode='EN')?$select=MessageUUID,LanguageCode,LanguageName,MessageContentHTMLString&$format=json
```

## Get a Message Content's Message Blocks

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/MessageContents(MessageUUID=guid'0050569F-4A52-1ED7-8481-8A95A404CF53',LanguageCode='EN')/MessageBlocks?$format=json
```

## Get a Block's Block Contents

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/Blocks(guid'0050569F-4A52-1ED7-8481-8A95A404CF53')/MessageBlockContents?&$format=json
```

## POST Requests - Examples

### Create a Message Block

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/Blocks
```

#### Sample Code

```
{
  "MessageUUID" : "0050569F-4A52-1ED7-8481-8A95A404CF53",
  "LanguageCode" : "EN",
  "BlockType" : "Offer"
}
```

### Create Block Content

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/BlockContents
```

#### Sample Code

```
{
  "BlockUUID" : "0050569F-4A52-1ED7-8481-8A95A404CF53",
  "BlockContentHTMLString" : "The Message API creates a block content for a block-2018-04-20T12:36:04.0000000",
  "BlockContentConditionName" : "cond2018-04-20T12:36:04.0000000",
  "BlockPosition" : 2,
  "BlockControl" : "DR",
  "BlockControlName" : "Divya"
}
```

## Update Requests - Examples

### Put a Block Content

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/  
BlockContents(guid'{{New_BlCnt_UUID}}')
```

#### Sample Code

```
{  
    "BlockContentHTMLString" : "PUT of Block content by DR on  
2018-04-20T12:36:04.0000000",  
    "BlockPosition" : 1,  
    "BlockControl" : "DR-PUT",  
    "BlockControlName" : "Divya-PUT"  
}
```

## OData Batch Requests - Examples

### Post Block and Block Content in a Batch

```
/sap/opu/odata/SAP/API_MKT_CAMPAIGN_MESSAGE_SRV/$batch
```

#### Sample Code

```
--batch_01869434-0005  
Content-Type: multipart/mixed; boundary=changeset_01869434-0005-0001  
--changeset_01869434-0005-0001  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
POST Blocks HTTP/1.1  
Content-Type: application/json  
Content-Length: 1021  
Content-ID: 1  
{ "MessageUUID" : "0050569F-4A52-1ED7-8481-8A95A404CF53", "LanguageCode" :  
"HE" , "BlockType" : "text" }  
--changeset_01869434-0005-0001  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
POST Blocks HTTP/1.1  
Content-Type: application/json  
Content-Length: 1021  
Content-ID: 2  
{ "MessageUUID" : "0050569F-4A52-1ED7-8481-8A95A404CF53", "LanguageCode" :  
"RU", "BlockType" : "offer" }  
--changeset_01869434-0005-0001  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
POST $1/MessageBlockContents HTTP/1.1  
Content-Type: application/json  
Content-Length: 1021
```

```

{"BlockContentType" : "agfs","BlockContentHTMLString" : "Post of content by
Batch-1!!!","BlockContentConditionName" : "new cond1"}
--changeset_01869434-0005-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST $2/MessageBlockContents HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{"BlockContentType" : "adssd","BlockContentHTMLString" : "Post of content by
Batch-2!!!","BlockContentConditionName" : "JULY13"}
--changeset_01869434-0005-0001--
--batch_01869434-0005--

```

## 5.5.6 Campaign Success Data

Public OData API (API\_MKT\_CMPGN\_SUCCESS\_IMPORT) for importing aggregated success data for Campaigns.

### Technical Data

Name of the Service	API_MKT_CMPGN_SUCCESS_IMPORT
Authorizations	The following business catalog role is required: SAP_CEC_BC_MKT_API_SUC_PC
Communication Scenario ID	SAP_COM_0390
OData Version	2.0
Root URI	https://Server:Port/sap/opu/odata/SAP/ API_MKT_CMPGN_SUCCESS_IMPORT_SRV
Service Metadata URI	https://Server:Port/sap/opu/odata/SAP/ API_MKT_CMPGN_SUCCESS_IMPORT_SRV/ \$metadata
Field Extensibility Supported	Yes

#### i Note

The SAP\_COM\_0304 communication scenario that was previously used by this API is obsolete as of SAP Marketing Cloud. For detailed information about how to create a new communication arrangement using SAP\_COM\_0390, see SAP Note [2913447](#).



## Technical Field Documentation

You can access technical documentation for the API fields at the following service metadata URL:

```
https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_CMPGN_SUCCESS_IMPORT_SRV/  
$metadata?sap-documentation=all.
```

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## 5.5.6.1 Basic Concepts

Campaign success data can provide insights that allow marketing teams to better plan and allocate campaign budget. Use the `API_MKT_CMPGN_SUCCESS_IMPORT` Public OData API to import aggregated campaign success data from external systems and write the data to campaigns in SAP Marketing Cloud.

### Processing Info

The `API_CMPGN_SUCCESS_IMPORT_SRV` Public OData service only supports POST operations.

Single requests are submitted as a single HTTP POST request to Successes endpoint. Batch requests are submitted as a single HTTP POST request to `$batch` endpoint. The batch request must contain a content-type header specifying a content type of `multipart/mixed` and a boundary specification.

If data with the following properties is already stored in SAP Marketing Cloud, a POST request containing the same properties updates the data stored in SAP Marketing Cloud:

- Campaign ID
- Ext. Campaign ID  
This also includes the following:
  - Advertiser ID
  - Managing Party
  - Ext. Cpg. System ID (Multichannel campaigns)
- Date  
This also includes the following:
  - Time Zone

- Year Month
- Year Week
- Communication Medium

## Error Messages

If the OData service is not accessible, for example due to missing authorization, or because the system is not available, a corresponding HTTP status code is returned.

If the batch request contains many sub-requests, the HTTP status code 202 is always returned. The real HTTP returning status code and messages are shown in the response body for each individual sub-request.

## Field Extensibility

In addition to the pre-delivered attributes, you can add customer-specific fields using the [Custom Fields](#) app. For more information about how to do this, see [Custom Fields](#).

### 5.5.6.2 Structure of OData Service

#### API\_MKT\_CMPGN\_SUCCESS\_IMPORT

This document describes the structure of the Public OData API service `API_MKT_CMPGN_SUCCESS_IMPORT`. Make sure you read the Basic Concepts topic before you start.

### Entity Set: Successes

This Public OData API provides the **Successes** entity set, which contains campaign success data and resides in `/Successes`.

You can perform the following operation on this entity set:

HTTP Method	Operation	URI
POST	Single import of campaign success data.	<code>/Successes</code>
	Batch import of campaign success data.	<code>/Successes/\$batch</code>

#### i Note

This API is designed for use with communication users only.

## 5.5.6.3 Payload Examples

The following examples show how you can use the Campaign Success Data API.

### POST

The following request is without batch:

#### Sample Code

```
{
  "CampaignID": "0000000001",
  "CampaignCategoryID": "EEM",
  "ExternalCampaignID": "ext1",
  "ExternalCampaignName": "ext1",
  "Advertiser": "Advertiser1",
  "AdvertiserName": "AdvertiserName1",
  "ExternalCmpgnManagingParty": "party1",
  "ExternalCmpgnManagingPartyName": "partyName1",
  "CommunicationMedium": "DISPLAY_ADS",
  "SuccessDataDate": "/Date(662725468168)/",
  "SuccessDataDateTimeZone": "UTC",
  "YearWeek": "",
  "YearMonth": "",
  "AdServingSpendAmount": "100",
  "AdServingSpendAmtCrcyISOCODE": "USD",
  "SuggestedAdServingSpendAmount": "200",
  "AgeRangeLowerLimit": 9,
  "AgeRangeUpperLimit": 10,
  "GenderFreeText": "female",
  "CountryFreeText": "usa",
  "RegionFreeText": "region",
  "InteractionStatus": "99",
  "InteractionType": "InvalidType",
  "InteractionReason": "InvalidReason",
  "CampaignContent": 98,
  "CampaignContentName": "98name",
  "CampaignContentLinkName": "97Name",
  "DeviceFreeText": "device1",
  "AdNetworkFreeText": "AdNetwork1",
  "PaidSearchKeywordText": "searchKeywordBaidu",
  "PaidSearchSearchTermText": "SearchTermBaidu",
  "CampaignReach": "96",
  "CampaignReachInPercent": "101.02",
  "NumberOfImpressions": "11",
  "NumberOfClicks": "12",
  "NumberOfUniqueClicks": "13",
  "NumberOfPageLikes": "14",
  "NumberOfPostEngagements": "15",
  "NumberOfOfferClaims": "16",
  "NumberOfVideoViews": "17",
  "NumberOfWebsiteConversions": "18",
  "NumberOfAppInstalls": "19",
  "NumberOfAppEngagements": "20",
  "NumberOfEventResponses": "21",
  "NumberOfRejectedMessages": "22",
  "NumberOfSentMessages": "23",
  "NumberOfDeliveredMessages": "24",
  "NumberOfOpenedMessages": "25",
```

```

"NumberOfHardBounces": "26",
"NumberOfSoftBounces": "27",
"NumberOfOrders": "28",
"OrderAmount": "29",
"MultiTouchAttributedOrderAmt": "29",
"OrderAmountCurrencyISOCode": "USD",
"NrOfMultiTchAttrCnvrns": "29",
"ProjectedOrderAmount": "290",
"ProjectedNumberOfConversions": "290",
"NumberOfRegistrations": "30",
"NumberOfDownloads": "31",
"VideoViewedAverageInPercent": "32",
"GrossRatingPoints": "33",
"GrossRatingPointBase": "34",
"NumberOfLeads": "35",
"NumberOfOpportunities": "36",
"OpportunityAmount": "37",
"OpportunityAmountCrcyISOCode": "USD",
"NumberOfPhoneCalls": "38",
"NumberOfAppointments": "39",
"NumberOfFailedInteractions": "40",
"NumberOfMarketingOfferViews": "41",
"NumberOfEmailComplaints": "42",
"NnbrOfOpenChannelInteractions": "43",
"NumberOfExecutedInteractions": "44",
"NumberOfTasks": "45"
}

```

## POST

### Sample Code

```

--batchtest
Content-Type: multipart/mixed; boundary=changeset_1
--changeset_1
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Successes HTTP/1.1
Content-Type: application/json
Content-Length: 168
{
  "CampaignID": "0000000004",
  "SuccessDataDate": "2017-08-03T00:00:00",
  "CommunicationMedium": "",
  "NrOfMultiTchAttrCnvrns": "21.54",
  "MultiTouchAttributedOrderAmt": "2125.34",
  "OrderAmountCurrencyISOCode": "USD"
}
--changeset_1
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Successes HTTP/1.1
Content-Type: application/json
Content-Length: 168
{
  "CampaignID": "1000000004",
  "SuccessDataDate": "2017-08-04T00:00:00",
  "CommunicationMedium": "",
  "NrOfMultiTchAttrCnvrns": "21.54",
  "MultiTouchAttributedOrderAmt": "2125.34",

```

```

"OrderAmountCurrencyISOCode": "USD"
}
--changeset_1--
--batchtest
Content-Type: multipart/mixed; boundary=changeset_9970-5898-d67d
--changeset_9970-5898-d67d
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Successes HTTP/1.1
Content-Type: application/json
Content-Length: 168
{
  "CampaignID": "0000000004",
  "SuccessDataDate": "2017-09-02T00:00:00",
  "CommunicationMedium": "",
  "NrOfMultiTchAttrCnvrns": "22.34",
  "MultiTouchAttributedOrderAmt": "2225.34",
  "OrderAmountCurrencyISOCode": "USD"
}
--changeset_9970-5898-d67d
Content-Type: application/http
Content-Transfer-Encoding: binary
POST Successes HTTP/1.1
Content-Type: application/json
Content-Length: 168
{
  "CampaignID": "0000000005",
  "SuccessDataDate": "2017-09-02T00:00:00",
  "CommunicationMedium": "",
  "NrOfMultiTchAttrCnvrns": "22.34",
  "MultiTouchAttributedOrderAmt": "2225.34",
  "OrderAmountCurrencyISOCode": "USD"
}
--changeset_9970-5898-d67d--
--batchtest--

```

## 5.5.7 Import Campaign Performance Data

You have the following options for uploading success data:

- CSV Upload - [Data File Load](#)
- Public OData API - [Campaign Success Data \[page 812\]](#)
- OData Pull Interface for Externally Executed Campaigns - [Implementing Interfaces for External Campaign Execution \[page 157\]](#)

You can also manually enter target data in the *Campaigns* app, in addition to using the CSV upload. Campaign performance data can also be automatically retrieved through standard processes in email and Facebook campaigns.

When using any of the import methods, there are a few things you need to keep in mind about your data.

## Different Drill-Down Depending on KPI

### Import Data by Campaign

Importing success data on campaign level allows you to use cross-campaign reporting on campaign-related dimensions such as Campaign ID or Marketing Area.

#### Example

Data by Campaign

Campaign ID	Impressions	Clicks
100	500	5
101	1000	10

### Import Data with Drill-Downs by One or Several Dimensions

Additional drill-downs allow for corresponding reporting, for example, over time when a success data date is given.

Importing data with a different drill-down level for various KPIs is possible, as long as you ensure that the totals remain correct.

#### Example

In an email campaign with two different links, you can divide the clicks based on which link was clicked, as you can see here:

Data with Drill-Downs by One or Several Dimensions

Campaign ID	Success Data Date	Campaign Content		Clicks
		Link Name	Impressions	
100	1.1.2016		1000	
100	1.1.2016	Learn More		9
100	1.1.2016	Unsubscribe		1
Total Campaign 100			1000	10

Drilling down this way can impact calculated measures, such as Click-Through Rate (Clicks/Impressions \* 100%). Both variables must be available on the same level to make the calculation. In the example above, there is no division of Impression on the Campaign Content Link Name level, therefore, you cannot calculate the Click-Through Rate based on the Campaign Content Link Name. However, at other drill-down levels, such as Success Data Date, you get the expected Click-Through Rate of 1%.

## Totals Must Be Correct

Most imported measures like Number of Impressions, Number of Clicks or Ad Serving Spend Amount are aggregated by summing up the values. To ensure a consistent reporting you have to import the data in a way

that the totals are correct independent from any drilldown or filtering. In the following example, you are totalling up the number of impressions across two days, using a drilldown based on gender. These data sets do not overlap each other, providing you with an accurate total number of impressions over those two days.

Correct Totals

Campaign ID	Success Data Date	Gender	Age Range	Impressions
100	1.1.2016	female		1000
100	1.1.2016	male		1000
	Subtotal 1.1.2016			2000
100	2.1.2016	female		1500
100	2.1.2016	male		1500
	Subtotal 2.1.2016			3000
Total Campaign 100				5000

This means that you cannot combine overlapping data sets like, for example, gender and age range. If you total together the number of impressions drilled down by gender and the number of impressions drilled down by age range, you will count the same impressions twice. This leads to inconsistent data, as shown in the example below:

Incorrect Totals

Campaign ID	Success Data Date	Gender	Age Range	Impressions
100	1.1.2016	female		1000
100	1.1.2016	male		1500
100	1.1.2016		18–24	500
100	1.1.2016		25–34	2000
	Subtotal 1.1.2016			<b>5000</b>
	Subtotal All Genders (correct)			2500
	Subtotal All Age Ranges (correct)			2500
Total Campaign 100				<b>5000</b>

The total in the table above is incorrect, as the number of impressions has actually been counted twice. To avoid incorrect data, you would need to organize your data with a combined drill-down, as show below:

Corrected Totals

Campaign ID	Success Data Date	Gender	Age Range	Impressions
100	1.1.2016	female	18–24	400
100	1.1.2016	female	25–34	600
100	1.1.2016	male	18–24	500
100	1.1.2016	male	25–34	1000
Subtotal 1.1.2016				<b>2500</b>
Subtotal All Genders				2500
Subtotal All Age Ranges				2500
Total Campaign 100				<b>2500</b>

## Aggregating Reach and Unique Clicks

For Reach and Unique Clicks, there are two measures that count the number of different people who saw an ad or clicked a link. You have to be careful with summing these measures up. It is rather straightforward to measure the number of impressions, but connecting those to distinct individuals is a bit more complicated. You may have the same person view an ad or click a link more than once. For that reason it's strongly recommended to import data with no drill-down by Success Data Date for Reach and Unique Clicks. Note that a drill-down by Gender or Age Range is allowed, as one person typically doesn't change age range or gender during one campaign.

You may want to be careful with cross campaign reporting on Reach and Unique Clicks, as well. Overlapping target groups may create incorrect results.

## Discrepancies in Amounts in Some Currencies

Due to the difference in the number of decimal places stored for monetary amounts in different systems, there may be a discrepancy when viewing these amounts in the different systems. In SAP Marketing Cloud, the number of decimal places in an amount depends on the currency. For example, an amount in USD is stored with two decimal places, while an amount in JPY is stored with no decimal places. However, some other systems may send amounts with more decimal places than are stored in SAP Marketing Cloud. These additional digits are cut off, which may result in minor differences when looking at totals.



## Overwriting Data by Date and Campaign ID

The semantical key of the data consists of the SAP Marketing Cloud Campaign ID, External Campaign ID (including Advertiser and Managing Party), Date, Time Unit, Communication Medium, and Input Method. To overwrite data, you must upload aggregated success data with the same success data date and external campaign ID as the data you wish to overwrite. This will overwrite all of the success data for that date and campaign ID combination.

The success data date is optional. If no success data date is given with the imported data, all other success data of the referenced external campaign ID without a success data date will be overwritten.

To delete aggregated success data, you can upload a CSV file with the success data date and external campaign ID of the data you wish to delete and leave the rest of fields blank. This will cause of the success data to be overwritten with empty fields, essentially deleting the data.

## Related Information

[Campaign Performance](#)

[Interaction and Aggregated Success Data](#)

[Custom Fields for Campaign Performance](#)

[Assigning External References to Externally Executed, Facebook, and Google Ads Campaigns](#)

### 5.5.7.1 Campaign Performance Measures and Dimensions

The list of measures and dimensions for campaign performance.

To overwrite data, you must upload aggregated success data with the same semantic key as the data you wish to overwrite. This will overwrite all of the success data for that key combination.

The semantic key for overwriting data can consist of the following:

- Campaign ID
- Ext. Campaign ID  
This also includes the following:
  - Advertiser ID
  - Managing Party
  - Ext. Cpg. System ID (Multichannel campaigns)
- Date  
This also includes the following:
  - Time Zone
  - Year Month
  - Year Week
- Communication Medium

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
CalendarYear	Calendar Year	Performance Dimension	Calendar year the campaign performance measures refer to			
CampaignContentLinkAliasName	Link Alias	Performance Dimension	Alias name of the link in the campaign content the campaign performance measures refer to	CampaignContentLinkAliasName Old CSV: CAMPAIGNCONTENTLINKALIASNAME	CampaignContentLinkAliasName	
PaidSearchKeyWordText	Baidu Key Word	Performance Dimension	The key word of a paid search campaign the campaign performance measures refer to, in standard processes used for Baidu campaigns only	PaidSearchKeyWordText	PaidSearchKeyWordText	PaidSearchKeyWordText
PaidSearchSearchTermText	Baidu Search Term	Performance Dimension	The search term of a paid search campaign the campaign performance measures refer to, in standard processes used for Baidu campaigns only	PaidSearchSearchTermText	PaidSearchSearchTermText	PaidSearchSearchTermText

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetAdServingCostPerClick	Target CPC	Target	Not shown on the campaign UI - use TgtAdSrvg-Cost-Per1000Clicks instead. Target value for ad serving cost per click, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetAdServingCostPerClick and TgtAdSrvg-SpendTransCurrency with CampaignPerformanceType="TARGET" only		
TargetAdServingCostPerLead	Target Cost per Lead	Target	Target value for ad serving cost per lead, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetAdServingCostPerLead and TgtAdSrvg-SpendTransCurrency with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetAdServingCostPerOrder	Target Order Cost	Target	Target value for ad serving cost per order, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetAdServingCostPerOrder and TgtAdSrvgSpendTransCurrency with CampaignPerformanceType="TARGET" only		
TargetAdServingCostPerRegn	Target Reg Cost	Target	Target value for ad serving cost per registration, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetAdServingCostPerRegn and TgtAdSrvgSpendTransCurrency with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetAdSrvg-Cost1000Reach	Tgt 1000 Reach Cost	Target	Target value for ad serving cost per 1000 people reached, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetAdSrvg-Cost1000Reach and TgtAdSrvg-SpendTransCurrency with CampaignPerformanceType="TARGET" only		
TargetAdSrvg-CostPerEventRsp	Tgt Event Rsp Cost	Target	Target value for ad serving cost per event response, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetAdSrvg-CostPerEventRsp and TgtAdSrvg-SpendTransCurrency with CampaignPerformanceType="TARGET" only		
TargetAdSrvg-CostPerLeadInDC	Target Cost per Lead	Converted Amount in Display Currency	Target value for the ad serving cost per lead converted to the display currency using the currency exchange rate of the campaign start date			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetAdSrvg-CostPerPage-Like	Tgt Page Like Cost	Target	Target value for ad serving cost per page like, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetAdSrvg-CostPerPage-Like and TgtAdSrvg-SpendTransCurrency with CampaignPerformanceType="TARGET" only		
TargetAdSrvg-CostPerReg-nInDC	Target Reg Cost	Converted Amount in Display Currency	Target value for the ad serving cost per registration converted to the display currency using the currency exchange rate of the campaign start date			
TargetBounceRateInPercent	Target Bounce Rate	Target	Target value for bounce rate in percent, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetBounceRateInPercent with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetClickThroughRateInPct	Target CTR	Target	Target value for click-through rate in percent, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetClickThroughRateInPct with CampaignPerformanceType="TARGET" only		
TargetClickToOpenRateInPct	Target Click Open	Target	Target value for click-to-open rate in percent, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TargetClickToOpenRateInPct with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdServing-CostPerDownload	Target Download Cost	Target	Target value for ad serving cost per download, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdServing-CostPerDownload and TgtAdSrvg-SpendTransCurrency with CampaignPerformanceType="TARGET" only		
TgtAdSrvg-Cost1000ClicksInDC	Target CPC (1000)	Converted Amount in Display Currency	Target value for the ad serving cost per 1000 clicks converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvg-Cost1000ImpsInDC	Target CPM (1000)	Converted Amount in Display Currency	Target value for the ad serving cost per 1000 impressions converted to the display currency using the currency exchange rate of the campaign start date			



Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvg-Cost1000ReachInDC	Tgt 1000 Reach Cost	Converted Amount in Display Currency	Target value for the ad serving cost per 1000 people reached converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvg-Cost1000VideoViews	Tgt Cost 1000 Views	Target	Target value for ad serving cost per 1000 video views, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvg-Cost1000VideoViews and TgtAdSrvg-SpendTransCurrency with CampaignPerformanceType="TARGET" only		
TgtAdSrvg-Cost1000VideoViewsInDC	Tgt Cost 1000 Views	Converted Amount in Display Currency	Target value for the ad serving cost per 1000 video views converted to the display currency using the currency exchange rate of the campaign start date			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvgCostAppEngmntInDC	Tgt App Engmnt Cost	Converted Amount in Display Currency	Target value for the ad serving cost per app engagement converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvgCostAppInstallInDC	Tgt App Install Cost	Converted Amount in Display Currency	Target value for the ad serving cost per app install converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvgCostDownloadInDC	Target Download Cost	Converted Amount in Display Currency	Target value for the ad serving cost per download converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvgCostEventRspInDC	Tgt Event Rsp Cost	Converted Amount in Display Currency	Target value for the ad serving cost per event response converted to the display currency using the currency exchange rate of the campaign start date			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvgCostGrossRatingPt	Target Cost per GRP	Target	Target value for ad serving cost per gross rating point, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCostGrossRatingPt and TgtAdSrvgSpendTransCurrency with CampaignPerformanceType="TARGET" only		
TgtAdSrvgCostGrssRatgPtInDC	Target Cost per GRP	Converted Amount in Display Currency	Target value for the ad serving cost per gross rating point converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvgCostMktgOfferClaim	Tgt Offer Claim Cost	Target	Target value for ad serving cost per offer claim, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCostMktgOfferClaim and TgtAdSrvgSpendTransCurrency with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvgCostOfferClaimInDC	Tgt Offer Claim Cost	Converted Amount in Display Currency	Target value for the ad serving cost per offer claim converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvgCostPer1000Clicks	Target CPC (1000)	Target	Target value for ad serving cost per 1000 clicks, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCostPer1000Clicks and TgtAdSrvgSpendTransCurrency with CampaignPerformanceType="TARGET" only		
TgtAdSrvgCostPer1000Impressions	Target CPM	Target	Target value for ad serving cost per 1000 impressions, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCostPer1000Impressions and TgtAdSrvgSpendTransCurrency with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvgCost-PerAppEngmnt	Tgt App Engmnt Cost	Target	Target value for ad serving cost per app engagement, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCost-PerAppEngmnt and TgtAdSrvg-SpendTran-sCurrency with CampaignPerformanceType="TARGET" only		
TgtAdSrvgCost-PerAppInstall	Tgt App Install Cost	Target	Target value for ad serving cost per app install, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCost-PerAppInstall and TgtAdSrvg-SpendTran-sCurrency with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvgCost-PerClickInDC	Target CPC	Converted Amount in Display Currency	Not shown on the campaign UI - use TgtAdSrvg-Cost-Per1000Clicks instead. Target value for the ad serving cost per click converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvgCost-PerOrderInDC	Target Order Cost	Converted Amount in Display Currency	Target value for the ad serving cost per order converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvgCost-PerPgLikeInDC	Tgt Page Like Cost	Converted Amount in Display Currency	Target value for the ad serving cost per page like converted to the display currency using the currency exchange rate of the campaign start date			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvgCost-PerPostEngmnt	Tgt Post Engmnt Cost	Target	Target value for ad serving cost per post engagement, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCost-PerPostEngmnt and TgtAdSrvg-SpendTran-sCurrency with CampaignPerformanceType="TARGET" only		
TgtAdSrvgCost-PerVideoView	Tgt Cost Video View	Target	Not shown on the campaign UI - use TgtAdSrvg-Cost1000VideoViews instead. Target value for ad serving cost per video view, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCost-PerVideoView and TgtAdSrvg-SpendTran-sCurrency with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvgCost-PostEngmntInDC	Tgt Post Engmnt Cost	Converted Amount in Display Currency	Target value for the ad serving cost per post engagement converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvgCost-VideoViewInDC	Tgt Cost Video View	Converted Amount in Display Currency	Not shown on the campaign UI - use TgtAdSrvg-Cost1000VideoViews instead. Target value for the ad serving cost per video view converted to the display currency using the currency exchange rate of the campaign start date			
TgtAdSrvg-Cost-WbsteCnvr-sInDC	Target Web Conv Cost	Converted Amount in Display Currency	Target value for the ad serving cost per website conversion converted to the display currency using the currency exchange rate of the campaign start date			



Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtAdSrvgCost-WebsiteCnvrnsn	Target Web Conv Cost	Target	Target value for ad serving cost per website conversion, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtAdSrvgCost-WebsiteCnvrnsn and TgtAdSrvg-SpendTran-sCurrency with CampaignPerformanceType="TARGET" only		
TgtLeadConversionRateInPct	Tgt Lead Conv. Rate	Target	Target value for lead conversion rate in percent, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtLeadConversionRateInPct with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtNumberOfConverted-Leads	Tgt Converted Leads	Target	Target value for number of converted leads, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtNumberOfConverted-Leads with CampaignPerformanceType="TARGET" only		
TgtOpenedMessageRateInPct	Target Opened Mess.	Target	Target value for opened message rate in percent, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtOpenedMessageRateInPct with CampaignPerformanceType="TARGET" only		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtUniqueClickThroughRatePct	Target Unique CTR	Target	Target value for unique click-through rate in percent, there is no meaningful aggregation possible, technically summed up when aggregated, don't set this target in combination with further dimensions such as time or gender to avoid meaningless aggregation	TgtUniqueClickThroughRatePct with CampaignPerformanceType="TARGET" only		
AdNetwork	Ad Network	Performance Dimension	Ad network code of the campaign performance measures, a mapping of external values to internal codes is maintained with the "Map Free Texts" app	AdNetwork-FreeText  Old CSV: AD-NETWORK-FREETEXT	AdNetwork	AdNetwork-FreeText
AdNetworkName	Ad Network (Description)	Language Dependent Description	Ad network name of the campaign performance measures			
AdServingCost1000ImpressionsInDC	CPM (1000)	Converted Amount in Display Currency	Ad serving cost per 1000 impressions in display currency calculated as (ad serving spend in display currency / number of impressions) *1000			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
AdServing-Cost1000PeopleReached	Cost per 1000 Reach	Calculated Actual	Ad serving cost per 1000 people reached calculated as (ad serving cost / reach) *1000			
AdServing-Cost1000VideoViewsInDC	Cost 1000 Views	Converted Amount in Display Currency	Ad serving cost per 1000 video views in display currency calculated as (ad serving spend in display currency / number of video views) *1000			
AdServingCostAppEngagementInDC	App Eng. Cost	Converted Amount in Display Currency	Ad serving cost per app engagement in display currency calculated as ad serving spend in display currency / number of app engagements			
AdServingCostGrossRatingPoint	Cost per GRP	Calculated Actual	Ad serving cost per gross rating point calculated as ad serving cost / gross rating points			
AdServingCostGrossRatingPointInDC	Cost per GRP	Converted Amount in Display Currency	Ad serving cost per gross rating point in display currency calculated as ad serving spend in display currency / gross rating points			
AdServing-Cost-Per1000Clicks	CPC (1000)	Calculated Actual	Ad serving cost per 1000 clicks calculated as (ad serving cost / number of clicks) *1000			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
AdServing-Cost-Per1000ClicksInDC	Cost per 1000 Clicks	Converted Amount in Display Currency	Ad serving cost per click in display currency calculated as (ad serving spend in display currency / number of clicks) *1000			
AdServing-Cost-Per1000Impressions	CPM	Calculated Actual	Ad serving cost per 1000 impressions calculated as (ad serving cost / number of impressions) *1000			
AdServing-Cost-Per1000VideoViews	Cost per 1000 Views	Calculated Actual	Ad serving cost per 1000 video views calculated as (ad serving cost / number of video views) *1000			
AdServingCost-PerAppEngagement	App Engagement Cost	Calculated Actual	Ad serving cost per app engagement calculated as ad serving cost / number of app engagements			
AdServingCost-PerAppInstall	Cost per App Install	Calculated Actual	Ad serving cost per app install calculated as ad serving cost / number of app installs			
AdServingCost-PerAppInstallInDC	Cost App Install	Converted Amount in Display Currency	Ad serving cost per app install in display currency calculated as ad serving spend in display currency / number of app installs			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
AdServingCost-PerClick	Cost per Click	Calculated Actual	Ad serving cost per click calculated as ad serving cost / number of clicks			
AdServingCost-PerClickInDC	Cost per Click	Converted Amount in Display Currency	Ad serving cost per click in display currency calculated as ad serving spend in display currency / number of clicks			
AdServingCost-PerDownload	Cost per Download	Calculated Actual	Ad serving cost per download calculated as ad serving cost / number of downloads			
AdServingCost-PerDownloadInDC	Cost per Download	Converted Amount in Display Currency	Ad serving cost per download in display currency calculated as ad serving spend in display currency / number of downloads			
AdServingCost-PerEventResponse	Event Response Cost	Calculated Actual	Ad serving cost per event response calculated as ad serving cost / number of event responses			
AdServingCost-PerEventResponseInDC	Event Response Cost	Converted Amount in Display Currency	Ad serving cost per event response in display currency calculated as ad serving spend in display currency / number of event responses			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
AdServingCost-PerLead	Cost per Lead	Calculated Actual	Ad serving cost per lead calculated as ad serving cost / number of leads			
AdServingCost-PerLeadInDC	Cost per Lead	Converted Amount in Display Currency	Ad serving cost per lead in display currency calculated as ad serving spend in display currency / number of leads			
AdServingCost-PerMktgOffer-Claim	Cost per Offer Claim	Calculated Actual	Ad serving cost per offer claim calculated as ad serving cost / number of offer claims			
AdServingCost-PerOfferClaimInDC	Cost Offer Claim	Converted Amount in Display Currency	Ad serving cost per offer claim in display currency calculated as ad serving spend in display currency / number of offer claims			
AdServingCost-PerOrder	Cost per Order	Calculated Actual	Ad serving cost per order calculated as ad serving cost / number of orders			
AdServingCost-PerOrderInDC	Cost per Order	Converted Amount in Display Currency	Ad serving cost per order in display currency calculated as ad serving spend in display currency / number of orders			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
AdServingCost-PerPageLike	Cost per Page Like	Calculated Actual	Ad serving cost per page like calculated as ad serving cost / number of page likes			
AdServingCost-PerPageLikeInDC	Cost Page Like	Converted Amount in Display Currency	Ad serving cost per page like in display currency calculated as ad serving spend in display currency / number of page likes			
AdServingCost-PerPostEngagement	Post Engagement Cost	Calculated Actual	Ad serving cost per post engagement calculated as ad serving cost / number of post engagements			
AdServingCost-PerPostEngmntInDC	Post Eng. Cost	Converted Amount in Display Currency	Ad serving cost per post engagement in display currency calculated as ad serving spend in display currency / number of post engagements			
AdServingCost-PerRegistration	Registration Cost	Calculated Actual	Ad serving cost per registration calculated as ad serving cost / number of registrations			
AdServingCost-PerVideoView	Cost per Video View	Calculated Actual	Ad serving cost per video view calculated as ad serving cost / number of video views			



Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
AdServingCost-PerVideoViewInDC	Cost Video View	Converted Amount in Display Currency	Ad serving cost per video view in display currency calculated as ad serving spend in display currency / number of video views			
AdServingCost-PerWebsiteConversion	Web Conversion Cost	Calculated Actual	Ad serving cost per website conversion calculated as ad serving cost / number of website conversions			
AdServingCost-RegistrationInDC	Registration Cost	Converted Amount in Display Currency	Ad serving cost per registration in display currency calculated as ad serving spend in display currency / number of registrations			
AdServingCost-WebsiteConversionInDC	Web Conv. Cost	Converted Amount in Display Currency	Ad serving cost per website conversion in display currency calculated as ad serving spend in display currency / number of website conversions			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
AdServing-SpendAmount	Ad Serving Spend	Persistent Actual	Amount spend for ad serving, summed up when aggregated	AdServing-SpendAmount and AdServing-SpendTransCurrency with CampaignPerformanceType="ACTUAL" only  Old CSV: SPEND_AMOUNT and SPEND_CURRENCY	SpendAmount and SpendCurrency	AdServing-SpendAmount and AdServing-SpendAmtCurrencyISOCODE
AdServing-SpendAmountInDC	Ad Serving Spend	Converted Amount in Display Currency	Ad serving spend amount converted to the display currency using the currency exchange rate of the campaign start date			
AdSrvg-Cost1000PeopleReachInDC	Cost per 1000 Reach	Converted Amount in Display Currency	Ad serving cost per 1000 people reached in display currency calculated as (ad serving spend in display currency / reach) *1000			
Advertiser	Advertiser ID	External Campaign Reference	Advertiser ID of the external campaign, the advertiser ID is part of the semantical key of the external campaign, not supported for campaign targets	Advertiser (with CampaignPerformanceType="ACTUAL" only)  Old CSV: ADVERTISER		Advertiser

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
Advertiser-Name	Advertiser Name	External Campaign Reference	Advertiser name of the external campaign, not supported for campaign targets	Advertiser-Name (with CampaignPerformanceType="ACTUAL" only) Old CSV: ADVERTISER-NAME		Advertiser-Name
ExternalCampaignSystemType	Ext. Cpg. System ID	External Campaign Reference	ID of the external campaign system. This ID is defined in the communication arrangement required for the <a href="#">Steps on External Platform</a> action. For more information, see <a href="#">Steps on External Platform</a> .	ExternalCampaignSystemType (with CampaignPerformanceType="ACTUAL" only)		ExternalCampaignSystemType
AgeRange	Age Range	Performance Dimension	The age range the campaign performance measures refer to	AgeRangeLowerLimit and AgeRangeUpperLimit Old CSV: AGE_RANGE_LOW and AGE_RANGE_HIGH	AgeRangeLow and AgeRangeHigh	AgeRangeLowerLimit and AgeRangeUpperLimit
AverageFrequency	Average Frequency	Calculated Actual	Average frequency of impressions calculated as number of impressions / reach			
BounceRateInPercent	Bounce Rate	Calculated Actual	Bounce rate in percent calculated as ((hard + soft bounces)/number of sent messages) *100%			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
CampaignAutomationActionUUID	Node ID	Performance Dimension	UUID of campaign automation action. Will only be used for automation tab, not for performance tab			
CampaignCategory	Campaign Category	Campaign Dimension	Category code of the campaign	CampaignCategory		CampaignCategoryID
CampaignCategoryName	Campaign Category (Description)	Language Dependent Description	Category name of the campaign			
CampaignContentLinkName	Content Link Name	Performance Dimension	Name of the link in the campaign content the campaign performance measures refer to	CampaignContentLinkName Old CSV: CAMPAIGNCONTENTLINKNAME	CampaignContentLinkName	CampaignContentLinkName
CampaignContentName	Content Name	Performance Dimension	Name of the campaign content the campaign performance measures refer to	CampaignContentName Old CSV: EXT_CONTENT_TITLE	CampaignContentName	CampaignContentName
CampaignEndDate	Campaign End Date	Campaign Dimension	End date of the campaign			
CampaignID	Campaign ID	Campaign Dimension	ID of the SAP Marketing Cloud campaign.	CampaignID Old CSV: CAMPAIGN_ID		CampaignID
CampaignLifecycleStatus	Camp. Life Cycle St.	Campaign Dimension	Life cycle status code of the campaign			
CampaignLifecycleStatusName	Camp. Life Cycle St. (Description)	Language Dependent Description	Life cycle status name of the campaign			
CampaignName	Campaign Name	Campaign Dimension	Name of the campaign			
CampaignOwner	Campaign Owner	Campaign Dimension	ID of the campaign owner			
CampaignOwnerName	Campaign Owner (Description)	Campaign Dimension	Name of the campaign owner			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
CampaignProcessType	Process Type	Campaign Dimension	Process type code of the campaign			
CampaignProcessTypeName	Process Type (Description)	Language Dependent Description	Process type name of the campaign			
CampaignReach	Reach	Persistent Actual	Reach, summed up when aggregated, this is a people centric measure - be careful to combine it with non-person related dimensions such as time when importing data as this leads to wrong aggregated values, a combination with people related dimensions such as gender is fine	CampaignReach with CampaignPerformanceType="ACTUAL"  Old CSV: UNIQUE_IMPRESSIONS	UniqueImpressions	CampaignReach
CampaignReachInPercent	Reach in Percent	Persistent Actual	Reach in percent, summed up when aggregated, this is a people centric measure - be careful to combine it with non-person related dimensions such as time when importing data as this leads to wrong aggregated values, a combination with people related dimensions such as gender is fine	CampaignReachInPercent with CampaignPerformanceType="ACTUAL"  Old CSV: CAMPAIGNREACHINPERCENT	UniqueImpressionsInPercent	CampaignReachInPercent
CampaignStartDate	Campaign Start Date	Campaign Dimension	Start date of the campaign			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
CampaignSuccessImportMethod	Import Method	Other	The code of the method the campaign performance data was imported with, filled automatically by the system if performance data is retrieved			
ClickThroughRateInPercent	CTR	Calculated Actual	Click-through rate in percent calculated as (number of clicks / (impressions + sent messages)) * 100%, depending on the campaign type typically either impressions or sent messages are given			
ClickToOpenRateInPercent	Click-To-Open Rate	Calculated Actual	Click-to-open rate in percent calculated as (number of clicks / number of opened messages) * 100%			
CmpgnPerfAltvDrillDown	Drill Down	Other	Alternative drill down of the campaign performance data used to separate independent data sets that must not be aggregated together, only used for Baidu campaigns			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
CmpgnPerformanceTimeUnit	Time Unit	Other	The time unit code the campaign performance data was imported with, filled automatically by the system if performance data is retrieved depending on the granularity of the incoming data being daily, weekly or monthly			
CmpgnPerfTimeUnitName	Time Unit (Description)	Language Dependent Description	The time unit name the campaign performance data was imported with			
CmpgnSuccessImportMethod	Import Method (Description)	Language Dependent Description	The name of the method the campaign performance data was imported with			
CommunicationMedium	Communication Medium	Performance Dimension	Communication medium code of the campaign performance measures, the communication medium used to deliver the ad, possible values can be looked up and maintained with the Manage Your Solution app	CommunicationMedium Old CSV: COMM_MEDIUM	CommunicationMedium	CommunicationMedium
CommunicationMediumName	Communication Medium (Description)	Language Dependent Description	Communication medium name of the campaign performance measures			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
CountryCode	Country/Region	Performance Dimension	Country/region code of the campaign performance measures, a mapping of external values to internal codes is maintained with the "Map Free Texts" app	CountryFree-Text  Old CSV: COUNTRY_FT	Country	CountryFree-Text
CountryName	Country/Region (Description)	Language Dependent Description	Country/region name of the campaign performance measures			
DeviceType	Device Type	Performance Dimension	Device type code of the campaign performance measures, a mapping of external values to internal codes is maintained with the "Map Free Texts" app	DeviceFreeText  Old CSV: DEVI-CEFREETEXT	DeviceType	DeviceFreeText
DeviceType-Name	Device Type (Description)	Language Dependent Description	Device type name of the campaign performance measures			
ExternalCampaignAction-Name	Ext. Cpg. Action	External Campaign Reference	Name of external campaign action used in multichannel campaigns.			
ExternalCampaignID	Ext. Campaign ID	External Campaign Reference	ID of a campaign executed on an external platform assigned to the SAP Marketing Cloud campaign, not supported for campaign targets	ExternalCampaignID (with CampaignPerformanceType="ACTUAL" only)  Old CSV: EXT_CAMPAIGN_ID	ServerCampaignId	ExternalCampaignID



Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
ExternalCampaignName	Ext. Campaign Name	External Campaign Reference	Name of the external campaign, not supported for campaign targets	ExternalCampaignName (with CampaignPerformanceType="ACTUAL" only)  Old CSV: EXT_CAMPAIGN_NAME		ExternalCampaignName
ExternalCampaignURL	Ext. Campaign URL	External Campaign Reference	Link to the campaign on the external platform			
ExternalCmpgnManagingParty	Managing Party	External Campaign Reference	Party ID of the party managing the external campaign, the party ID is part of the semantical key of the external campaign, not supported for campaign targets	ExternalCmpgnManagingParty (with CampaignPerformanceType="ACTUAL" only)  Old CSV: EXTER-NALCMPGN-MANAGING-PARTY		ExternalCmpgnManagingParty
ExternalCmpgnManagingPartyName	Managing Party Name	External Campaign Reference	Party name of the party managing the external campaign, not supported for campaign targets	ExternalCmpgnManagingPartyName (with CampaignPerformanceType="ACTUAL" only)  Old CSV: EXTER-NALCMPGN-MANAGING-PARTYNAME		ExternalCmpgnManagingPartyName

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
GenderCode	Gender	Performance Dimension	Gender code of the campaign performance measures, a mapping of external values to internal codes is maintained with the "Map Free Texts" app	GenderFreeText Old CSV: GENDER_FT	Gender	GenderFreeText
GenderCode-Name	Gender (Description)	Language Dependent Description	Gender code name of the campaign performance measures			
GrossRating-PointBase	GRP Base	Performance Dimension	Base the gross rating points measure refers to, typically contains a custom string describing the audience the gross rating points are related to, not supported for campaign targets	GrossRating-PointBase (with CampaignPerformanceType="ACTUAL" only) Old CSV: GROSSRATINGPOINT-BASE	GrossRating-PointBase	GrossRating-PointBase
GrossRating-Points	Gross Rating Points	Persistent Actual	Gross rating points, summed up when aggregated, be aware that an aggregation of gross rating points with a different gross rating point base doesn't make sense but technically is not prevented	GrossRating-Points with CampaignPerformanceType="ACTUAL" Old CSV: GROSSRATINGPOINTS	GrossRating-Points	GrossRating-Points

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
InteractionReason	Interaction Reason	Performance Dimension	The interaction reason code the campaign performance measures refer to, mainly providing failure reasons for SAP Marketing Cloud internally executed campaigns such as a failed marketing permission check, possible values can be looked up and maintained with the Manage Your Solution app, not supported for campaign targets	InteractionReason (with CampaignPerformanceType="ACTUAL" only) Old CSV: INTERACTION-REASON	InteractionReason	InteractionReason
InteractionReasonName	Interaction Reason (Description)	Language Dependent Description	The interaction reason name the campaign performance measures refer to			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
InteractionStatus	Interaction Status	Performance Dimension	<p>Status code of the interaction the campaign performance measures refer to, mainly used for measures related to business documents such as leads or sales orders to provide a status of the business document, possible values are:</p> <ul style="list-style-type: none"> <li>• 01 In Process</li> <li>• 02 Released</li> <li>• 03 Completed</li> <li>• 04 Cancelled</li> <li>• 05 Converted</li> <li>• 06 Successful</li> <li>• 07 Unsuccessful</li> <li>• 00 New</li> </ul> <p>Not supported for campaign targets</p>	<p>InteractionStatus (with CampaignPerformanceType="ACTUAL" only)</p> <p>Old CSV: INTERACTION-STATUS</p>	InteractionStatus	InteractionStatus
InteractionStatusName	Interaction Status (Description)	Language Dependent Description	Status name of the interaction the campaign performance measures refer to			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
InteractionType	Interaction Type	Performance Dimension	Type code of the interaction the campaign performance measures refer to, mainly used together with interaction reasons, possible values can be looked up and maintained with the Manage Your Solution app, not supported for campaign targets	InteractionType (with CampaignPerformanceType="ACTUAL" only) Old CSV: INTERACTION-TYPE	InteractionType	InteractionType
InteractionTypeName	Interaction Type (Description)	Language Dependent Description	Type name of the interaction the campaign performance measures refer to			
LeadConversionRateInPercent	Lead Conversion Rate	Calculated Actual	Lead conversion rate in percent calculated as (number of converted leads / number of leads) *100%			
LeadNurture	Lead Nurture ID	Campaign Dimension	Technical ID of a lead nurture using a campaign	CSV upload ignores field		
LeadNurtureName	Lead Nurture Name	Campaign Dimension	Name of a lead nurture using a campaign	CSV upload ignores field		
LeadNurtureStageUUID	Lead Nurture Stage UUID	Campaign Dimension	Guid of lead nurture stage	CSV upload ignores field		
LeadNurtureStageName	Lead Nurture Stage Name	Campaign Dimension	Name of a lead nurture stage using a campaign	CSV upload ignores field		
MarketingArea	Marketing Area	Campaign Dimension	Marketing area ID of the campaign			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
MarketingArea-Name	Marketing Area (Description)	Language Dependent Description	Marketing area name of the campaign			
MarketingPlan	Marketing Plan ID	Campaign Dimension	ID of the marketing plan associated with the campaign			
MarketingPlan-Name	Mktg Plan Name	Campaign Dimension	Name of the marketing plan associated with the campaign			
MarketingProgramID	Program	Campaign Dimension	ID of the marketing program associated with the campaign			
MarketingProgramName	Program Name	Campaign Dimension	Name of the marketing program associated with the campaign			
MediaType	Media Type ID	Performance Dimension	Media type code of the campaign performance measures, the media type is derived from the communication medium, if no communication medium is given the media type is taken from the campaign, possible values can be looked up and maintained with the Manage Your Solution app			
MediaType-Name	Media Type ID (Description)	Language Dependent Description	Media type name of the campaign performance measures			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
MultiTouchAttributedOrdAmtInDC	Multi Touch Ord Amt	Converted Amount in Display Currency	Multi touch attributed order amount converted to the display currency using the currency exchange rate of the campaign start date			
MultiTouchAttributedOrderAmt	Multi Touch Ord Amt	Persistent Actual	Order amount attributed to the campaign using multi touch attribution, summed up when aggregated, typically used with SAP Hybris Customer Attribution integration, summed up when aggregated	MultiTouchAttributedOrderAmt and OrderTransactionCurrency with CampaignPerformanceType="ACTUAL"  Old CSV: MULTITOUCHATTRIBUTEDORDERAMT and ORDERTRANSACTIONCURRENCY		MultiTouchAttributedOrderAmt and OrderAmountCurrencyISOCode
NmbrOfOpenChannelInteractions	Open Channel Interac	Persistent Actual	Number of open channel interactions, summed up when aggregated, used for campaigns executed in SAP Marketing Cloud that include custom actions implemented using the open channel	NmbrOfOpenChannelInteractions with CampaignPerformanceType="ACTUAL"  Old CSV: NMBROFOPENCHANNELINTERACTIONS		NmbrOfOpenChannelInteractions

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NrOfMultiTch-AttrCnvrnsns	Multi Touch Cnvrnsns	Persistent Actual	Number of conversions attributed to the campaign using multi touch attribution, typically used with SAP Hybris Customer Attribution integration, summed up when aggregated	NrOfMultiTch-AttrCnvrnsns with Campaign-Performance-Type= "ACTUAL" Old CSV: NROF-MULTI-TCHATTRCNVR SNS		NrOfMultiTch-AttrCnvrnsns
NumberOfAppEngagements	App Engagements	Persistent Actual	Number of app engagements as a result of the campaign, summed up when aggregated	NumberOfAppEngagements with Campaign-Performance-Type= "ACTUAL" Old CSV: APP_ENGAGEMENTS	AppEngagements	NumberOfAppEngagements
NumberOfAppInstalls	App Installs	Persistent Actual	Number of app installs attributed to the campaign, summed up when aggregated	NumberOfAppInstalls with CampaignPerformanceType= "ACTUAL" Old CSV: APP_INSTALLS	AppInstalls	NumberOfAppInstalls
NumberOfAppointments	Appointments	Persistent Actual	Number of appointments scheduled as a result of the campaign, summed up when aggregated	NumberOfAppointments with Campaign-Performance-Type= "ACTUAL" Old CSV: NUMBEROFAPPOINTMENTS	Appointments	NumberOfAppointments
NumberOfClicks	Clicks	Persistent Actual	Number of clicks, summed up when aggregated	NumberOfClicks with CampaignPerformanceType= "ACTUAL" Old CSV: CLICKS	Clicks	NumberOfClicks



Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NumberOfConvertedLeads	Converted Leads	Calculated Actual	Number of converted leads created as a result of the campaign			
NumberOfDeliveredMessages	Delivered Messages	Persistent Actual	Number of messages successfully delivered by the campaign, for campaigns executed in SAP Marketing Cloud delivered messages = sent messages - hard and soft bounces - rejected messages, summed up when aggregated	NumberOfDeliveredMessages with CampaignPerformanceType= "ACTUAL" Old CSV: DELIVERED_MESSAGES	DeliveredMessages	NumberOfDeliveredMessages
NumberOfDownloads	Downloads	Persistent Actual	Number of downloads attributed to the campaign, summed up when aggregated	NumberOfDownloads with CampaignPerformanceType= "ACTUAL" Old CSV: NUMBEROFDOWNLOADS	Downloads	NumberOfDownloads
NumberOfEmailComplaints	Email Complaints	Persistent Actual	Number of email complaints, the number of times a mail sent by the campaign was marked as spam, summed up when aggregated	NumberOfEmailComplaints with CampaignPerformanceType= "ACTUAL" Old CSV: NUMBEROFEMAILCOMPLAINTS	EmailComplaints	NumberOfEmailComplaints

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NumberOfEventResponses	Event Responses	Persistent Actual	Number of event responses, typically used with Facebook campaigns, summed up when aggregated	NumberOfEventResponses with CampaignPerformanceType= "ACTUAL" Old CSV: EVENT_RESPONSES	EventResponses	NumberOfEventResponses
NumberOfExecutedInteractions	Executed Interact.	Persistent Actual	Number of interactions executed by the SAP Marketing Cloud campaign, summed up when aggregated	NumberOfExecutedInteractions (with CampaignPerformanceType= "ACTUAL" only) Old CSV: NUMBEROFEXECUTEDINTERACTIONS	ExecutedInteractions	NumberOfExecutedInteractions
NumberOfFailedInteractions	Failed Interactions	Persistent Actual	Number of failed interactions, for campaigns executed in SAP Marketing Cloud this is the number of interactions that have been triggered but could not be executed for various reasons such as missing marketing permissions, summed up when aggregated	NumberOfFailedInteractions (with CampaignPerformanceType= "ACTUAL" only) Old CSV: NUMBEROFFAILEDINTERACTIONS	FailedInteractions	NumberOfFailedInteractions

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NumberOfHardBounces	Hard Bounces	Persistent Actual	Number of hard bounces for sent messages, summed up when aggregated	NumberOfHardBounces with CampaignPerformanceType= "ACTUAL"  Old CSV: NUMBEROFHARDBOUNCES	HardBounces	NumberOfHardBounces
NumberOfImpressions	Impressions	Persistent Actual	Number of impressions, summed up when aggregated	NumberOfImpressions with CampaignPerformanceType= "ACTUAL"  Old CSV: IMPRESSIONS	Impressions	NumberOfImpressions
NumberOfLeads	No. of Leads	Persistent Actual	Number of leads created as a result of the campaign, summed up when aggregated	NumberOfLeads with CampaignPerformanceType= "ACTUAL"  Old CSV: NUMBEROFLEADS	Leads	NumberOfLeads
NumberOfMarketingOfferViews	Offer Views	Persistent Actual	Number of offer views as a result of the campaign, summed up when aggregated	NumberOfMarketingOfferViews with CampaignPerformanceType= "ACTUAL"  Old CSV: NUMBEROFMARKETINGOFFERVIEWS	OfferViews	NumberOfMarketingOfferViews
NumberOfMarketingOfferClaims	Offer Claims	Persistent Actual	Number of offer claims as a result of the campaign, summed up when aggregated	NumberOfOfferClaims with CampaignPerformanceType= "ACTUAL"  Old CSV: OFFER_CLAIMS	OfferClaims	NumberOfOfferClaims

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NumberOfOpenedMessages	Opened Messages	Persistent Actual	Number of opened messages, summed up when aggregated	NumberOfOpenedMessages with CampaignPerformanceType= "ACTUAL"  Old CSV: NUMBEROFOPENEDMESSAGES	OpenedMessages	NumberOfOpenedMessages
NumberOfOpportunities	No. of Opportunities	Persistent Actual	Number of opportunities created as a result of the campaign, summed up when aggregated	NumberOfOpportunities with CampaignPerformanceType= "ACTUAL"  Old CSV: NUMBEROFOPPORTUNITIES	Opportunities	NumberOfOpportunities
NumberOfOrders	Number of Orders	Persistent Actual	Number of orders attributed to the campaign, summed up when aggregated	NumberOfOrders with CampaignPerformanceType= "ACTUAL"  Old CSV: NUMBEROFORDERS	Orders	NumberOfOrders
NumberOfPageLikes	Page Likes	Persistent Actual	Number of page likes as a result of the campaign, typically used for Facebook campaigns, summed up when aggregated	NumberOfPageLikes with CampaignPerformanceType= "ACTUAL"  Old CSV: PAGE_LIKES	PageLikes	NumberOfPageLikes
NumberOfPhoneCalls	Phone Calls	Persistent Actual	Number of phone calls triggered as a result of the campaign, summed up when aggregated	NumberOfPhoneCalls with CampaignPerformanceType= "ACTUAL"  Old CSV: NUMBEROFPHONECALLS	PhoneCalls	NumberOfPhoneCalls

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NumberOfPostEngagements	Page Post Eng.	Persistent Actual	Number of engagements with a page post, typically used with Facebook campaigns, summed up when aggregated	NumberOfPostEngagements with CampaignPerformanceType= "ACTUAL"  Old CSV: POST_ENGAGEMENTS	PostEngagements	NumberOfPostEngagements
NumberOfRegistrations	Registrations	Persistent Actual	Number of registrations attributed to the campaign, summed up when aggregated	NumberOfRegistrations with CampaignPerformanceType= "ACTUAL"  Old CSV: NUMBEROFREGISTRATIONS	Registrations	NumberOfRegistrations
NumberOfRejectedMessages	Rejected Messages	Persistent Actual	Number of rejected messages, for campaigns executed in SAP Marketing Cloud this is the number of messages that has been sent successfully to an external platform but that have been rejected for any reason by this platform without being counted as hard or soft bounces, summed up when aggregated	NumberOfRejectedMessages with CampaignPerformanceType= "ACTUAL"  Old CSV: REJECTED_MESSAGES	RejectedMessages	NumberOfRejectedMessages

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NumberOfSentMessages	Sent Messages	Persistent Actual	The number of messages sent by the campaign, the type of the message such as email or SMS typically is given by the communication medium, summed up when aggregated	NumberOfSentMessages with CampaignPerformanceType="ACTUAL"  Old CSV: SENT_MESSAGES	SentMessages	NumberOfSentMessages
NumberOfSoftBounces	Soft Bounces	Persistent Actual	Number of soft bounces for sent messages, summed up when aggregated	NumberOfSoftBounces with CampaignPerformanceType="ACTUAL"  Old CSV: NUMBEROFSOFTBOUNCES	SoftBounces	NumberOfSoftBounces
NumberOfTasks	Tasks	Persistent Actual	Number of tasks triggered as a result of the campaign, typically tasks are created in a connected CRM system, summed up when aggregated	NumberOfTasks with CampaignPerformanceType="ACTUAL"  Old CSV: NUMBERTASKS	Tasks	NumberOfTasks
NmbrOfTriggeredInteractions	Triggered Interact.	Calculated Actual	Number of triggered interactions calculated as executed interactions + failed interactions			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NumberOfUniqueClicks	Unique Clicks	Persistent Actual	Number of unique clicks, the number of different people that clicked an ad. For campaigns executed in SAP Marketing Cloud the number of unique clicks is per content (and not per link in the content or per campaign), summed up when aggregated, this is a people centric measure - be careful to combine it with non-person related dimensions such as time when importing data as this leads to wrong aggregated values, a combination with people related dimensions such as gender is fine	NumberOfUniqueClicks with CampaignPerformanceType="ACTUAL"  Old CSV: UNIQUE_CLICKS	UniqueClicks	NumberOfUniqueClicks
NumberOfVideoViews	Video Views	Persistent Actual	Number of video views, summed up when aggregated	NumberOfVideoViews with CampaignPerformanceType="ACTUAL"  Old CSV: VIDEO_VIEWS	VideoViews	NumberOfVideoViews

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
NumberOfWebsiteConversions	Website Conversions	Persistent Actual	Number of website conversions attributed to the campaign, typically used for Facebook campaigns, for other scenarios there are dedicated measures for the different types of conversions such as number of orders or number of registrations, summed up when aggregated	NumberOfWebsiteConversions with CampaignPerformanceType= "ACTUAL"  Old CSV: WEBSITE_CONVERSIONS	WebsiteConversions	NumberOfWebsiteConversions
OpenedMessageRateInPercent	Opened Messages in %	Calculated Actual	Rate of opened messages in percent calculated as (number of opened messages / number of delivered messages) *100%			
OpportunityAmount	Opportunity Amount	Persistent Actual	Opportunity amount attributed to the campaign, summed up when aggregated	OpportunityAmount and OpportunityTransactionCurrency with CampaignPerformanceType= "ACTUAL"  Old CSV: OPPORTUNITYAMOUNT and OPPORTUNITYTRANSACTIONCURRENCY	OpportunityAmount / OpportunityAmountCurrency	OpportunityAmount and OpportunityAmountCrcylISOCode



Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
OpportunityAmountInDC	Opportunity Amount	Converted Amount in Display Currency	Opportunity amount converted to the display currency using the currency exchange rate of the campaign start date			
OrderAmount	Order Amount	Persistent Actual	Order amount attributed to the campaign, summed up when aggregated	OrderAmount and OrderTransactionCurrency with CampaignPerformanceType="ACTUAL"  Old CSV: ORDERAMOUNT and ORDERTRANSACTIONCURRENCY	OrderAmount / OrderAmountCurrency	OrderAmount and OrderAmountCurrencyISOCode
OrderAmountInDC	Order Amount	Converted Amount in Display Currency	Order amount converted to the display currency using the currency exchange rate of the campaign start date			
ProjectedNumberOfConversions	Proj. Conversions	Persistent Actual	Projected number of conversions that could have been achieved with an optimized campaign, typically used with SAP Hybris Customer Attribution integration, summed up when aggregated	ProjectedNumberOfConversions (with CampaignPerformanceType="ACTUAL" only)  Old CSV: PROJECTEDNUMBEROFCONVERSIONS		ProjectedNumberOfConversions

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
ProjectedOrderAmount	Proj.Order Amount	Persistent Actual	Projected order amount that could have been achieved with an optimized campaign, typically used with SAP Hybris Customer Attribution integration, summed up when aggregated	ProjectedOrderAmount and OrderTransactionCurrency (with CampaignPerformanceType="ACTUAL" only) Old CSV: PROJECTEDORDERAMOUNT and ORDERTRANSACTIONCURRENCY		ProjectedOrderAmount and OrderAmountCurrencyISOCode
ProjectedOrderAmountInDC	Proj. Order Amount	Converted Amount in Display Currency	Projected order amount converted to the display currency using the currency exchange rate of the campaign start date			
Region	Region	Performance Dimension	Region code of the campaign performance measures, a mapping of external values to internal codes is maintained with the "Map Free Texts" app	RegionFreeText Old CSV: REGION_FT	Region	RegionFreeText
RegionName	Region (Description)	Language Dependent Description	Region name of the campaign performance measures			
SuccessDataDate	Date	Performance Dimension	Date the campaign performance measures refer to	CampaignPerformanceDate Old CSV: REPORTING_DATE	Date	SuccessDataDate

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
SuccessData-LastChangeDateTime	Time Last Changed	Other	Date and time when the performance data record was last updated, filled automatically by the system if performance data is retrieved			
SuccessDataReplicationStatus	Reporting Status	Other	Status code of the campaign success data replication, filled automatically by the system if performance data is retrieved			
SuccessData-DateTimeZone	Timezone	Performance Dimension	Time zone of success data date	CampaignPerformanceDateZone	TimeZone	SuccessData-DateTimeZone
SuggestedAd-ServingSpendAmount	Suggested Spend	Persistent Actual	Suggested ad serving spend amount for an optimized campaign, typically used with SAP Hybris Customer Attribution integration, summed up when aggregated	SuggestedAd-ServingSpendAmount and AdServingSpendTransCurrency with CampaignPerformanceType="ACTUAL" only  Old CSV: SUGGESTEDAD-SERVINGSPENDAMOUNT and SPEND_CURRENCY		SuggestedAd-ServingSpendAmount
SuggestedAd-ServingSpendAmtInDC	Suggested Spend	Converted Amount in Display Currency	Suggested ad serving spend amount converted to the display currency using the currency exchange rate of the campaign start date			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetCampaignReach	Target Reach	Target	Target value for the reach, summed up when aggregated, this is a people centric measure - be careful to combine it with non-person related dimensions such as time when importing data as this leads to wrong aggregated values, a combination with people related dimensions such as gender is fine	Campaign-Reach with CampaignPerformanceType="TARGET"  Old CSV: UNIQUE_IMPRESSIONS		
TargetGrossRatingPoints	Target GRPs	Target	Target value for the gross rating points, summed up when aggregated, be aware that an aggregation of gross rating points with a different gross rating point base doesn't make sense but technically is not prevented	GrossRating-Points with CampaignPerformanceType="TARGET"  Old CSV: GROSSRATINGPOINTS		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetMultiTchAttrOrderAmt	Target MTA Ord Amt	Target	Target value for the multi touch attributed order amount, summed up when aggregated	MultiTouchAttributedOrderAmt and OrderTransactionCurrency with CampaignPerformanceType="TARGET"  Old CSV: MULTITOUCHATTRIBUTEDORDERAMT and ORDERTRANSACTIONCURRENCY		
TargetNmbrOfAppEngagements	Tgt App Engagements	Target	Target value for the number of app engagements, summed up when aggregated	NumberOfAppEngagements with CampaignPerformanceType="TARGET"  Old CSV: APP_ENGAGEMENTS		
TargetNumberOfAppInstalls	Target App Installs	Target	Target value for the number of app installs, summed up when aggregated	NumberOfAppInstalls with CampaignPerformanceType="TARGET"  Old CSV: APP_INSTALLS		
TargetNumberOfAppointments	Target Appointments	Target	Target value for the number of appointments, summed up when aggregated	NumberOfAppointments with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFAPPOINTMENTS		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetNumber-OfClicks	Target Clicks	Target	Target value for the number of clicks, summed up when aggregated	NumberOfClicks with CampaignPerformanceType="TARGET"  Old CSV: CLICKS		
TargetNumber-OfDownloads	Target Downloads	Target	Target value for the number of downloads, summed up when aggregated	NumberOfDownloads with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFDOWNLOADS		
TargetNumber-OfHardBounces	Target Hard Bounces	Target	Target value for the number of hard bounces, in contrast to most other targets less is considered to be better, summed up when aggregated	NumberOfHardBounces with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFHARDBOUNCES		
TargetNumber-OfImpressions	Target Impressions	Target	Target value for the number of impressions, summed up when aggregated	NumberOfImpressions with CampaignPerformanceType="TARGET"  Old CSV: IMPRESSIONS		
TargetNumber-OfLeads	Target No. of Leads	Target	Target value for the number of leads, summed up when aggregated	NumberOfLeads with CampaignPerformanceType="TARGET" Old  CSV: NUMBEROFLEADS		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetNumber-OfOpportunities	Target Opportunities	Target	Target value for the number of opportunities, summed up when aggregated	NumberOfOpportunities with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFOPPORTUNITIES		
TargetNumber-OfOrders	Tgt Number of Orders	Target	Target value for the number of orders, summed up when aggregated	NumberOfOrders with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFORDERS		
TargetNumber-OfPageLikes	Target Page Likes	Target	Target value for the number of page likes, summed up when aggregated	NumberOfPageLikes with CampaignPerformanceType="TARGET"  Old CSV: PAGE_LIKES		
TargetNumber-OfPhoneCalls	Target Phone Calls	Target	Target value for the number of phone calls, summed up when aggregated	NumberOfPhoneCalls with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFPHONECALLS		
TargetNumber-OfRegistrations	Target Registrations	Target	Target value for the number of registrations, summed up when aggregated	NumberOfRegistrations with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFREGISTRATIONS		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetNumber-OfSentMessages	Target Sent Messages	Target	Target value for the number of sent messages, summed up when aggregated	NumberOfSentMessages with CampaignPerformanceType="TARGET"  Old CSV: SENT_MESSAGES		
TargetNumber-OfSoftBounces	Target Soft Bounces	Target	Target value for the number of soft bounces, in contrast to most other targets less is considered to be better, summed up when aggregated	NumberOfSoftBounces with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFSOFTBOUNCES		
TargetNumber-OfTasks	Target Tasks	Target	Target value for the number of tasks, summed up when aggregated	NumberOfTasks with CampaignPerformanceType="TARGET"  Old CSV: NUMBEROFTASKS		
TargetNumber-OfUniqueClicks	Target Unique Clicks	Target	Target value for the number of unique clicks, summed up when aggregated, this is a people centric measure - be careful to combine it with non-person related dimensions such as time when importing data as this leads to wrong aggregated values, a combination with people related dimensions such as gender is fine	NumberOfUniqueClicks with CampaignPerformanceType="TARGET"  Old CSV: UNIQUE_CLICKS		



Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetNumberOfVideoViews	Target Video Views	Target	Target value for the number of video views, summed up when aggregated	NumberOfVideoViews with CampaignPerformanceType="TARGET"  Old CSV: VIDEO_VIEWS		
TargetOpportunityAmount	Tgt Opportunity Amt	Target	Target value for the opportunity amount, summed up when aggregated	OpportunityAmount and OpportunityTransactionCurrency with CampaignPerformanceType="TARGET"  Old CSV: OPPORTUNITYAMOUNT and OPPORTUNITYTRANSACTIONCURRENCY		
TargetOpportunityAmountInDC	Tgt Opportunity Amt	Converted Amount in Display Currency	Target value for the opportunity amount converted to the display currency using the currency exchange rate of the campaign start date			
TargetOrderAmount	Target Order Amount	Target	Target value for the order amount, summed up when aggregated	OrderAmount and OrderTransactionCurrency with CampaignPerformanceType="TARGET"  Old CSV: ORDERAMOUNT and ORDERTRANSACTIONCURRENCY		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TargetOrderAmountInDC	Target Order Amount	Converted Amount in Display Currency	Target value for the order amount converted to the display currency using the currency exchange rate of the campaign start date			
TgtCampaignReachInPercent	Tgt Reach in Percent	Target	Target value for the reach in percent, summed up when aggregated, this is a people centric measure - be careful to combine it with non-person related dimensions such as time when importing data as this leads to wrong aggregated values, a combination with people related dimensions such as gender is fine	Campaign-ReachInPercent with Campaign-Performance-Type= "TARGET"		
TgtMultiTouchAttribOrderAmountInDC	Target MTA Order Amt	Converted Amount in Display Currency	Target value for the multi touch attributed order amount converted to the display currency using the currency exchange rate of the campaign start date		Old CSV: CAMPAIGNREACHINPERCENT	

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtNmbrOfDeliveredMessages	Tgt Delivered Msgs	Target	Target value for the number of delivered messages, summed up when aggregated	NumberOfDeliveredMessages with CampaignPerformanceType= "TARGET"  Old CSV: DELIVERED_MESSAGES		
TgtNmbrOfOpnChnlInteractions	Tgt Open Channel Int	Target	Target value for the number of open channel interactions, summed up when aggregated	NmbrOfOpenChannelInteractions with CampaignPerformanceType= "TARGET"  Old CSV: NMBROFOPENCHANNELINTERACTIONS		
TgtNmbrOfWebsiteConversions	Target Conversions	Target	Target value for the number of website conversions, summed up when aggregated	NumberOfWebsiteConversions with CampaignPerformanceType= "TARGET"  Old CSV: WEBSITE_CONVERSIONS		
TgtNrOfMultiTchAttrCnvrns	Tgt MTA Conversions	Target	Target value for the number of multi touch attributed conversions, summed up when aggregated	NrOfMultiTchAttrCnvrns with CampaignPerformanceType= "TARGET"  Old CSV: NROFMULTITCHATTRCNVRSNS		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtNumberOfEmailComplaints	Tgt Email Complaints	Target	Target value for the number of email complaints, in contrast to most other targets less is considered to be better, summed up when aggregated	NumberOfEmailComplaints with CampaignPerformanceType= "TARGET"  Old CSV: NUMBEROFEMAILCOMPLAINTS		
TgtNumberOfEventResponses	Tgt Event Responses	Target	Target value for the number of event responses, summed up when aggregated	NumberOfEventResponses with CampaignPerformanceType= "TARGET"  Old CSV: EVENT_RESPONSES		
TgtNumberOfMktgOfferClaims	Target Offer Claims	Target	Target value for the number of offer claims, summed up when aggregated	NumberOfOfferClaims with CampaignPerformanceType= "TARGET"  Old CSV: OFFER_CLAIMS		
TgtNumberOfMktgOfferViews	Target Offer Views	Target	Target value for the number of offer views, summed up when aggregated	NumberOfMarketingOfferViews with CampaignPerformanceType= "TARGET"  Old CSV: NUMBEROFMARKETINGOFFER_VIEWS		

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
TgtNumberOfOpenedMessages	Tgt Opened Messages	Target	Target value for the number of opened messages, summed up when aggregated	NumberOfOpenedMessages with Campaign-Performance-Type= "TARGET"  Old CSV: NUMBEROFOPENEDMESSAGES		
TgtNumberOfPostEngagements	Tgt Page Post Eng.	Target	Target value for the number of page post engagements, summed up when aggregated	NumberOfPostEngagements with Campaign-Performance-Type= "TARGET"  Old CSV: POST_ENGAGEMENTS		
TgtNumberOfRejectedMessages	Target Rejected Msgs	Target	Target value for the number of rejected messages, in contrast to most other targets less is considered to be better, summed up when aggregated	NumberOfRejectedMessages with Campaign-Performance-Type= "TARGET"  Old CSV: REJECTED_MESSAGES		
UniqueClickThroughRateInPct	Unique CTR	Calculated Actual	Unique click-through rate in percent calculated as (number of unique clicks / (reach + sent messages)) * 100%, depending on the campaign type typically either reach or sent messages is given			

Global Field Name	Label	Type	Description	Column in CSV Upload	Name in External Campaign OData Pull Interface	Name in OData Push Interface
VideoViewedAverageInPercent	Per. Video Viewed	Persistent Actual	Average percentage of video viewed, aggregated as average weighted by the number of video views	VideoViewedAverageInPercent with CampaignPerformanceType="ACTUAL" Old CSV: VIDEOVIEWEDAVERAGEINPERCENT	VideoViewedAverageInPercent	VideoViewedAverageInPercent
YearMonth	Year Month	Performance Dimension	Year and month the campaign performance measures refer to	YearMonth Old CSV: YEAR-MONTH	YearMonth	YearMonth
YearQuarter	Year Quarter	Performance Dimension	Year and quarter the campaign performance measures refers to			
YearWeek	Year Week	Performance Dimension	Year and week the campaign performance measures refer to	YearWeek Old CSV: YEAR-WEEK	YearWeek	YearWeek

## 5.5.7.2 Aggregated Success Data from Interactions

Some interactions are used to update aggregated success for campaigns.

### Interactions

Certain interactions are used to update the aggregated success data of campaigns. These interactions need to have a timestamp and a reference to a campaign. Therefore, the following interaction attributes must be correctly maintained:

- UTC Time Stamp in Long Form (TIMESTAMP)
- Campaign ID (INITIATIVE\_ID)

#### i Note

Aggregated success data can't be extended to include any additional measures and dimensions from interactions than those that are already used.

The interactions can have any communication medium besides *Business Document* (BUSINESS\_DOCUMENT) and *WeChat* (WEC). To see a list of business documents used to update aggregated success data, see the *Business Documents* section below.

#### Interactions Used to Update Aggregated Success

Measure	Global Field Names	Interaction Types
Delivered Messages The number of delivered messages equals the number of successfully sent messages, minus the bounces.	NumberOfDeliveredMessages	Outbound Email (EMAIL_OUTBOUND) Emails Classified as Complaint (Spam) (EMAIL_COMPLAINT) Outbound Text Message (SMS_OUTBOUND) Mobile Notification Sent (MOB_APP_NOTIF_SENT) Outbound Message from Digital Account (DIG_ACC_OUTBOUND)
Hard Bounces	NumberOfHardBounces	Hard Bounce (EMAIL_BOUNCE_HARD) Hard Bounce (SMS_BOUNCE_HARD) Hard Bounce (BOUNCE_HARD)
Soft Bounces	NumberOfSoftBounces	Soft Bounce (EMAIL_BOUNCE_SOFT) Soft Bounce (SMS_BOUNCE_SOFT)
Email Complaints	NumberOfEmailComplaints	Emails Classified as Complaint (Spam) (EMAIL_COMPLAINT)
Opened Messages	NumberOfOpenedMessages	Email Opened (EMAIL_OPENED) Mobile Notification Viewed (MOB_APP_NOTIF_VIEWED)

Measure	Global Field Names	Interaction Types
<p>Sent Messages</p> <p>The number of sent messages is the sum of interactions with one of the applicable interaction types.</p>	NumberOfSentMessages	<p>Outbound Email (EMAIL_OUTBOUND)</p> <p>Hard Bounce (EMAIL_BOUNCE_HARD)</p> <p>Soft Bounce (EMAIL_BOUNCE_SOFT)</p> <p>Emails Classified as Complaint (Spam) (EMAIL_COMPLAINT)</p> <p>Outbound Text Message (SMS_OUTBOUND)</p> <p>Hard Bounce (SMS_BOUNCE_HARD)</p> <p>Soft Bounce (SMS_BOUNCE_SOFT)</p> <p>Hard Bounce (BOUNCE_HARD)</p> <p>Mobile Notification Sent (MOB_APP_NOTIF_SENT)</p> <p>Outbound Message from Digital Account (DIG_ACC_OUTBOUND)</p> <p>Delivery Failed (DELIVERY_FAILED)</p>
Rejected Messages	NumberOfRejectedMessages	Delivery Failed (DELIVERY_FAILED)
Clicks	NumberOfClicks	Click Through (CLICK_THROUGH)
<p>Executed Interactions</p> <p>These include all interactions that aren't inbound.</p> <p>An execution run key (EXECUTION_RUN_KEY) is required.</p>	NumberOfExecutedInteractions	Any interaction type
Failed Interactions	NumberOfFailedInteractions	<p>Outbound Failed (OUTBOUND_FAILED)</p> <p>Outbound Check Failed (OUTBOUND_CHK_FAILED)</p>
<p>Unique Clicks</p> <p>Unique clicks are calculated per email or message and can't be broken down by link.</p>	NumberOfUniqueClicks	Click Through (CLICK_THROUGH)
<p>Open Channel Interactions</p> <p>The execution run key (EXECUTION_RUN_KEY) must contain a value for an Open Channel Action.</p>	NmbrOfOpenChannelInteractions	Open Channel (OPEN_CHANNEL)

For more information about the Open Channel Integration, see [Open Channel Integration \[page 194\]](#).



For all of these measures, except for *Unique Clicks*, the following dimensions are available as drilldowns in the aggregated success, if data is available:

- Communication Medium
- Interaction UTC Date
- Interaction Date in Campaign Time Zone
- Campaign Time Zone
- Campaign Content ID
- Campaign Content Name
- Campaign Automation Action UUID

For *Clicks*, the following dimensions are also available:

- Campaign Content Link Name
- Campaign Content Link Alias Name

For *Failed Interactions and Delivery Failed Messages*, the following dimensions are also available:

- Interaction Type
- Interaction Reason

For *Unique Clicks*, the following dimensions are available:

- Communication Medium
- Campaign Content ID
- Campaign Content Name
- Campaign Automation Action UUID
- Campaign Execution Run Date

## Business Documents

The measures in aggregated success listed below are updated using business documents. These business documents are handled the same as the interactions above. All of the business documents must have the communication medium *Business Document* (BUSINESS\_DOCUMENT).

Business Documents Used to Update Aggregated Success

Measure	Global Field Names	Interaction Types
Leads	NumberOfLeads	Lead (MARKETING_LEAD)
Opportunities	NumberOfOpportunities	Opportunity ('OPPORTUNITY)
Opportunity Amount	OpportunityAmount, OpportunityTransactionCurrency	Opportunity ('OPPORTUNITY)
Orders	NumberOfOrders	Sales Order (SALES_ORDER)
Order Amount	OrderAmount, OrderTransactionCurrency	Sales Order (SALES_ORDER)

Measure	Global Field Names	Interaction Types
Phone Calls	NumberOfPhoneCalls	Incoming Telephone Call (TELEPHONE_INBOUND) Outgoing Telephone Call (TELEPHONE_OUTBOUND) Unsuccessful Telephone Call (TELEPHONE_UNSUCCESSFL)
Appointments	NumberOfAppointments	Appointment (APPOINTMENT) Canceled Appointment (APPOINTMENT_CANCELLED)
Tasks	NumberOfTasks	Task (TASK)
Executed Interactions	NumberOfExecutedInteractions	Any interaction type (excluding inbound interactions)
Failed Interactions	NumberOfFailedInteractions	Outbound Failed (OUTBOUND_FAILED) Outbound Check Failed (OUTBOUND_CHK_FAILED)

For all of the business documents, the following dimensions are available as drilldowns, when data is available:

- Communication Medium
- Interaction Status
- Interaction UTC Date
- Interaction Date in Campaign Time Zone
- Campaign Time Zone
- Campaign Automation Action UUID

## 5.5.8 Survey

Set up the SAP Marketing Cloud integration with either a third-party survey tool or SAP Qualtrics Surveys via the SAP Cloud Integration. You can integrate survey metadata and survey responses from either a third-party survey tool or SAP Qualtrics Surveys using an OData service. Use the imported survey responses to create target groups and view analytics in the [Query Browser](#) app.

### Prerequisites

- You have a third-party survey tool or SAP Qualtrics Surveys.
- You have configured communication management by using the communication scenario `SAP_COM_0073`. For more information, see [Communication Management](#).
- You've created a [Survey](#) channel by performing the following steps:

1. In SAP Marketing Cloud, launch the *Manage Your Solution* app.
2. Add a new interaction channel *Survey*.
3. Assign communication medium *Web* and interaction type *Survey Response* to the *Survey* channel.
4. Choose *Save*.

## Context

As a marketer, it's important to get constant feedback from your valued customers about the product or service that you're selling on the market. Marketers initiate online surveys using third-party survey provider tools or use SAP Qualtrics Surveys to collect valuable feedback from their customers, analyze survey responses, and use this data to improve customer experience.

## Procedure

1. Create a survey using a third-party survey tool or using SAP Qualtrics Surveys.
2. Use extensibility tools in the third-party survey tool to create a custom variable and name it as **soid**.
3. Copy the generated survey URL.
4. Create an email template in SAP Marketing Cloud:
  - a. Launch the *Content Studio* app.
  - b. In the *New Content* dialog box, select *Global* from the *Marketing Area* dropdown.
  - c. Choose *Create*.
  - d. In the *Design* tab, paste the survey URL into the *Text* field of the email.
  - e. Select the *Outbound ID for Consuming App* checkbox to append a unique reference to the survey URL. The `Outbound ID` is the external ID of the interaction contact.
  - f. Release the email template.
5. Use the email template in email campaigns. For more information, see [Email and Text Message Campaigns](#).

The target group members receive the email containing the survey link. The survey responses are stored in the third-party survey tool.

6. Use the OData API service to import the survey responses into SAP Marketing Cloud. For more information, see [Survey OData API \[page 890\]](#).

By default, the *Survey* node is hidden on the Segmentation UI. For any segmentation profile, you must enable *Survey Name* from the *Segmentation Configuration* app.

7. Use the survey responses to create a target group in segmentation:
  - a. Launch the *Segmentation Models* app and choose *Create*.
  - b. In the *Profile* dropdown, select the required segmentation profile. The *All Consumers* and *All Contacts* segmentation profiles include the data of the *All Interactions* segmentation profile. The *All Interactions* segmentation profile doesn't include the data of the *All Consumers* and *All Contacts* segmentation profiles.
  - c. Drag the *Provider* attribute from *Survey* group, and choose the required survey provider from the value help.
  - d. Enter the *Segment Name* (optional) and choose *Keep*.

- e. Drag the *Name* attribute from *Survey* group and choose the required survey name from the value help.
- f. Drag the *Question* attribute from *Survey* group and choose the required survey question from the value help.
  - o The following tables list the survey question types and subtype supported in SAP Marketing Cloud: Question type 'MX' - Matrix along with the question subtype 'MU' - Menu isn't supported in SAP Marketing Cloud.

**Survey Question Types:**

<b>Question Type</b>	<b>Description</b>
RB	Radio Button
CB	Checkbox
DL	Dropdown List
FT	Free Text
MX	Matrix
OE	Open Ended
DG	Demographic
DT	Date Time
OT	Rank Order

**Survey Question Subtypes:**

<b>Question Subtype</b>	<b>Description</b>
VT	Vertical
HZ	Horizontal
MU	Menu
SL	Single
RT	Rating
RK	Ranking
ML	Multiple
NU	Numerical
ES	Essay
IN	International

Question Subtype	Description
US	United States
BO	Both
DO	Date Only
TO	Time Only
DT	Descriptive Text

- The following table lists the Qualtrics question types and subtype supported by SAP Marketing Cloud:

Qualtrics	Question Type - SAP Marketing Cloud	Subtype - SAP Marketing Cloud
MAVR (Multiple Answers, Vertical)	CB	VT
MAHR (Multiple Answers, Horizontal)	CB	HZ
SAVR (Single Answer, Vertical)	RB	VT
SAHR (Single Answer, Horizontal)	RB	HZ
DL (Drop Down)	DL	MU
SB (Select Box)	RB	VT
MSB (Multiple Select Boxes)	CB	VT
NPS	RB	HZ
TE (Text Entry)	FT	SL/ML/ES
Slider	RB	RT
RO (Rank Order)	RO	RK
Matrix	MX	SL

SAP Marketing Cloud supports only Drop Down List and Single Answer subtype questions for Matrix question type on SAP Qualtrics Surveys.

- g. Drag the *Answer* attribute from *Survey* group and choose the survey responses that you would like to analyze. You can refine the segmentation model using the following survey fields:
  - Provider
  - Question
  - Answer

- Marketing Area ID
  - Survey Status
  - Free Text
  - Responded At
- h. After completing segmentation, create a target group based on survey responses.

8. Use the target group in the *Campaigns* app to send emails.

## 5.5.8.1 Survey OData API

OData API (CUAN\_SURVEY\_IMPORT\_SRV) that supports operations on survey metadata and survey responses.

### Overview

The Survey OData API supports operations on the `Survey Business Object`.

Name of the Service	CUAN_SURVEY_IMPORT_SRV
Authorizations	This feature can be enabled with the Communication Scenario <code>SAP_COM_0073</code> .
Communication Scenario ID	<code>SAP_COM_0073</code>
Component for Incidents	<ul style="list-style-type: none"> <li>• CEC-MKT-INT-SI (Survey Integration)</li> </ul>
<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>i Note</b></p> <p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p> </div>	
OData Version	2.0
Root URI	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/</code> <code>CUAN_SURVEY_IMPORT_SRV</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/</code> <code>CUAN_SURVEY_IMPORT_SRV/\$metadata</code>
Field Extensibility Supported	Yes

You can view sample payloads and test the API at <https://api.sap.com>.

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ CUAN_SURVEY_IMPORT_SRV;v=0002/ \$metadata?sap- documentation=all</pre>	Only for internal access. You must provide the server and port names.
<a href="#">Marketing - Survey Details Page</a>	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li><li>Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Survey ODATA API</a>	General access link takes you directly to the <i>Survey</i> metadata file. One-time registration or logon is required.

### Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Entities

The Survey OData API provides the following entities:

Entity	Description	Path
SurveySet	This entity contains survey metadata.	/SurveySet
QuestionSet	This entity contains survey questions.	/QuestionSet
SurveyResponseSet	This entity contains survey responses.	/SurveyResponseSet

Entity	Description	Path
AnswerSet	This entity contains survey answers to survey questions.	/AnswerSet
ChoiceSet	This entity contains choice options to the survey questions.	/ChoiceSet
SurveyResponseDetailSet	This entity contains details of the survey responses.	/SurveyResponseDetailSet
EventSet	This entity contains details of events associated with surveys.	/EventSet

## SurveySet

**Resource Path:** /SurveySet

You can perform the following operations on the `SurveySet` entity:

Operations on SurveySet Entity

HTTP Method	Description	Path
<i>GET</i>	Get a list of surveys.	/SurveySet
	Get a specific survey.	/SurveySet (SurveyId= '<Survey ID>', Provider= '<Survey Provider>', =Version= '<Survey Version Number>')
<i>POST</i>	Create a survey.	/SurveySet

### i Note

You can either create a survey or a poll. By default, the `SurveyType` is blank and creates a survey. If you want to create a poll, enter `P` in the `SurveyType` field.

You can extend the fields of the `SurveySet` entity as follows:

1. Configure communication management. For more information, see [Communication Management](#).
2. Create a custom field. For more information, see [Custom Fields](#)
3. On the *UIs and Reports* tab, enable the *Field Usage* for *Survey Import Service*.
4. The payload of the `SurveySet` entity contains the new field.



## QuestionSet

**Resource Path:** /QuestionSet

You can perform the following operations on the QuestionSet entity:

Operations on QuestionSet Entity

HTTP Method	Description	Path
GET	Get list of questions of a survey.	/QuestionSet (SurveyId= '<Survey ID>', Provider= '<Survey Provider>', =Version= '<Survey Version Number>')
	Get a specific question of a survey.	/QuestionSet (SurveyId= '<Survey ID>', Provider= '<Survey Provider>', =Version= '<Survey Version Number>', QuestionId= '<Question ID>')
	Get the list of questions in a particular survey.	/SurveySet (SurveyId= '<Survey ID>', Provider= '<Survey Provider>', =Version= '<Survey Version Number>', ) /SurveyQuestionSet
POST	Create a survey question.	/QuestionSet

## SurveyResponseSet

**Resource Path:** /SurveyResponseSet

You can perform the following operations on the SurveyResponseSet entity:

Operations on SurveyResponseSet Entity

HTTP Method	Description	Path
GET	Get list of responses of all surveys.	/SurveyResponseSet
	Get a response of a specific survey.	/SurveyResponseSet (SurveyId= '<Survey ID>', Provider= '<Survey Provider>', =Version= '<Survey Version Number>', ResponseId= '<Response ID>')
	Get all the responses of a specific survey.	/SurveySet (SurveyId= '<Survey ID>', Provider= '<Survey Provider>', =Version= '<Survey Version Number>', ) /SurveySurveyResponseSet

HTTP Method	Description	Path
<i>POST</i>	Create a survey response.	/SurveyResponseSet

### Note

- Use extensibility tools in the third-party survey tool to create a custom variable and name it as **soid**. For more information, see [Survey \[page 886\]](#).
- While importing survey responses, the e-mail ID must be unique so that there's no inconsistency in the [CONTACT\\_KEY](#) of the survey response.
- An interaction of type [SURVEY\\_RESPONSE](#) is created for each survey response.

## AnswerSet

**Resource Path:** /AnswerSet

You can perform the following operations on the AnswerSet entity:

Operations on AnswerSet Entity

HTTP Method	Description	Path
<i>GET</i>	Get the answer of a specific question in a survey.	/QuestionSet (QuestionId= '<Question ID>', Provider= '<Survey Provider>', =Version= '<Survey Version Number>', SurveyId= '<Survey ID>') / QuestionAnswerSet
	Get the list of questions and answers for a particular survey.	/SurveySet (SurveyId= '<Survey ID>', Provider= '<Survey Provider>', =Version= '<Survey Version Number>') ?&\$expand=SurveyQuestionSet/ QuestionAnswerSet

## ChoiceSet

**Resource Path:** /ChoiceSet

You can perform the following operations on the `ChoiceSet` entity:

#### Operations on ChoiceSet Entity

HTTP Method	Description	Path
<a href="#">GET</a>	Get the choices to a specific question in a survey.	<code>/QuestionSet (QuestionId= '&lt;Question ID&gt;', Provider= '&lt;Survey Provider&gt;', =Version= '&lt;Survey Version Number&gt;', SurveyId= '&lt;Survey ID&gt;') / QuestionChoiceSet</code>
	Get the list of questions and choices for a specific survey.	<code>/SurveySet (SurveyId= '&lt;Survey ID&gt;', Provider= '&lt;Survey Provider&gt;', =Version= '&lt;Survey Version Number&gt;') ?&amp;\$expand=SurveyQuestionSet/ QuestionChoiceSet</code>
	Get the list of questions, answers, and choices for a specific survey.	<ul style="list-style-type: none"> <li><code>/SurveySet (SurveyId= '&lt;Survey ID&gt;', Provider= '&lt;Survey Provider&gt;', =Version= '&lt;Survey Version Number&gt;') / SurveyQuestionSet?sap-client=100&amp;\$expand=QuestionAnswerSet, QuestionChoiceSet</code></li> <li><code>/SurveySet (SurveyId= '&lt;Survey ID&gt;', Provider= '&lt;Survey Provider&gt;', =Version= '&lt;Survey Version Number&gt;') ?&amp;\$expand=SurveyQuestionSet/ QuestionAnswerSet, SurveyQuestionSet/ QuestionChoiceSet (with root data)</code></li> </ul>

## SurveyResponseDetailSet

You can perform the following operations on the `SurveyResponseDetailSet` entity:

**Resource Path:** `/SurveyResponseDetailSet`

#### Operations on SurveyResponseDetailSet Entity

HTTP Method	Description	Path
<a href="#">GET</a>	Get the response details of a specific survey.	<code>/SurveyResponseSet (Provider= '&lt;Survey Provider&gt;', =Version= '&lt;Survey Version Number&gt;', ResponseId= '&lt;Response ID&gt;', SurveyId= '&lt;Survey ID&gt;') / SurveyResponseSurveyResponseDetailSet</code>

## EventSet

**Resource Path:** /EventSet

You can perform the following operations on the EventSet entity:

Operations on EventSet Entity

HTTP Method	Description	Path
<i>GET</i>	Get the list of all events of all surveys.	/EventSet
	Get the list of all events for a specific survey.	/SurveySet (SurveyId= '<Survey ID>', Provider='<Survey Provider>', Version='<Survey Version Number>')/SurveyEventSet
	Get the details of a particular event for a specific survey.	/EventSet (SurveyId= '<Survey ID>', Provider='<Survey Provider>', Version='<Survey Version Number>', MktgEventUUID='<MktgEventUUID>')
<i>POST</i>	Create an event.	/EventSet
<i>PUT</i>	Update a specific event for the specified survey.	/EventSet (SurveyId= '<Survey ID>', Provider='<Survey Provider>', Version='<Survey Version Number>', MktgEventUUID='<MktgEventUUID>')
<i>DELETE</i>	Delete a specific event for the specified survey.	/EventSet (SurveyId= '<Survey ID>', Provider='<Survey Provider>', Version='<Survey Version Number>', MktgEventUUID='<MktgEventUUID>')

### 5.5.8.2 Payload Examples for Survey

The following examples show how you can use the Survey API.

#### Payload Example for Survey ROOT: POST

OData End-Point: /sap/opu/odata/SAP/CUAN\_SURVEY\_IMPORT\_SRV/SurveySet

##### Sample Code

```
{
  "SurveyId": "1234",
  "Provider": "SurveyMonkey",
  "Version": 1,
  "Name": "SurveyMonkey Demo",
  "NickName": "Demo",
  "AccountId": "1",
```

```

"Category": "customer feedback",
"Url": "www.surveymonkey.com",
"MarketingAreaId": "Global",
"Language": "E",
"IsSurveyAnonymous": false,
"CreatedOn": "2018-01-01T02:03:04",
"ModifiedOn": "2018-02-02T02:03:04",
"ValidFrom": "2018-01-01T02:03:04",
"ValidTo": "2018-12-31T02:03:04",
"IsMultipleRespAllowed": true
}

```

## Payload Example for Survey ROOT and Question: POST

OData End-Point: /sap/opu/odata/SAP/CUAN\_SURVEY\_IMPORT\_SRV/SurveySet

### Sample Code

```

{
  "SurveyId": "1234",
  "Provider": "SurveyMonkey",
  "Version": 1,
  "Name": "SurveyMonkey Demo",
  "NickName": "Demo",
  "AccountId": "1",
  "Category": "customer feedback",
  "Url": "www.surveymonkey.com",
  "MarketingAreaId": "Global",
  "Language": "E",
  "IsSurveyAnonymous": false,
  "CreatedOn": "2018-01-01T02:03:04",
  "ModifiedOn": "2018-02-02T02:03:04",
  "ValidFrom": "2018-01-01T02:03:04",
  "ValidTo": "2018-12-31T02:03:04",
  "IsMultipleRespAllowed": false,
  "SurveyQuestionSet": [
    {
      "SurveyId": "1234",
      "Provider": "SurveyMonkey",
      "Version": 1,
      "QuestionId": "Q2",
      "Language": "E",
      "PageId": 1,
      "Position": 1,
      "IsMandatory": false,
      "Type": "RB",
      "TypeName": "Single",
      "SubType": "VT",
      "Text": "Gender"
    }
  ]
}

```

## Payload Example for Survey ROOT, Question, and Answer: POST

OData End-Point: /sap/opu/odata/SAP/CUAN\_SURVEY\_IMPORT\_SRV/SurveySet

## ☰ Sample Code

```
{
  "SurveyId": "1234",
  "Provider": "SurveyMonkey",
  "Version": 1,
  "Name": "SurveyMonkey Demo",
  "NickName": "Demo",
  "AccountId": "1",
  "Category": "customer feedback",
  "Url": "www.surveymonkey.com",
  "MarketingAreaId": "Global",
  "Language": "E",
  "IsSurveyAnonymous": false,
  "CreatedOn": "2018-01-01T02:03:04",
  "ModifiedOn": "2018-02-02T02:03:04",
  "ValidFrom": "2018-01-01T02:03:04",
  "ValidTo": "2018-12-31T02:03:04",
  "IsMultipleRespAllowed": false,
  "SurveyQuestionSet": [
    {
      "SurveyId": "1234",
      "Provider": "SurveyMonkey",
      "Version": 1,
      "QuestionId": "Q2",
      "Language": "E",
      "PageId": 1,
      "Position": 1,
      "IsMandatory": false,
      "Type": "RB",
      "TypeName": "Single",
      "SubType": "VT",
      "Text": "Gender",
      "QuestionAnswerSet": [
        {
          "RowId": "A1",
          "Language": "E",
          "RowText": "Male",
          "RowPosition": 1
        },
        {
          "RowId": "A2",
          "Language": "E",
          "RowText": "Female",
          "RowPosition": 2
        }
      ]
    }
  ]
}
```

## Payload Example for Survey Metadata (\$batch): POST

OData End-Point: /sap/opu/odata/SAP/CUAN\_SURVEY\_IMPORT\_SRV/\$batch

## ☰ Sample Code

```
--batch_01869434-0005
Content-Type: multipart/mixed; boundary=changeset_01869434-0005-0001
--changeset_01869434-0005-0001
Content-Type: application/http
```

```

Content-Transfer-Encoding: binary
POST SurveySet HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{"SurveyId":"1234","Provider":"SurveyMonkey","Version":
1,"ValidFrom":"2018-01-03T02:03:04","ValidTo":"2018-12-31T02:03:04","CreatedOn
":"2018-01-02T02:03:04","ModifiedOn":"2018-02-02T02:03:04","SurveyQuestionSet{
"SurveyId":"1234","Provider":"SurveyMonkey","Version":
1,"QuestionId":"Q1","Language":"E","Text":"Gender","Type":"RB","QuestionAnswer
Set":[{"RowId":"A1","RowText":"Male","Language":"E","RowPosition":1},
{"RowId":"A2","RowText":"Female","Language":"E","RowPosition":2}]}]}
--changeset_01869434-0005-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST SurveySet HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{"SurveyId":"12345","Provider":"SurveyMonkey","Version":
1,"ValidFrom":"2018-01-03T02:03:04","ValidTo":"2018-12-31T02:03:04","CreatedOn
":"2018-01-02T02:03:04","ModifiedOn":"2018-02-02T02:03:04"}
--changeset_01869434-0005-0001--
--batch_01869434-0005--

```

## Payload Example for Survey Response: POST

OData End-Point: /sap/ops/odata/SAP/CUAN\_SURVEY\_IMPORT\_SRV/SurveyResponseSet

### Sample Code

```

{
  "SurveyId": "1234",
  "Provider": "SurveyMonkey",
  "Version": 1,
  "ResponseId": "11",
  "Id": "response@sap.com",
  "IdOrigin": "EMAIL",
  "IsResponseAnonymous": false,
  "ResponseUrl": " www.surveymonkey.com ",
  "RespondedOn": "2018-03-08T02:03:04",
  "SurveyResponseSurveyResponseDetailSet": [
    {
      "QuestionId": "Q1",
      "ResponseIdRow": "A1"
    },
    {
      "QuestionId": "Q2",
      "ResponseIdRow": "R1",
      "ResponseIdCol": "C2"
    }
  ],
  "SurveyResponseContactSet":
  {
    "SurveyId": "1234",
    "Provider": "SurveyMonkey",
    "Version": 1,
    "NameFirst": "Suresh",
    "NameLast": "K",
    "EmailAddr": "suresh.r.kai@sap.com",
    "TelephoneNo": "9008122077"
  }
}

```

## Payload Example for Survey Response (\$batch): POST

OData End-Point: /sap/opu/odata/SAP/CUAN\_SURVEY\_IMPORT\_SRV/\$batch

### Sample Code

```
--batch_01869434-0005
Content-Type: multipart/mixed; boundary=changeset_01869434-0005-0001
--changeset_01869434-0005-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST SurveyResponseSet HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{
  "Provider": "SurveyMonkey",
  "ResponseId": "11",
  "SurveyId": "1234",
  "Version": 1,
  "Id": "response1r11@sap.com",
  "IdOrigin": "EMAIL",
  "IsResponseAnonymous": false,
  "ResponseUrl": "www.surveymonkey.com",
  "RespondedOn": "2018-02-04T02:03:04",
  "SurveyResponseSurveyResponseDetailSet": [
    {
      "QuestionId": "Q1",
      "ResponseIdRow": "A1"
    },
    {
      "QuestionId": "Q2",
      "ResponseIdRow": "R1",
      "ResponseIdCol": "C2"
    }
  ]
}
--changeset_01869434-0005-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST SurveyResponseSet HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{
  "Provider": "SurveyMonkey",
  "ResponseId": "12",
  "SurveyId": "1234",
  "Version": 1,
  "Id": "response1r12@sap.com",
  "IdOrigin": "EMAIL",
  "IsResponseAnonymous": false,
  "ResponseUrl": "www.surveymonkey.com",
  "RespondedOn": "2018-02-04T02:03:04",
  "SurveyResponseSurveyResponseDetailSet": [
    {
      "QuestionId": "Q1",
      "ResponseIdRow": "A1"
    },
    {
      "QuestionId": "Q2",
      "ResponseIdRow": "R1",
      "ResponseIdCol": "C2"
    }
  ]
}
```



```

    ]
  }
}
--changeset_01869434-0005-0001--
--batch_01869434-0005--

```

## Payload Example for an Event Creation: POST

OData End-Point: /sap/opu/odata/SAP/CUAN\_SURVEY\_IMPORT\_SRV/EventSet

### Sample Code

```

{
  "SurveyId": "SRV_EVT1_0002",
  "Provider": "Qualtrics",
  "Version": 1,
  "MktgEventExternalID": "5482759",
  "MktgEventUUID": "6C0B84B7-5523-1EE9-B2B0-5DAC5A630B55",
  "MktgEventProvider": "ON24_ID",
  "MktgEventProviderAccount": "23192"
}

```

## 5.5.9 Read Content of Export Files in Campaigns

Public OData API for Export Definition. An export definition is a template for structuring the export of target group member data, included in a target group or a campaign, to CSV files.

### Overview

The public API for Export Definition supports operations on the `Export Definition Business Object`.

Name of the Service	API_MKT_EXPORT_DEFINITION
Underlying BO	BO_HPA_EXPORT_DEFINITION Read of BO_CUAN_INITIATIVE and CUAN_MARKETING_ORCHESTRATION
Package	CUAN_ODATA_API_EXPORT_DEF

## Technical Data

Technical Data of Service

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_EXPORT_DEFINITION_SRV</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_EXPORT_DEFINITION_SRV/\$metadata</code>
<b>Communication Scenario ID</b>	SAP_COM_0311
<b>Component for Incidents</b>	CEC-MKT-EXP

### i Note

Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_EXPORT_DEFINITION_SRV/\$metadata?sap-documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Read Content of Export Files in Campaigns</a>	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li><li>Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Read Content of Export Files in Campaigns API</a>	General access link takes you directly to the <i>Read Content of Export Files in Campaigns</i> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Entity Sets

The Export Definition OData API provides the following entity sets:

Entity Set	Description	Path
ObjectFiles	This entity contains data for an export file created during campaign execution.	/ObjectFiles

**Resource Path:** /ObjectFiles

You can perform the following operations on the /ObjectFiles entity set:

Operations on ObjectFiles entity set

HTTP Method	Description	Path
<i>GET</i>	<p>Get a list of <code>Export File Names</code> for a campaign ID.</p> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>Provide property <code>CampaignID</code></li> <li>Provide property <code>FileName</code> (space)</li> <li>Use property <code>DateFrom</code> as filter to get files not older than the specified date/time</li> </ul>	<ul style="list-style-type: none"> <li>/ObjectFiles(CampaignID='nnnnnnnnnn',FileName='')</li> <li>/ObjectFiles?\$filter=CampaignID eq 'nnnnnnnnnn' and DateFrom gt datetime 'yyyy-mm-ttThh:mm:ss'</li> </ul>
<i>GET (STREAM)</i>	<p>Get the stream of an <code>Export File Content</code> in Xstring format for the specified properties <code>FileName</code> and <code>CampaignID</code>.</p> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>Provide properties <code>CampaignID</code> and <code>FileName</code> (from result of first GET)</li> <li>The <code>\$value</code> parameter is mandatory.</li> </ul>	<ul style="list-style-type: none"> <li>/ObjectFiles(CampaignID='nnnnnnnnnn',FileName='mmmmmmmmmmmmmmmmmmmm')/\$value</li> </ul>

You can view sample payloads and test the API at <https://api.sap.com>.

## 5.5.10 Marketing Events

This section about marketing events gives you all the details about the public OData API (`API_MKT_EVENT_SRV`).

### [Marketing Events OData API \[page 904\]](#)

Public OData API (`API_MKT_EVENT_SRV`) for importing events data from third-party event provider platforms.

### [Payload Examples for Marketing Events \[page 917\]](#)

The following examples show how you can use the marketing events API:

### [Function Imports \[page 922\]](#)

Function imports are used to perform custom operations on an entity, in addition to typical OData operations. This section also provides payload examples.

### 5.5.10.1 Marketing Events OData API

Public OData API (`API_MKT_EVENT_SRV`) for importing events data from third-party event provider platforms.

#### Overview

With the OData Service `API_MKT_EVENT_SRV`, you can import events, participants, and participants Q&A from third-party event provider platforms such as ON24, Zoom, and so on.

Participants are imported as contacts and are used for event promotions and follow-up marketing activities in SAP Marketing Cloud.

OData Version	2.0
Root URI	<code>https://Server:Port/sap/opu/odata/SAP/API_MKT_EVENT_SRV</code>
Service Metadata URI	<code>https://Server:Port/sap/opu/odata/SAP/API_MKT_EVENT_SRV/\$metadata</code>
Authorizations	Business Catalog: <code>SAP_CEC_BC_MKT_API_EVENT_PC</code>
Communication Scenario ID	<code>SAP_COM_CSR_0371</code>
Component for Incidents	<ul style="list-style-type: none"><li>CEC-MKT-EVT</li></ul>
Field Extensibility Supported	Yes

#### i Note



Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

## → Recommendation

APIs do not support parallel calls for a single marketing event. SAP recommends that you make API calls in a sequence for a given marketing event.

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/API_MKT_EVENT_SRV/ \$metadata?sap- documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Marketing Events Page</a> 	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"><li>1. On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li><li>2. Specify which application you want to use to open the EDMX file type.</li></ol>
<a href="#">Marketing Events API</a> 	General access link takes you directly to the <i>Marketing Events</i> metadata file. One-time registration or logon is required.

## Support of OData Features

Feature	Support
Query options for value help entities	The current implementation of the value help entities supports the following query options, which can be either passed as a query or path parameters: <ul style="list-style-type: none"><li>• \$top and \$skip</li><li>• \$select</li><li>• \$orderby</li><li>• \$count and \$inlinecount</li></ul>
Bulk processing using deep-create	The service supports bulk processing using deep-create.
Batch processing of multiple service call	Multiple services like import events, participants, and participant Q&A can be called together.
Contact check	While importing a participant as contact, checks if a contact with same ID and origin already exists in SAP Marketing system.

## Entity Data Model

**Service Metadata URI:** `https://Server:Port/sap/opu/odata/SAP/API_MKT_EVENT_SRV/$metadata`

### MarketingEvents

**Resource Path:** `/MarketingEvents`

You can perform the following operations on the `MarketingEvents` resource:

Operations on the `MarketingEvents` resource

HTTP Method	Description	URI
<i>GET</i>	Getting all marketing events	<code>/MarketingEvents</code>
<i>GET</i>	Getting a single marketing event	<code>/MarketingEvents ('&lt;MktgEventUUID&gt;')</code>
<i>POST</i>	Deep-create or create a single program	<code>/MarketingEvents</code>
<i>PUT</i>	Updating a single marketing event	<code>/MarketingEvents ('&lt;MktgEventUUID&gt;')</code>

### Properties

Parameter	Description	Creatable	Updatable	Key	Mandatory
<code>MktgEventUUID</code>	Unique identifier generated in SAP Marketing Cloud system for every marketing event				
<code>MktgEvent</code>	Identifier of the marketing event				
<code>MktgObjVersHdrUUID</code>	Unique identifier generated in SAP Marketing Cloud system for version of the marketing object				
<code>MktgEventExternalId</code>	External identifier of the imported marketing event	X			X (except for event with the status "In Preparation")

Parameter	Description	Creatable	Updatable	Key	Mandatory
MktgEventProvider	Name of the marketing event provider	X	X		X
	<p><b>i Note</b></p> <p>This field is validated with the ID Origin of contact maintained in the SAP Marketing Cloud system.</p>				
MktgEventProviderAccount	Provider account of the marketing event	X	X		
MktgEventName	Name of the marketing event	X	X		X
MktgEventDescription	Description of the marketing event	X	X		
MktgEventStatus	Identifier of the marketing event status. This is a pre-delivered and configurable field.	X	X		X
MktgEventStatusName	Name of the marketing event status				
MediaType	Identifier of the media type. This is a pre-delivered and configurable field.	X	X		
MediaTypeName	Name of the media type				
MktgEventEndDateTime	End date and time (in UTC) of the marketing event	X	X (except for events with status "Conducted" and "Cancelled")		X

Parameter	Description	Creatable	Updatable	Key	Mandatory
MktgEventStart DateTime	Start date and time (in UTC) of the marketing event	X	X (except for events with status "Conducted" and "Cancelled")		X

**i Note**

- Value of event start date should be lesser than or equal to the value of event end date.
- If the event start date is in the future, then you cannot set the event status to "Conducted".



Parameter	Description	Creatable	Updatable	Key	Mandatory
MktgEvtReplayAvailFromDateTime	Date and time (in UTC) from when the on-demand recording will be available.	X	X		
	<p><b>i Note</b></p> <p>Value of on-demand recording available from date (MKTGEVTREPLAYAVAILFROMDATETIME) should be lesser than or equal to the value of on-demand recording available until date (MKTGEVTREPLAYAVAILTODATETIME).</p>				
MktgEvtReplayAvailToDateTime	Date and time (in UTC) until when the on-demand recording will be available	X	X		
MktgEventTimeZone	Time zone of the marketing event. This is a pre-delivered and configurable field.	X	X		
MarketingArea	Identifier of the marketing area	X	X		
MarketingAreaName	Name of the marketing area name				
Language	Language key	X	X		

Parameter	Description	Creatable	Updatable	Key	Mandatory
MktgEventInfoURL	URL of the marketing event	X	X		
MktgEventRegistrationURL	Registration URL of the marketing event	X	X		
MktgEventOnlineURL	Live URL of the marketing event	X	X		
MktgEventReplyURL	On-demand URL of the marketing event	X	X		
MktgEventType	Type of the marketing event. This is a pre-delivered field.	X			
MktgEventTypeName	Type name of the marketing event				
CreatedByUser	Name of the user who created the marketing event				
CreationDateTime	Date and time when the marketing event was created				
LastChangedByUser	Name of the user who last changed the marketing event				
LastChangeDateTime	Date and time when the marketing event was last changed				

The following event status values are available:

- 0001 for In Preparation
- 0002 for Ready
- 0003 for Cancelled
- 0004 for Conducted
- 0005 for Archived

The following event type values are available:

- \_ (read as blank) for Online
- 10 for In Person

The following event status transitions are possible:

Source Status ID	Target Status ID
0001	0002
0001	0003
0001	0004
0001	0005
0002	0003
0002	0004
0002	0005
0003	0005
0004	0005

## EventParticipants

**Resource Path:** /EventParticipants

You can perform the following operations on the `EventParticipants` resource:

Operations on the `EventParticipants` resource

HTTP Method	Description	URI
<i>GET</i>	Getting all marketing event participants	/EventParticipants
<i>GET</i>	Getting a single marketing event participant	/EventParticipants ('<MktgEventParticipantUUID>')
<i>POST</i>	Deep-create or create a single marketing event participant	/MarketingEvents ('<MktgEventUUID>') / to_Participant
<i>PUT</i>	Updating a single marketing event participant	/EventParticipants ('<MktgEventParticipantUUID>')
<i>DELETE</i>	Deleting a single marketing event participant along with its interactions and survey response data	/EventParticipants ('<MktgEventParticipantUUID>')

## Properties

Parameter	Description	Creatable	Updatable	Key	Mandatory
MktgEventParticipantUUID	Unique identifier generated in SAP Marketing Cloud system for every marketing event participant			X	
MktgEventUUID	Unique identifier generated in SAP Marketing Cloud system for every marketing event				
MktgEventParticipantExternalID	External identifier of the imported marketing event participant	X			X
ContactOrigin	Origin of interaction contact data	X			X
MktgEngagementScore	Participation score	X	X		
TotalNumberOfMinutesAttended	Total number of minutes a participant attended the event live and on-demand	X	X		
NumberOfQuestions	Number of questions asked by participants	X	X		
NumberOfPollsAnswered	Number of polls responded by participants	X	X		
NumberOfSurveysAnswered	Number of surveys answered by participants	X	X		
NumberOfContentDownloads	Number of contents downloaded by participants	X	X		

Parameter	Description	Creatable	Updatable	Key	Mandatory
TotalNumberOfMinutesLive	Total number of minutes a participant attended the event live	X	X		
TotalNumberOfMinutesReplay	Total number of minutes a participant attended the event on-demand	X	X		
MktgEventParticipantStatus	Identifier of the participant status. A list of participant status is pre-delivered.	X	X		
MktgEventParticipantID	Identifier of the participant	X			
MktgEventParticipantStatusName	Name of the participant status				
CreationDateTime	Date and time when the marketing event was created				
CreatedByUser	Name of the user who created the marketing event				
LastChangeDateTime	Date and time when the marketing event was last changed				
LastChangedByUser	Name of the user who last changed the marketing event				
FullName	Full name of the participant				
EmailAddress	Email address of the participant				

Parameter	Description	Creatable	Updatable	Key	Mandatory
CompanyName	Company name of the participant				
IsEndOfPurposeBlocked	Indicates that a participant's data is discarded irreversibly on the event provider platform				

The following table shows if you are allowed to create participants based on the status of the event:

Participant Status	Event Status				
	In Preparation	Ready	Conducted	Cancelled	Archived
Attended	No	No	Yes	No	No
Registered	No	Yes	Yes	No	No
No show	No	No	Yes	No	No
Invited	No	No	No	No	No

## EventStatuses

**Resource Path:** /EventStatuses

You can perform the following operations on the EventStatuses resource:

Operations on the EventStatuses resource

HTTP Method	Description	URI
GET	Getting all marketing event participants	/EventStatuses

## Properties

Parameter	Description	Creatable	Updatable	Key
MktgEventStatus	Identifier of the marketing events status			X

Parameter	Description	Creatable	Updatable	Key
MktgEventStatusName	Name of the marketing event status			

## ParticipantQuestionAnswers

**Resource Path:** /ParticipantQuestionAnswers

You can perform the following operations on the EventStatuses resource:

Operations on the ParticipantQuestionAnswers resource

HTTP Method	Description	URI
<i>GET</i>	Getting all questions and answers of marketing event participant	/ParticipantQuestionAnswers
<i>GET</i>	Getting a single participant's questions and answers	/ParticipantQuestionAnswers ('<MktgEvtPrtcptntQstnAnswUUID>')
<i>GET</i>	Getting all questions and answers of a single marketing event's participants	/EventParticipants ('<MktgEventParticipantUUID>')/to_QuestionAnswer
<i>POST</i>	Deep-create or a single participant's questions and answers create	/EventParticipants ('<MktgEventParticipantUUID>')/to_QuestionAnswer
<i>PUT</i>	Updating a single participant's questions and answers	/ParticipantQuestionAnswers ('<MktgEvtPrtcptntQstnAnswUUID>')
<i>DELETE</i>	Deleting a single participant's questions and answers	/ParticipantQuestionAnswers ('<MktgEvtPrtcptntQstnAnswUUID>')

## Properties

Parameter	Description	Creatable	Updatable	Key
MktgEvtPrtcptntQstnAnswUUID	Unique identifier generated in SAP Marketing Cloud system for questions and answers of the marketing event participant			X

Parameter	Description	Creatable	Updatable	Key
MktgEventParticipantUUID	Unique identifier generated in SAP Marketing Cloud system for every marketing event participant			
MktgEventUUID	Unique identifier generated in SAP Marketing Cloud system for every marketing event			
MktgEvtPrtcpntQuestionText	Question asked by the marketing event participant	X	X	
MktgEvtPrtcpntAnswerText	Answer provided by the marketing event participant	X	X	

## Common Status and Error Codes

Code	Reason
204	Marketing events, participants, or Q&A updated successfully.
404	Not found, for example, marketing events, participants, or Q&A with the given key cannot be found in the system.
201	Marketing events, participants, or Q&A imported successfully.
400	Bad request, for example, a marketing event, participant, or Q&A with the same key already exists.

## Related Information

<https://api.sap.com> 



## 5.5.10.2 Payload Examples for Marketing Events

The following examples show how you can use the marketing events API:

### Payload Example for Deep-Crete of Marketing Events: POST

URI:/MarketingEvents

Request Example:[POST] https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_EVENT\_SRV/MarketingEvents

#### Sample Code

```
{
  "MktgEventExternalId" : "EXt1",
  "MktgEventProvider" : "ON24_ID",
  "MktgEventProviderAccount" : "31835",
  "MktgEventName" : "EVENT",
  "MktgEventDescription" : "EVENT",
  "MktgEventStatus" : "0001",
  "MediaType" : "EVENTS",
  "MktgEventEndDateTime" : "1018-10-10T12:34:32",
  "MktgEventStartDateTime" : "1018-10-10T12:34:32",
  "MktgEvtReplayAvailFromDateDateTime" : "1018-10-10T12:34:32",
  "MktgEvtReplayAvailToDateDateTime" : "1018-10-10T12:34:32",
  "MktgEventTimezone" : "CET",
  "MarketingArea" : "GLOBAL",
  "Language" : "EN",
  "MktgEventTypeName" : "Online",
  "MktgEventInfoURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventRegistrationURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventOnlineURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventReplayURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "to_Participant" : [{
    "MktgEventParticipantExternalID" : "barryallen@mailinator.com",
    "ContactOrigin" : "ON24_ID",
    "MktgEventParticipantID" : "ID1",
    "MktgEngagementScore" : 10,
    "TotalNumberOfMinutesAttended" : 20,
    "NumberOfQuestions" : 60,
    "NumberOfPollsAnswered" : 50,
    "NumberOfSurveysAnswered" : 0,
    "NumberOfContentDownloads" : 0,
    "TotalNumberOfMinutesLive" : 10,
    "TotalNumberOfMinutesReplay" : 0,
    "MktgEventParticipantStatus" : "1003",
    "to_QuestionAnswer" : [
      {
        "MktgEvtPrtcpcntQstnTxt" : "Can I replay the recording?",
        "MktgEvtPrtcpcntAnswTxt" : "Yes you can"
      }
    ]
  }
]
```

## Payload Example for Marketing Events Create: POST

URI:/MarketingEvents

**Request Example:**[POST] https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_EVENT\_SRV/MarketingEvents

### Sample Code

```
{
  "MktgEventExternalId" : "EXt11",
  "MktgEventProvider" : "ON24_ID",
  "MktgEventProviderAccount" : "31835",
  "MktgEventName" : "EVENT",
  "MktgEventDescription" : "EVENT",
  "MktgEventStatus" : "0004",
  "MediaType" : "EVENTS",
  "MktgEventEndDateTime" : "2019-04-10T12:37:33",
  "MktgEventStartDateTime" : "2019-04-10T11:34:32",
  "MktgEvtReplayAvailFromDate" : "2019-10-10T12:34:32",
  "MktgEvtReplayAvailToDate" : "2020-10-10T12:34:33",
  "MktgEventTimezone" : "CET",
  "MarketingArea" : "GLOBAL",
  "Language" : "EN",
  "MktgEventType" : "Online",
  "MktgEventInfoURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventRegistrationURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventOnlineURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventReplayURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A"
}
```

## Payload Example for Marketing Events Update: PUT

URI:/MarketingEvents ('<MktgEventUUID>')

**Request Example:**[PUT] https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_EVENT\_SRV/MarketingEvents('<MktgEventUUID>')

### Sample Code

```
{
  "MktgEventExternalId" : "EXt11",
  "MktgEventProvider" : "ON24_ID",
  "MktgEventProviderAccount" : "31835",
  "MktgEventName" : "EVENT",
  "MktgEventDescription" : "EVENT",
  "MktgEventStatus" : "0004",
  "MediaType" : "EVENTS",
  "MktgEventEndDateTime" : "2019-04-10T12:37:33",
  "MktgEventStartDateTime" : "2019-04-10T11:34:32",
  "MktgEvtReplayAvailFromDate" : "2019-10-10T12:34:32",
  "MktgEvtReplayAvailToDate" : "2020-10-10T12:34:33",
  "MktgEventTimezone" : "CET",
  "MarketingArea" : "GLOBAL",
}
```

```

"Language" : "EN",
"MktgEventTypeName" : "Online"
"MktgEventInfoURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
"MktgEventRegistrationURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
"MktgEventOnlineURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
"MktgEventReplayURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A"
}

```

## Payload Example for Marketing Events Participant Update: PUT

**URI:** /EventParticipants ('<MktgEventParticipantUUID>')

**Request Example:** [PUT] https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_EVENT\_SRV/EventParticipants ('<MktgEventParticipantUUID>')

### Sample Code

```

{
  "MktgEventParticipantExternalID" : "EventParticipant@eventstest.com",
  "ContactOrigin" : "ON24_ID",
  "MktgEngagementScore" : 10,
  "TotalNumberOfMinutesAttended" : 20,
  "NumberOfQuestions" : 60,
  "NumberOfPollsAnswered" : 50,
  "NumberOfSurveysAnswered" : 0,
  "NumberOfContentDownloads" : 0,
  "TotalNumberOfMinutesLive" : 10,
  "TotalNumberOfMinutesReplay" : 0,
  "MktgEventParticipantStatus" : "1002",
  "MktgEventParticipantID" : "ID1"
}

```

## Payload Example for Participant Questions and Answers Create: POST

**URI:** /EventParticipants ('<MktgEventParticipantUUID>')/to\_QuestionAnswer

**Request Example:** [POST] https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_EVENT\_SRV/EventParticipants ('<MktgEventParticipantUUID>')/to\_QuestionAnswer

### Sample Code

```

{
  "MktgEvtPrtcptntQstnTxt": "Can I replay the recording?",
  "MktgEvtPrtcptntAnswTxt": "Yes you can"
}

```

## Payload Example for Participant Questions and Answers Update: PUT

URI:/ParticipantQuestionAnswers ('<MktgEvtPrtcptQstnAnswUUID>')

**Request Example:** [PUT] https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_EVENT\_SRV/ParticipantQuestionAnswers ('<MktgEvtPrtcptQstnAnswUUID>')

### Sample Code

```
{
  "MktgEvtPrtcptQstnTxt": "Is recording available?",
  "MktgEvtPrtcptAnswTxt": "Yes "
}
```

## Payload Example for Marketing Events (\$batch): POST

URI:\$batch

**Request Example:** [POST] https://<Server>:<Port>/sap/opu/odata/SAP/API\_MKT\_EVENT\_SRV/\$batch

### Sample Code

```
--batch
Content-Type: multipart/mixed; boundary=changeset_01869434-0008-0001
--changeset_01869434-0008-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST MarketingEvents HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{
  "MktgEventExternalId" : "EXt1",
  "MktgEventProvider" : "ON24_ID",
  "MktgEventProviderAccount" : "31835",
  "MktgEventName" : "EVENT",
  "MktgEventDescription" : "EVENT",
  "MktgEventStatus" : "0001",
  "MediaType" : "EVENTS",
  "MktgEventEndDateTime" : "1018-10-10T12:34:32",
  "MktgEventStartDateTime" : "1018-10-10T12:34:32",
  "MktgEvtReplayAvailFromDate" : "1018-10-10T12:34:32",
  "MktgEvtReplayAvailToDate" : "1018-10-10T12:34:32",
  "MktgEventTimezone" : "CET",
  "MarketingArea" : "GLOBAL",
  "Language" : "EN",
  "MktgEventInfoURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventRegistrationURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventOnlineURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "MktgEventReplayURL" : "https://event.on24.com/wcc/r/1918973/12F107615216042B4905524623A91E9A",
  "to_Participant" : [{
    "MktgEventParticipantExternalID" : "barryallen@mailinator.com",
    "ContactOrigin" : "ON24_ID",
    "MktgEngagementScore" : 10,
  }
}
```

```

        "TotalNumberOfMinutesAttended" : 20,
        "NumberOfQuestions" : 60,
        "NumberOfPollsAnswered" : 50,
        "NumberOfSurveysAnswered" : 0,
        "NumberOfContentDownloads" : 0,
        "TotalNumberOfMinutesLive" : 10,
        "TotalNumberOfMinutesReplay" : 0,
        "MktgEventParticipantStatus" : "1003"
    }
}
--changeset_01869434-0008-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
POST MarketingEvents(guid'6c0b84b7-5523-1ed9-8ba5-cf63a92f034f')/
to_Participant HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{
    "MktgEventParticipantExternalID" : "eveprt2002@mailinator.com",
    "ContactOrigin" : "EMAIL",
    "MktgEngagementScore" : 10,
    "TotalNumberOfMinutesAttended" : 20,
    "NumberOfPollsAnswered" : 50,
    "NumberOfSurveysAnswered" : 0,
    "NumberOfContentDownloads" : 0,
    "TotalNumberOfMinutesLive" : 0,
    "TotalNumberOfMinutesReplay" : 0,
    "MktgEventParticipantStatus" : "1002"
}
--changeset_01869434-0008-0001
Content-Type: application/http
Content-Transfer-Encoding: binary
PUT EventParticipants(guid'6c0b84b7-5523-1ed9-8ba9-3c522fd891d6') HTTP/1.1
Content-Type: application/json
Content-Length: 1021
{
    "MktgEventParticipantExternalID" : "eveprt22@mailinator.com",
    "ContactOrigin" : "EMAIL",
    "MktgEngagementScore" : 10,
    "TotalNumberOfMinutesAttended" : 20,
    "NumberOfPollsAnswered" : 90,
    "NumberOfSurveysAnswered" : 0,
    "NumberOfContentDownloads" : 0,
    "TotalNumberOfMinutesLive" : 0,
    "TotalNumberOfMinutesReplay" : 0,
    "MktgEventParticipantStatus" : "0002"
}
--changeset_01869434-0008-0001--
--batch--

```

## 5.5.10.3 Function Imports

Function imports are used to perform custom operations on an entity, in addition to typical OData operations. This section also provides payload examples.

### Get Event Languages

HTTP Method	Description	Path
GET	Get the list of event languages	/GetEventLanguage

### Payload Examples

- Get Event Languages  
`/sap/opu/odata/SAP/API_MKT_EVENT_SRV/GetEventLanguage`

## 5.6 Commerce Marketing

The following APIs are available for the Commerce Marketing business area.

### [Recommendations \(SAP Business Technology Platform\) \[page 923\]](#)

Public OData API (`API_MKT_RECOMMENDATION_SRV`) that allows a client system to obtain recommendations from the SAP Marketing Cloud using the SAP Business Technology Platform.

### [Recommendations \[page 939\]](#)

The `PROD_RECO_RUNTIME_SRV` OData service enables customer channels to receive recommendations generated by *Recommendation*.

### [External Recommendations \[page 954\]](#)

Use the public OData API `API_MKT_EXTERNAL_RECMDN_SRV` to upload (import) recommendations from external sources.

### [Recommendations Interaction Data \[page 971\]](#)

OData service (`PROD_RECO_RUNTIME_SRV`) for posting interactions to an SAP HANA database.

### [Import Offers \[page 973\]](#)

Use the public OData API `CUAN_OFFER_IMPORT_SRV` to upload (import) offers from external sources.

### [Read Offers \[page 1002\]](#)

Public OData API (`API_MKT_OFFER_SRV`) for Offers

### [Discover Offers \[page 1008\]](#)

Use the API OData service `CUAN_OFFER_DISCOVERY_SRV` for SAP Marketing Cloud Offers to find suitable offers for a consumer.

[Coupons \[page 1026\]](#)

Public OData API (API\_MKT\_COUPON\_SRV) for Coupons.

## 5.6.1 Recommendations (SAP Business Technology Platform)

Public OData API (API\_MKT\_RECOMMENDATION\_SRV) that allows a client system to obtain recommendations from the SAP Marketing Cloud using the SAP Business Technology Platform.

### Technical Data

Name of Service	API_MKT_RECOMMENDATION_SRV
Authorization	No authorization is required, however, a <i>Security String</i> must be obtained from the <i>Recommendation Scenarios</i> app. Append the security string to every request for validation to occur.
	<div style="background-color: #f0f0f0; padding: 10px;"><p><b>i Note</b></p><p>The security string is only valid for requests that originate from your system, however, it should not be shared.</p></div>
OData Version	2.0
Root URL	<code>https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/</code>  The root URL points to the SAP Business Technology Platform tenant that is assigned to you. The Root URL (Recommendation Scenario URL) is provided through the <i>Recommendation Scenarios</i> app. With the proper user (for example, Marketing Expert or Business Analyst), you can log on to their SAP Marketing Cloud tenant and obtain the necessary information to consume the service.
Service Metadata URI	<code>https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/\$metadata</code>
Field Extensibility Supported	Yes. For more information, search for extensibility in <a href="#">Get Offers [page 928]</a> .

## Error Messages

If the API encounters an error, the following HTTP status codes are returned:

Code	Cause
429	The API has reached the maximum number of allocated API calls. The maximum number of allocated API calls is 200 per second.
400	A bad request was submitted. For example, the security string is invalid.

## Security String Parameters

All function import calls must include the security string parameters. The *Security String* for a given recommendation scenario, can be obtained through the *Recommendations Scenario* application, in the *Advanced* tab.

Parameter	Required	Data Type	Description
<code>_L54AD1F204_</code>	Yes	String	Encrypted key information (part of the security string).
<code>_K13_</code>	Yes	Integer	Secret key version (part of the security string).
<code>_K14_</code>	Yes	String	Generated HMAC string from the key information (part of the security string).
<code>_V_</code>	Yes	Integer	Version of the validation (part of the security string)

## Secure User ID

If you opt for using a recommendation scenario with the *Secure User ID* option, make sure that you enable this option in the *Recommendation Scenarios* application in your SAP Marketing Cloud system. This option affects your *Security String* for the scenario, so you need to copy the new value. The *User ID Salt* value (which needs to be passed along with the *User ID* in your HTTP request headers) can also be obtained in the same application.

## Technical Field Documentation and Payload Examples

For function import technical field documentation and payload examples, see [Function Imports \[page 925\]](#).



## 5.6.1.1 Function Imports

Function imports are used to perform custom operations on an entity, in addition to typical OData operations. This section contains technical field documentation and payload examples for the following function imports:

### [Get Recommendations \[page 925\]](#)

`GetRecommendations` retrieves recommendations from either the SAP Business Technology Platform (cached recommendations) or the SAP Marketing Cloud.

### [Get Offers \[page 928\]](#)

`GetOfferRecommendations` retrieves offer recommendations (offer content) from either SAP Business Technology Platform (cached recommendations) or SAP Marketing Cloud.

### [Get Products \[page 933\]](#)

`GetProducts` retrieves product master data from either SAP Business Technology Platform (cached products) or SAP Marketing Cloud.

### [Send Interactions \[page 934\]](#)

`SendProductClickThrough` and `SendOfferClickThrough` send interactions that occur between consumers and recommendations.

### 5.6.1.1.1 Get Recommendations

`GetRecommendations` retrieves recommendations from either the SAP Business Technology Platform (cached recommendations) or the SAP Marketing Cloud.

<b>HTTP Method</b>	GET
<b>Function Import</b>	<code>GetRecommendations</code>  Retrieves recommendations from either the SAP BTP (cached recommendations) or the SAP Marketing Cloud. Depending on the Recommendation Scenario, the function import can either return product or offer (offer content GUID and score) recommendations.  Each call to this service results in an impression being recorded for the scenario. When a recommendation is retrieved from the SAP Marketing Cloud, the impression is recorded in the back-end system. However, if the recommendation is retrieved from the cache, the impression is stored in a database table in the SAP BTP. The impressions that are stored in the database table are periodically aggregated and then posted to the SAP Marketing Cloud.
<b>URL</b>	<code>https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/GetRecommendations</code>

## HTTP Request Header Fields

Field	Mandatory	Data Type	Max. Length	Description
<code>_u_</code>	Optional – Appears if <a href="#">Secure User ID</a> checkbox is checked in <a href="#">Recommendation Scenarios</a> app.	String	N/A	The user ID.
<code>_h_</code>	Optional – Appears if <a href="#">Secure User ID</a> checkbox is checked in <a href="#">Recommendation Scenarios</a> app.	String	N/A	An SHA-256 cryptographic hash of the user ID and the salt

## Request Parameters

Field	Mandatory	Data Type	Max. Length	Description
<code>LeadingItemIds</code>	Only for association algorithms.	String	1000	The comma-separated product IDs to be passed to the recommendation scenario.
<code>LeadingItemType</code>	Only when <code>LeadingItemIds</code> is provided.	String	50	The product type (origin) as defined in SAP Marketing Cloud.
<code>BasketItemIds</code>	No	String	1000	The comma-separated product IDs which are already in the cart.
<code>BasketItemType</code>	Only when <code>BasketItemIds</code> is provided	String	50	The product type (origin) as defined in SAP Marketing Cloud.
<code>ContextParameters</code>	No	String	1000	The context parameters as configured in SAP Marketing Cloud. For example, interaction type, interaction contact type, or any of the other algorithm data source pre-filters (standard delivery or custom). For more information, see <a href="#">Algorithm Data Source Pre-filters</a> and <a href="#">Recommendation Data Source Pre-Filters</a> .

Field	Mandatory	Data Type	Max. Length	Description
UserId	No	String	71	The user ID, depending on the specified user type.  This parameter is ignored if the <a href="#">Secure User ID</a> checkbox is checked in the custom <a href="#">Recommendation Scenarios</a> app.
UserType	Only when <a href="#">UserId</a> is provided or when <a href="#">Secure User ID</a> checkbox is checked in <a href="#">Recommendation Scenarios</a> app.	String	50	The user facet, as defined in SAP Marketing Cloud.

## Example of Request

### Sample Code

```
https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/
GetRecommendations
?LeadingItemIds='11100,10020'
&LeadingItemType='SAP_COMMERCE_PRODUCT'
&BasketItemIds='10013'
&BasketItemType='SAP_COMMERCE_PRODUCT'
&UserId='username@sap.com'
&UserType='EMAIL'
&ContextParameters='PARAM1_ID eq 123,PARAM2_ID eq xyz'
&_L54AD1F204_='c2N1bmFyaW89UkVDTyZ0ZW5hbnQ9W215LXRlbnFudF0ub25kZW1hbmQuY29tJnJ
1Y29fc2N1bmFyaW89U0FQX0NST1NTX1NFTExfV0VCX1BST0RVQ1RfREVUQU1MUyZzZWN1cmVfdXNlc
j0='
&_K13_=20
&_V_=2
&_K14_='7875919344b1bbf1970ae84c9a180828c84e6cda40c8873e4ed88abaac0aa7ba2'
```

### Note

The scenario ID and the marketing tenant are encoded in the `_L54AD1F204_` parameter. For example, a decoded version of the parameter contained in the example request would be `RECO&tenant=[my-tenant].ondemand.com&reco_scenario=SAP_CROSS_SELL_WEB_PRODUCT_DETAILS&secure_user=.`

## Example of Response

### Sample Code

```
<GetRecommendations xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m=http://schemas.microsoft.com/ado/2007/08/dataservices/metadata>
  <element m:type="API_MKT_RECOMMENDATION_SRV.Recommendation">
    <ResultObjectId>10023</ResultObjectId>
    <ResultObjectType>SAP_HYBRIS_PRODUCT</ResultObjectType>
    <ResultObjectScore>1.00000</ResultObjectScore>
  </element>
  <element m:type="API_MKT_RECOMMENDATION_SRV.Recommendation">
    <ResultObjectId>10024</ResultObjectId>
    <ResultObjectType>SAP_HYBRIS_PRODUCT</ResultObjectType>
    <ResultObjectScore>0.79831</ResultObjectScore>
  </element>
</GetRecommendations>
```

### 5.6.1.1.2 Get Offers

GetOfferRecommendations retrieves offer recommendations (offer content) from either SAP Business Technology Platform (cached recommendations) or SAP Marketing Cloud.

<b>HTTP Method</b>	GET
<b>Function Import</b>	GetOfferRecommendations  Retrieves offer recommendations from either SAP BTP (cached recommendations) or SAP Marketing Cloud.  This function import returns offer content, as opposed to the offer content GUID and score returned by the GetRecommendations function import.
<b>URL</b>	https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/GetOfferRecommendations
<b>Field Extensibility</b>	The OfferContent complex type supports field extensibility. For more information see, <a href="#">Custom Fields for Offer Header and Offer Content</a> .

### HTTP Request Header Fields

Field	Mandatory	Data Type	Description
_u_	Optional – Appears if <a href="#">Secure User ID</a> checkbox is checked in <a href="#">Recommendation Scenarios</a> app.	String	The user ID.

Field	Mandatory	Data Type	Description
_h_	Optional – Appears if <i>Secure User ID</i> checkbox is checked in <i>Recommendation Scenarios</i> app.	String	An SHA-256 cryptographic hash of the user ID and the salt

## Request Parameters

Field	Mandatory	Data Type	Max. Length	Description
LeadingItemIds	Only for association algorithms.	String	1000	The comma-separated product IDs to be passed to the recommendation scenario.
LeadingItemType	Only when <code>LeadingItemIds</code> is provided.	String	50	The product type (origin) as defined in the SAP Marketing Cloud.
BasketItemIds	No	String	1000	The comma-separated product IDs which are already in the cart.
BasketItemType	Only when <code>BasketItemIds</code> is provided	String	50	The product type (origin) as defined in the SAP Marketing Cloud.
ContextParameters	No	String	1000	The context parameters as defined in the SAP Marketing Cloud, e.g.: Interaction Type, Interaction Contact Type.

### i Note

Additional request parameters that are not defined in the `ContextParameters` are appended.

Field	Mandatory	Data Type	Max. Length	Description
UserId	No	String	71	<p>The user ID, depending on the specified user type.</p> <p>This parameter is ignored if the <i>Secure User ID</i> checkbox is checked in the custom <i>Recommendation Scenarios</i> app.</p>
UserIdType	Only when <i>UserId</i> is provided or when <i>Secure User ID</i> checkbox is checked in <i>Recommendation Scenarios</i> app.	String	50	The user facet, as defined in SAP Marketing Cloud.
Language	No	String	2	The ISO language code of the offer content. In a Web shop, the language may correspond to the user's logon language. If no language is passed to the OData service, the result contains all available languages.
Position	No	String	40	The position in the Web shop where offers are to be displayed, such as <i>Top</i> or <i>Bottom</i> . This information must be maintained for the offer content in the SAP Marketing Cloud.
CommunicationMedium	No	String	20	The communication medium as defined in the SAP Marketing Cloud. The parameter filters the offer content by the communication medium.
OfferContentType	No	String	2	The <i>OfferContentType</i> parameter filters offers by content type.

Field	Mandatory	Data Type	Max. Length	Description
WithCoupon	No	String	1	<p>The following <code>WithCoupon</code> parameter values filter offers by whether or not they have a coupon assigned:</p> <ul style="list-style-type: none"> <li>'x' retrieves offers with coupons assigned.</li> <li>' ' retrieves offers without coupons assigned.</li> </ul> <div style="border: 1px solid #0070c0; padding: 5px; margin-top: 10px;"> <p><b>i Note</b></p> <p>Requests without the parameter retrieve offers with and without coupons assigned.</p> </div>
MarketingArea	No	String	40	The marketing area as defined in the SAP Marketing Cloud. The parameter filters offers by marketing area.

## Example of Request

### Sample Code

```

https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/
GetOfferRecommendations
?LeadingItemIds='PRODUCT_ID_1'
&LeadingItemType='SAP_HYBRIS_PRODUCT'
&UserId='username@sap.com'
&UserType='EMAIL'
&_L54AD1F204_='c2N1bmFyaW89UkVDTyZ0ZW5hbnQ9W215LXRlbnFudF0uczRoYW5hLm9uZGVtYW5kLmNvbSZyZW5hbnQ9ZWNvX3NjZW5hcmlvPVRPUF9PRkZFU1Mmc2VjdXJlX3VzZXI9WA=='
&_K13_=20
&_V_=2
&_K14_='343fd3e7f98cc3842765a4fe965685344560c05207075519ab2e5f9248b51810'

```

### i Note

The scenario ID and the marketing tenant are encoded in the `_L54AD1F204_` parameter. For example, a decoded version of the parameter contained in the example request would be `RECO&tenant=[my-tenant].ondemand.com&reco_scenario=SAP_CROSS_SELL_WEB_PRODUCT_DETAILS&secure_user=.`

## Example of Response

### Sample Code

```
<?xml version='1.0' encoding='utf-8'?>
<GetOfferRecommendations
xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<element m:type="API_MKT_RECOMMENDATION_SRV.OfferContent">
  <MarketingOffer>32</MarketingOffer>
  <MarketingOfferContent>00001</MarketingOfferContent>
  <OfferContentType>01</OfferContentType>
  <OfferContentTypeName>Image</OfferContentTypeName>
  <CommunicationMedium>EMAIL</CommunicationMedium>
  <CommunicationMediumName>Email</CommunicationMediumName>
  <LanguageISOCODE>EN</LanguageISOCODE>
  <MarketingArea>CXXGLOBAL</MarketingArea>
  <OfferContentPosition>Home</OfferContentPosition>
  <OfferContentSourceURL>https://img.freepik.com/free-vector/colorful-
shopping-sale-banner-template_1201-1308.jpg?size=338&ext=jpg</
OfferContentSourceURL>
  <OfferContentSourceURLDesc>50% Special Offer</OfferContentSourceURLDesc>
  <OfferContentTargetURL>https://img.freepik.com/free-vector/colorful-
shopping-sale-banner-template_1201-1308.jpg?size=338&ext=jpg</
OfferContentTargetURL>
  <OfferContentTargetURLDesc>50% Special Offer</OfferContentTargetURLDesc>
  <CouponUUID></CouponUUID>
  <ExternalOffers></ExternalOffers>
</element>
</GetOfferRecommendations >
```

### Note

The same parameters could be used with the `getRecommendations` function import and result in the following response:

### Sample Code

```
<?xml version='1.0' encoding='utf-8'?>
<GetRecommendations
xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <element m:type="API_MKT_RECOMMENDATION_SRV.Recommendation">
    <ResultObjectId>00163E2C7B391ED8AB8E78EE6425E6A0</ResultObjectId>
    <ResultObjectType>CUAN_OFFER</ResultObjectType>
    <ResultObjectScore>1.00000</ResultObjectScore>
  </element>
</GetRecommendations>
```



### 5.6.1.1.3 Get Products

GetProducts retrieves product master data from either SAP Business Technology Platform (cached products) or SAP Marketing Cloud.

<b>HTTP Method</b>	GET
<b>Function Import</b>	GetProducts Retrieves product master data from either SAP BTP (cached products) or SAP Marketing Cloud.
<b>URL</b>	https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/GetProducts

#### Request Parameters

Field	Mandatory	Data Type	Max. Length	Description
ProductIds	Yes	String	2000	The comma-separated product IDs to be passed to the recommendation scenario.
ProductOrigin	Yes	String	50	The product type (origin) as defined in SAP Marketing Cloud.
Language	Yes	String	2	The Language of the master data.

#### Note

All of the parameters in the table are query strings.

#### Example of Request

##### Sample Code

```
https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/
GetProducts
?ProductIds='PRODUCT1,PRODUCT2'
&ProductOrigin='SAP_COMMERCE_PRODUCT'
&Language='EN'
&_L54AD1F204_='c2N1bmFyaW89UkVDTyZ0ZW5hbnQ9W01ZLU1LVC1URU5BT1RdLnM0aGFuYS5vbmR
lbWFuZC5jb20mcmVjb19zY2VuYXJpbz1TQVBfVE9QX1NFTExFUL9IT01FX1BBR0U='
&_K13_=1
&_V_=2
&_K14_='9fd60e90b810e9c24fbabd44d2158d564333b4a5f13fc597b343c5bdafc50ea4'
```

## Example of Response

### Sample Code

```
<GetProducts xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <element m:type="com.sap.cec.mkt.recommendation.ProductStructure">
    <ProductId>PRODUCT1</ProductId>
    <ProductDescription>Product One Description Text</ProductDescription>
    <ProductName>Product 1 Name</ProductName>
    <ProductTargetUrl>
      https://[some_host]/.../PRODUCT_DETAILS/PRODUCT1.html
    </ProductTargetUrl>
    <ProductImageUrl>
      https://[some_host]/sap/files/PRODUCT_BRAND/PRODUCT_CATEGORY/
      PRODUCT1.jpg
    </ProductImageUrl>
  </element>
  <element m:type="com.sap.cec.mkt.recommendation.ProductStructure">
    <ProductId>PRODUCT2</ProductId>
    <ProductDescription>Product Two Description Text</ProductDescription>
    <ProductName>Product 2 Name</ProductName>
    <ProductTargetUrl>
      https://[some_host]/.../PRODUCT_DETAILS/PRODUCT1.html </
      ProductTargetUrl> <ProductImageUrl> https://[some_host]/sap/files/
      PRODUCT_BRAND/PRODUCT_CATEGORY/PRODUCT2.jpg
    </ProductImageUrl>
  </element>
</GetProducts>
```

### 5.6.1.1.4 Send Interactions

SendProductClickThrough and SendOfferClickThrough send interactions that occur between consumers and recommendations.

#### Send Product Click-Throughs

HTTP Method	GET
Function Import	SendProductClickThrough Send interactions that occur between consumers and the products that are recommended.
URL	https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/ SendProductClickThrough

## Request Parameters

Field	Mandatory	Data Type	Max. Length	Description
ItemId	Yes	String	50	The comma-separated item IDs to be passed to the recommendation scenario.
ItemType	Yes	String	30	The item type as defined in SAP Marketing Cloud.
UserId	No	String	71	The user ID, depending on the specified user type.  This parameter is ignored if the <i>Secure User ID</i> checkbox is checked in the custom <i>Recommendation Scenarios</i> app.
UserType	Only when <code>UserId</code> is provided or when <i>Secure User ID</i> checkbox is checked in <i>Recommendation Scenarios</i> app.	String	50	The <i>User Type</i> configured for the recommendation scenario in SAP Marketing Cloud.
TargetUrl	No	String	2000	The target URL to be used for the redirection.
CommunicationMedium	Yes	String	20	The <i>Communication Medium</i> configured for the recommendation scenario in SAP Marketing Cloud.
SourceObjectId	Yes	String	50	The ID of the object that triggered the interaction. For example, a session or sales order ID.
SourceObjectType	Yes	String	30	A custom defined string that describes the type of object passed as the <code>SourceObjectId</code> .

Field	Mandatory	Data Type	Max. Length	Description
SourceSystemId	No	String	255	The ID of the system that is receiving the recommendations, and providing interaction data. For example, a commerce webshop (www.xyz.com).
SourceSystemType	Required with SourceSystemId	String	20	The type of system that is receiving the recommendations and providing interaction data. For example, <b>SAP _Commerce</b> .

### Example of Request

#### Sample Code

```
https://recow62890cfa.cert.int.sap.hana.ondemand.com/reco/api/
API_MKT_RECOMMENDATION_SRV/SendProductClickThrough
?CommunicationMedium='EMAIL'
&SourceObjectId='mySourceObjectId'
&SourceObjectType='mySourceObjectType'
&ItemID='myItemId'
&ItemType='SAP_PRODUCT'
&L54AD1F204='c2N1bmFyaW89UkVDTyZ0ZW5hbnQ9bXkzMDA0NzAuczRoYW5hLm9uZGVtYW5kLmN
vbSZyZWNVX3NjZW5hcmlvPUFMTE9GRkVSUw=='
&K13_1=1
&V_2=2
&K14_='1f18e01b8c3ceac1bf6669203d4f60a5a137cc2142ce01165b8297b62400c9f9'
```

### Example of Response

#### Sample Code

```
<?xml version='1.0' encoding='utf-8'?>
<SendProductClickThrough
xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">true
</SendProductClickThrough>
```

## Send Offer Click-Throughs

HTTP Method	GET
Function Import	SendOfferClickThrough
	Send interactions that occur between consumers and the offers that are recommended.

**URL** `https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/SendOfferClickThrough`

---

## Request Parameters

Field	Mandatory	Data Type	Max. Length	Description
MarketingOffer	Yes	String	10	The comma-separated offer IDs to be passed to the recommendation scenario.
MarketingOfferContent	Yes	String	5	The Marketing Offer Content ID
ExternalOffer	No	String	60	External Offer ID
ExternalOfferOrigin	Only when ExternalOffer is provided.	String	30	External Offer Origin
Coupon	No	String	32	Marketing Coupon ID
UserId	No	String	71	The user ID, depending on the specified user type.  This parameter is ignored if the <a href="#">Secure User ID</a> checkbox is checked in the custom <a href="#">Recommendation Scenarios</a> app.
UserIdType	Only when UserId is provided or when <a href="#">Secure User ID</a> checkbox is checked in <a href="#">Recommendation Scenarios</a> app.	String	50	The user facet, as defined in SAP Marketing Cloud.
CommunicationMedium	Yes	String	20	The <a href="#">Communication Medium</a> configured for the recommendation scenario in SAP Marketing Cloud.
SourceObjectId	Yes	String	50	The ID of the object that triggered the interaction. For example, a session or sales order ID.
SourceObjectType	Yes	String	30	A custom defined string that describes the type of object passed as the SourceObjectId.

Field	Mandatory	Data Type	Max. Length	Description
SourceSystemId	No	String	255	The ID of the system that is receiving the recommendations, and providing interaction data. For example, a commerce webshop (www.xyz.com).
SourceSystemType	Required with SourceSystemId	String	20	The type of system that is receiving the recommendations and providing interaction data. For example, <a href="#">SAP Commerce</a> .

### Example of Request

#### Sample Code

```
https://[Recommendation Scenario URL]/api/API_MKT_RECOMMENDATION_SRV/
SendOfferClickThrough
?CommunicationMedium='EMAIL'
&SourceObjectId='mySourceObjectId'
&SourceObjectType='mySourceObjectType'
&MarketingOffer='32'
&MarketingOfferContent='00001'
&_L54AD1F204_='c2NlbnFyaW89UkVDTyZ0ZW5hbnQ9bXkzMdA0NzAuczRoYW5hLm9uZGVtYW5kLmN
vbSzyZWNvX3NjZW5hcmlvPUFMTE9GRkVSUw=='
&_K13_=1
&_v_=2
&_K14_='1f18e01b8c3ceac1bf6669203d4f60a5a137cc2142ce01165b8297b62400c9f9'
```

### Example of Response

#### Sample Code

```
<?xml version='1.0' encoding='utf-8'?>
<SendOfferClickThrough
xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">true
</SendOfferClickThrough>
```

## 5.6.2 Recommendations

The `PROD_RECO_RUNTIME_SRV` OData service enables customer channels to receive recommendations generated by *Recommendation*.

### Prerequisites

- You have assigned the *Marketing - Recommendation Integration* communication scenario to your communication user in *Maintain Communication Users*.
- You have setup the communication system by doing the following:
  1. From the SAP Fiori launchpad, choose ► *Communication Management* ► *Communication Systems* ►.
  2. Create a communication system, and enter an ID and a system name.
  3. Under *Technical Data*, enter the *Host Name* of the SAP Cloud Integration system to be connected.
  4. Choose *Save* and return back to the SAP Fiori launchpad.
- You have setup the communication arrangement by doing the following:
  1. From the SAP Fiori launchpad, choose the *Communication Arrangements* app.
  2. Create a communication arrangement for the scenario `SAP_COM_0019` (Marketing - Recommendation Integration), and enter an arrangement name.
  3. In the *Communication Arrangement* for Marketing - Recommendation Integration, choose the communication system that you created earlier.
  4. Under *Inbound Communication*, enter your communication user name, and choose an authentication method.
  5. Under *Inbound Services*, the system provides the relevant services from the communication scenario.
  6. Choose *Save* and return back to the SAP Fiori launchpad.
- To receive the recommendations, call the service using the deep insert functionality of OData. For more information about the deep insert functionality of OData, see [https://help.sap.com/viewer/product/SAP\\_GATEWAY/2.0/en-US](https://help.sap.com/viewer/product/SAP_GATEWAY/2.0/en-US). Choose ► *Developer's Guide* ► *OData Channel* ► *Advanced Features* ► *Deep Insert* ►.

### Details of Service Entity

**Root URL:** `https://<Server>:<Port>/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/RecommendationScenarios`

**Request Mode:** POST

**Entity Data Model:** `RecommendationScenarios`

The nested structure of the entities that can be navigated to from the `RecommendationScenarios` entity are as follows:

- `RecommendationScenarios`
  - `Scenarios`

- LeadingObjects
- BasketObjects
- ContextParams
- ScenarioHashes
- ResultObjects

### RecommendationScenario Entity Parameters

The following table contains the parameters of the `RecommendationScenario` entity:

Property	Description	Edm Core Type	Max Length	Key
<code>UserId</code>	The ID of the user who performs the interaction, for example, customer ID or contact ID.	Edm.String	50	TRUE
<code>UserType</code>	The type of user who performs the interaction, for example, SAP Commerce Consumer or SAP Marketing Interaction Contact.	Edm.String	20	TRUE
<code>ExternalTracking</code>	A flag that implies external tracking of impressions using the <code>PostImpressions</code> function import (Optional).	Edm.Boolean	1	FALSE

### Scenario Entity Parameters

The following table contains the parameters of the `Scenario` entity:

Property	Description	Edm Core Type	Max Length	Key
<code>ScenarioId</code>	The scenario ID represents a model type and related usage information, for example, promotion model type and user type.	Edm.String	50	TRUE



Property	Description	Edm Core Type	Max Length	Key
HashId	A hash associated to a specific user. The hash accelerates retrieving recommendations from the cache of an optimized algorithm.	Edm.String	32	FALSE

### i Note

You can use the `ProductRecoScenario` entity to enable your customer channel to choose `ScenarioId` from a value help. For more information, see [Value Help Enabling Entities \[page 952\]](#).

### LeadingObject Entity Parameters

The following table contains the parameters of the `LeadingObject` entity:

Property	Description	Edm Core Type	Max Length	Key
LeadingObjectId	The ID of the leading object, for example, material number.	Edm.String	50	TRUE
LeadingObjectType	A recommendation data source type that is defined to an <code>ITEM</code> data source class, for example, SAP Commerce Product.	Edm.String	30	TRUE

### i Note

You can use the `ItemSourceTypes` entity to enable your customer channel to choose `LeadingObjectType` from a value help. For more information, see [Value Help Enabling Entities \[page 952\]](#).

### BasketObject Entity Parameters

The following table contains the parameters of the `BasketObjectId` entity:

Property	Description	Edm Core Type	Max Length	Key
BasketObjectId	The ID of the leading object, for example, material number.	Edm.String	50	TRUE

Property	Description	Edm Core Type	Max Length	Key
BasketObjectType	A recommendation data source type that is defined to an ITEM data source class, for example, SAP Commerce Product.	Edm.String	30	TRUE

### i Note

You can use the `ItemSourceTypes` entity to enable your customer channel to choose `BasketObjectType` from a value help. For more information, see [Value Help Enabling Entities \[page 952\]](#).

### ContextParam Entity Parameters

The following table contains the parameters of the `ContextParam` entity:

Property	Description	Edm Core Type	Max Length	Key
ContextId	The prefilter parameter ID.	Edm.Int32	n.a.	TRUE
ContextParamId	The parent prefilter parameter ID.	Edm.Int32	n.a.	FALSE
Value	The value of the prefilter parameter.	Edm.String	100	FALSE
ValueType	The value type of the prefilter parameter.	Edm.String	32	FALSE

### ScenarioHash Entity Parameters

The following table contains the parameters of the `ScenarioHashes` entity:

Property	Description	Edm Core Type	Max Length	Key
ScenarioId	The recommendation scenario ID.	Edm.String	50	TRUE
HashID	A hash returned by the system that is associated to a specific user. The hash accelerates retrieving recommendations from the cache of an optimized algorithm.	Edm.String	32	TRUE

Property	Description	Edm Core Type	Max Length	Key
ExpiresOn	Expiry date of HashID.	Edm.DateTime		FALSE
ResultScope	The scope of the result. For example, Generic, Restricted, or Personalized.	Edm.String	1	FALSE

### ResultObject Entity Parameters

The following table contains the parameters of the `ResultObject` entity:

Property	Description	Edm Core Type	Max Length	Key
ScenarioId	The recommendation scenario ID.	Edm.String	50	TRUE
ResultObjectType	A recommendation data source type that is defined to an <code>ITEM</code> data source class. For example, SAP Commerce Product.	Edm.String	30	TRUE

#### i Note

The `ResultObjectType` (Recommendation Type) parameter is defined in the *Recommendation Model Types* app. The *Recommendation Type* reflects either offers or the products contained in the system receiving the recommendations.

#### ❖ Example

To enable an SAP Commerce Web shop to receive recommendations; SAP Commerce Product is defined as the Recommendation Type. Only Recommendation objects of type SAP Commerce Product will be returned by the API.

ResultObjectId	The ID of the result object, for example, material number.	Edm.String	50	TRUE
ResultObjectScore	The score of the result object.	Edm.Decimal	10.5	FALSE

### PostImpressions Function Import Parameters

The following table contains the parameters of the `PostImpressions` function import:

Property	Description	Edm Core Type	Max Length
<code>ScenarioId</code>	The recommendation scenario ID.	Edm.String	50
<code>TimeStamp</code>	The timestamp of the impression.	Edm.DateTimeOffset	30
<code>ImpressionCount</code>	The total number of impressions performed.	Edm.Int16	
<code>ItemCount</code>	The total number of Item recommended.	Edm.Int16	

If the `ExternalTracking` parameter in the `RecommendationScenario` entity is set to `TRUE`, as it is in the HTTP post request example, SAP Marketing Cloud does not count the impressions for the recommendation scenario that is being solicited. To keep the number of impressions in SAP Marketing Cloud accurate, it is necessary for the external system to convey the impression count. To do so, an additional separate call must be made to increase the impression count. For example, if the scenario `INT_TEST` returns 3 items that were consumed once; the additional call would contain the following:

```
https://[sap -marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/
PostImpressions?ScenarioId='
INT_TEST' &TimeStamp=datetimeoffset '2016-12-03T12:45:29Z' &ImpressionCount=1&ItemCoun
t=3&saml2=disabled
```

## JSON Examples

### HTTP Post Request Using Deep Insert Functionality of OData in JSON Encoding

```
{
  "UserId" : "40F2E9306E391ED59BDE581AFE71F329 ",
  "UserType" : "COOKIE_ID",
  "ExternalTracking" : true,
  "Scenarios" :
  [
    {
      "ScenarioId" : "INT_TEST",
      "HashId" : "D33DD1F71615D50334FB2F1043365430",
      "LeadingObjects" :
      [
        {
          "LeadingObjectType" : "SAP_ERP_MATNR",
          "LeadingObjectId" : "M-01"
        }
      ],
      "BasketObjects" :
      [
        {
          "BasketObjectType" : "SAP_ERP_MATNR",
          "BasketObjectId" : "100-100"
        }
      ]
    }
  ],
  "ContextParams" : [],
  "ScenarioHashes" : [],
  "ResultObjects" : []
}
```

## HTTP Post Response Payload in JSON Encoding

```
{
  "d":
  {
    "__metadata":
    {
      "id": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(UserId='',UserType='')"
      "uri": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(UserId='',UserType='')"
      "type": "PROD_RECO_RUNTIME_SRV.RecommendationScenario"
    }
    "UserId": ""
    "UserType": ""
    "ExternalTracking": true,
    "ScenarioHashes": {
      "results": [
        {
          "__metadata": {
            "id": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/ScenarioHashes('SAP_TOP_SELLERS_EMAIL_CAMPAIGN')",
            "uri": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/ScenarioHashes('SAP_TOP_SELLERS_EMAIL_CAMPAIGN')",
            "type": "PROD_RECO_RUNTIME_SRV.ScenarioHash"
          },
          "ScenarioId": "SAP_TOP_SELLERS_EMAIL_CAMPAIGN"
          "HashId": "D33DD1F71615D50334FB2F1043365429",
          "ExpiresOn": "/Date(1478180969524)/",
          "ResultScope": "G"
        },
      ],
    },
    "ResultObjects":
    {
      "results": [3]
      0:
      {
        "__metadata":
        {
          "id": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectI
d='100-100')"
          "uri": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectI
d='100-100')"
          "type": "PROD_RECO_RUNTIME_SRV.ResultObject"
        }
        "ScenarioId": "INT_TEST"
        "ResultObjectType": "SAP_ERP_MATNR"
        "ResultObjectId": "100-100"
        "ResultObjectScore": "1.00000"
      }
      1:
      {
        "__metadata":
        {
          "id": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectI
d='P-102')"
          "uri": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
```

```

ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectI
d='P-102')"
  "type": "PROD_RECO_RUNTIME_SRV.ResultObject"
}
"ScenarioId": "INT_TEST"
"ResultObjectType": "SAP_ERP_MATNR"
"ResultObjectId": "P-102"
"ResultObjectScore": "0.01906"
}
2:
{
  "_metadata":
  {
    "id": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectI
d='P-100')"
    "uri": "https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectI
d='P-100')"
    "type": "PROD_RECO_RUNTIME_SRV.ResultObject"
  }
  "ScenarioId": "INT_TEST"
  "ResultObjectType": "SAP_ERP_MATNR"
  "ResultObjectId": "P-100"
  "ResultObjectScore": "0.00554"
}
}
}
}
}

```

## XML Examples

### HTTP Post Request Payload in XML Encoding

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<entry
xml:base="https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/"
xmlns:m="http://www.w3.org/2005/Atom" xmlns:d="http://schemas.microsoft.com/ado/
2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns="http://www.w3.org/2005/Atom">

  <id>https://[sap -marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/
RecommendationScenarios(0)</id>
  <title type="text">RecommendationScenarios</title>
  <link rel="self" href="RecommendationScenarios"
title="RecommendationScenarios(0)"/>
  <link
rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/
Scenarios"
type="application/atom+xml;type=feed"
title="Scenarios" href="RecommendationScenarios(0)/Scenarios">
    <m:inline>
      <feed>
        <title type="text">Scenarios</title>
        <id>https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(0)/Scenarios</id>
        <link rel="self" title="Scenarios"
href="RecommendationScenarios(0)/Scenarios" />
      </feed>
    </m:inline>
  </link>
</entry>

```

```

        <id>https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/Scenarios(0)</id>
        <title type="text">Scenarios</title>
        <link
related/Scenarios"
        type="application/atom+xml;type=entry"
        title="Scenarios" href="RecommendationScenarios(0)/
Scenarios"/>
        <category term="PROD_RECO_RUNTIME_SRV.Scenario"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
        <content type="application/xml">
            <m:properties>
                <d:ScenarioId>[Scenario ID]</d:ScenarioId>
                <d:HashId></d:HashId>
            </m:properties>
        </content>
        </entry>
    </feed>
</m:inline>
</link>

    <link rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/
ScenarioHashes"
        type="application/atom+xml;type=feed"
        title="ScenarioHashes" href="RecommendationScenarios(0)/ScenarioHashes">
        <m:inline>
            <feed>
                <title type="text">Scenarios</title>
                <id>https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(0)/ScenarioHashes</id>
                <link rel="self" title="ScenarioHashes"
href="RecommendationScenarios(0)/ScenarioHashes" />
                <entry>
                    <id>https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/ScenarioHashes(0)</id>
                    <title type="text">ScenarioHashes</title>
                    <link
related/ScenarioHashes"
                    type="application/atom+xml;type=entry"
                    title="ScenarioHashes" href="RecommendationScenarios(0)/
ScenarioHashes"/>
                    <category term="PROD_RECO_RUNTIME_SRV.ScenarioHashes"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
                    <content type="application/xml">
                        <m:properties>

                            </m:properties>
                    </content>
                </entry>
            </feed>
        </m:inline>
    </link>
    <link
rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/
ResultObjects"
        type="application/atom+xml;type=feed"
        title="ResultObjects" href="RecommendationScenarios(0)/ResultObjects">
        <m:inline>
            <feed>
                <title type="text">ResultObjects</title>
                <id>https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(0)/ResultObjects</id>
                <link rel="self" title="ResultObjects"
href="RecommendationScenarios(0)/ResultObjects" />
                <entry>

```

```

        <id>https://[sap -marketing-server]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/ResultObjects(0)</id>
        <title type="text">ResultObjects</title>
        <link
rel="http://schemas.microsoft.com/ado/2007/08/dataservices/
related/ResultObjects"
type="application/atom+xml;type=entry"
title="ResultObjects" href="RecommendationScenarios(0)/
ResultObjects"/>
        <category term="PROD_RECO_RUNTIME_SRV.ResultObjects"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
        <content type="application/xml">
            <m:properties>

                </m:properties>
            </content>
        </entry>
    </feed>
</m:inline>
</link>

<category term="PROD_RECO_RUNTIME_SRV.RecommendationScenario"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
<content type="application/xml">
    <m:properties>
        <d:UserId m:type="Edm.String"></d:UserId>
        <d:ExternalTracking m:type="Edm.Boolean">>false</
d:ExternalTracking>
    </m:properties>
    </content>
</entry>

```

## HTTP Post Response Payload in XML Encoding

```

<?xml version="1.0" encoding="utf-8"?>
<entry xml:base="https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/" xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:d="http://
schemas.microsoft.com/ado/2007/08/dataservices">
    <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(UserId='',UserType='')</id>
    <title type="text">RecommendationScenarios(UserId='',UserType='')</title>
    <updated>2019-05-27T14:51:16Z</updated>
    <category term="PROD_RECO_RUNTIME_SRV.RecommendationScenario" scheme="http://
schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
    <link href="RecommendationScenarios(UserId='',UserType='')" rel="self"
title="RecommendationScenario"/>
    <link href="RecommendationScenarios(UserId='',UserType='')/ScenarioHashes"
rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/
ScenarioHashes" type="application/atom+xml;type=feed" title="ScenarioHashes">
        <m:inline>
            <feed xml:base="https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/
odata/sap/PROD_RECO_RUNTIME_SRV/">
                <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(UserId='',UserType='')/
ScenarioHashes</id>
                <title type="text">ScenarioHashes</title>
                <updated>2019-05-27T14:51:16Z</updated>
                <author>
                    <name/>
                </author>
                <link href="RecommendationScenarios(UserId='',UserType='')/"
ScenarioHashes" rel="self" title="ScenarioHashes"/>
            </feed>
        </m:inline>
    </entry>
    <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/ScenarioHashes('SAMPLE_TOPN_SCENARIO')</id>

```



```

<title type="text">ScenarioHashes('SAMPLE_TOPN_SCENARIO')</
title>
  <updated>2019-05-27T14:51:16Z</updated>
  <category term="PROD_RECO_RUNTIME_SRV.ScenarioHash"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <link href="ScenarioHashes('SAMPLE_TOPN_SCENARIO') "
rel="self" title="ScenarioHash"/>
  <content type="application/xml">
    <m:properties>
      <d:ScenarioId>SAMPLE_TOPN_SCENARIO</d:ScenarioId>
      <d:HashId>85075A5D2C02E7FEAC9AE8090C798F95</d:HashId>
      <d:ExpiresOn>2019-05-28T08:40:44.1074560</
d:ExpiresOn>
      <d:ResultScope>G</d:ResultScope>
    </m:properties>
  </content>
</entry>
</feed>
</m:inline>
</link>
<link href="RecommendationScenarios(UserId='',UserType='')/Scenarios"
rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Scenarios"
type="application/atom+xml;type=feed" title="Scenarios">
  <m:inline>
    <feed xml:base="https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/
odata/sap/PROD_RECO_RUNTIME_SRV/">
      <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(UserId='',UserType='')/
Scenarios</id>
      <title type="text">Scenarios</title>
      <updated>2019-05-27T14:51:16Z</updated>
      <author>
        <name/>
      </author>
      <link href="RecommendationScenarios(UserId='',UserType='')/
Scenarios" rel="self" title="Scenarios"/>
    </feed>
  </m:inline>
</link>
<link href="RecommendationScenarios(UserId='',UserType='')/ContextParams"
rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/
ContextParams" type="application/atom+xml;type=feed" title="ContextParams"/>
  <link href="RecommendationScenarios(UserId='',UserType='')/ResultObjects"
rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/
ResultObjects" type="application/atom+xml;type=feed" title="ResultObjects">
    <m:inline>
      <feed xml:base="https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/
odata/sap/PROD_RECO_RUNTIME_SRV/">
        <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/RecommendationScenarios(UserId='',UserType='')/
ResultObjects</id>
        <title type="text">ResultObject</title>
        <updated>2019-05-27T14:51:16Z</updated>
        <author>
          <name/>
        </author>
        <link href="RecommendationScenarios(UserId='',UserType='')/
ResultObjects" rel="self" title="ResultObject"/>
      <entry>
        <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS_PRODU
CT',ResultObjectId='01_TTPRODW2')</id>
        <title
type="text">ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_
HYBRIS_PRODUCT',ResultObjectId='01_TTPRODW2')</title>
        <updated>2019-05-27T14:51:16Z</updated>

```

```

        <category term="PROD_RECO_RUNTIME_SRV.ResultObject"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
        <link
href="ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS
_PRODUCT',ResultObjectId='01_TTPRODW2')" rel="self" title="ResultObject"/>
        <content type="application/xml">
            <m:properties>
                <d:ScenarioId>SAMPLE_TOPN_SCENARIO</d:ScenarioId>
                <d:ResultObjectType>SAP_HYBRIS_PRODUCT</
d:ResultObjectType>
                <d:ResultObjectId>01_TTPRODW2</d:ResultObjectId>
                <d:ResultObjectScore>0.82353</d:ResultObjectScore>
            </m:properties>
        </content>
    </entry>
    <entry>
        <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS_PRODU
CT',ResultObjectId='01_TTACPRODW2')</id>
        <title
type="text">ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_
HYBRIS_PRODUCT',ResultObjectId='01_TTACPRODW2')</title>
        <updated>2019-05-27T14:51:16Z</updated>
        <category term="PROD_RECO_RUNTIME_SRV.ResultObject"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
        <link
href="ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS
_PRODUCT',ResultObjectId='01_TTACPRODW2')" rel="self" title="ResultObject"/>
        <content type="application/xml">
            <m:properties>
                <d:ScenarioId>SAMPLE_TOPN_SCENARIO</d:ScenarioId>
                <d:ResultObjectType>SAP_HYBRIS_PRODUCT</
d:ResultObjectType>
                <d:ResultObjectId>01_TTACPRODW2</d:ResultObjectId>
                <d:ResultObjectScore>0.41176</d:ResultObjectScore>
            </m:properties>
        </content>
    </entry>
    <entry>
        <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS_PRODU
CT',ResultObjectId='02_TTACPRODW2')</id>
        <title
type="text">ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_
HYBRIS_PRODUCT',ResultObjectId='02_TTACPRODW2')</title>
        <updated>2019-05-27T14:51:16Z</updated>
        <category term="PROD_RECO_RUNTIME_SRV.ResultObject"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
        <link
href="ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS
_PRODUCT',ResultObjectId='02_TTACPRODW2')" rel="self" title="ResultObject"/>
        <content type="application/xml">
            <m:properties>
                <d:ScenarioId>SAMPLE_TOPN_SCENARIO</d:ScenarioId>
                <d:ResultObjectType>SAP_HYBRIS_PRODUCT</
d:ResultObjectType>
                <d:ResultObjectId>02_TTACPRODW2</d:ResultObjectId>
                <d:ResultObjectScore>0.41176</d:ResultObjectScore>
            </m:properties>
        </content>
    </entry>
    <entry>
        <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS_PRODU
CT',ResultObjectId='07_TTACPRODW2')</id>

```

```

        <title
type="text">ResultObject (ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_
HYBRIS_PRODUCT',ResultObjectId='07_TTACPROD2')</title>
        <updated>2019-05-27T14:51:16Z</updated>
        <category term="PROD_RECO_RUNTIME_SRV.ResultObject"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
        <link
href="ResultObject (ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS
_PRODUCT',ResultObjectId='07_TTACPROD2')" rel="self" title="ResultObject"/>
        <content type="application/xml">
        <m:properties>
        <d:ScenarioId>SAMPLE_TOPN_SCENARIO</d:ScenarioId>
        <d:ResultObjectType>SAP_HYBRIS_PRODUCT</
d:ResultObjectType>
        <d:ResultObjectId>07_TTACPROD2</d:ResultObjectId>
        <d:ResultObjectScore>0.41176</d:ResultObjectScore>
        </m:properties>
        </content>
    </entry>
    <entry>
        <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject (ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS_PRODU
CT',ResultObjectId='03_TTPROD2')</id>
        <title
type="text">ResultObject (ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_
HYBRIS_PRODUCT',ResultObjectId='03_TTPROD2')</title>
        <updated>2019-05-27T14:51:16Z</updated>
        <category term="PROD_RECO_RUNTIME_SRV.ResultObject"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
        <link
href="ResultObject (ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS
_PRODUCT',ResultObjectId='03_TTPROD2')" rel="self" title="ResultObject"/>
        <content type="application/xml">
        <m:properties>
        <d:ScenarioId>SAMPLE_TOPN_SCENARIO</d:ScenarioId>
        <d:ResultObjectType>SAP_HYBRIS_PRODUCT</
d:ResultObjectType>
        <d:ResultObjectId>03_TTPROD2</d:ResultObjectId>
        <d:ResultObjectScore>0.41176</d:ResultObjectScore>
        </m:properties>
        </content>
    </entry>
    <entry>
        <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject (ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS_PRODU
CT',ResultObjectId='02_TTPROD2')</id>
        <title
type="text">ResultObject (ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_
HYBRIS_PRODUCT',ResultObjectId='02_TTPROD2')</title>
        <updated>2019-05-27T14:51:16Z</updated>
        <category term="PROD_RECO_RUNTIME_SRV.ResultObject"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
        <link
href="ResultObject (ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS
_PRODUCT',ResultObjectId='02_TTPROD2')" rel="self" title="ResultObject"/>
        <content type="application/xml">
        <m:properties>
        <d:ScenarioId>SAMPLE_TOPN_SCENARIO</d:ScenarioId>
        <d:ResultObjectType>SAP_HYBRIS_PRODUCT</
d:ResultObjectType>
        <d:ResultObjectId>02_TTPROD2</d:ResultObjectId>
        <d:ResultObjectScore>0.41176</d:ResultObjectScore>
        </m:properties>
        </content>
    </entry>
</entry>

```

```

        <id>https://[SAP_MARKETING_CLOUD_SYSTEM]/sap/opu/odata/sap/
PROD_RECO_RUNTIME_SRV/
ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS_PRODU
CT',ResultObjectId='07_TTPRODW2')</id>
        <title
type="text">ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_
HYBRIS_PRODUCT',ResultObjectId='07_TTPRODW2')</title>
        <updated>2019-05-27T14:51:16Z</updated>
        <category term="PROD_RECO_RUNTIME_SRV.ResultObject"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
        <link
href="ResultObject(ScenarioId='SAMPLE_TOPN_SCENARIO',ResultObjectType='SAP_HYBRIS
_PRODUCT',ResultObjectId='07_TTPRODW2')" rel="self" title="ResultObject"/>
        <content type="application/xml">
            <m:properties>
                <d:ScenarioId>SAMPLE_TOPN_SCENARIO</d:ScenarioId>
                <d:ResultObjectType>SAP_HYBRIS_PRODUCT</
d:ResultObjectType>
                <d:ResultObjectId>07_TTPRODW2</d:ResultObjectId>
                <d:ResultObjectScore>0.41176</d:ResultObjectScore>
            </m:properties>
        </content>
    </entry>
</feed>
</m:inline>
</link>
<content type="application/xml">
    <m:properties>
        <d:UserId/>
        <d:UserType/>
        <d:ExternalTracking>>false</d:ExternalTracking>
    </m:properties>
</content>
</entry>

```

### 5.6.2.1 Value Help Enabling Entities

Entities that enable you to choose recommendation scenario and item source type parameters from a value help.

The PROD\_RECO\_RUNTIME\_SRV OData service enables customer channels to receive recommendations generated by Recommendation. The RecommendationRecoScenarios and ItemSourceTypes entities enable customer channels to choose ScenarioID, LeadingObjectType, or BasketObjectType parameters from a value help.

### Prerequisites

You have assigned the *Marketing - Recommendation Integration* communication scenario to your communication user in *Maintain Communication Users*.

## ProductRecoScenarios Entity

**Root URL:** `https://<Server>:<Port>/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ProductRecoScenarios`

**Request Mode:** GET

### ProductRecoScenario Entity Parameters

The following table contains the parameters of the `ProductRecoScenario` entity:

Property	Description	Edm Core Type	Max Length	Key
ScenarioId	The ID of the scenario.	Edm.String	50	TRUE
ScenarioDescription	The description of the scenario.	Edm.String	255	FALSE
Language	The language of the scenario description.	Edm.String	30	FALSE

## ItemSourceTypes Entity

**Root URL:** `https://<Server>:<Port>/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ItemSourceTypes`

**Request Mode:** GET

### ItemSourceTypes Entity Parameters

The following table contains the parameters of the `ItemSourceTypes` entity:

Property	Description	Edm Core Type	Max Length	Key
ItemSourceId	The ID of the item source.	Edm.String	2	TRUE
ItemSourceTypeDescription	The description of the item source type.	Edm.String	255	FALSE
ItemSourceType	The object type of the item source.	Edm.String	30	FALSE

## 5.6.3 External Recommendations

Use the public OData API `API_MKT_EXTERNAL_RECMDN_SRV` to upload (import) recommendations from external sources.

### Overview

This OData API provides functionality to import product and offer recommendations that have been calculated using external tools. External recommendations can be used in the recommendation processes for products and offers.

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;server&gt;:&lt;port&gt;/sap/opu/odata/sap/API_MKT_EXTERNAL_RECMDN_SRV</code>
<b>Service Metadata URI</b>	<code>https://&lt;server&gt;:&lt;port&gt;/sap/opu/odata/sap/API_MKT_EXTERNAL_RECMDN_SRV/\$metadata</code>
<b>Authorizations</b>	A copy of the following business catalog role is required: <code>SAP_COM_CSR_0300</code> . Read-only access is provided using the <code>SAP_BCR_CEC_MKT_API_EXTRECO_PC</code> business catalog role.
<b>Communication Scenario ID</b>	<code>SAP_COM_0300</code>
Component for Incidents	CEC-MKT-PRI (Recommendation)

**i Note**

Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

### Support of OData Features

Feature	Support
\$top query option	When performing GET calls on entity sets for this API, the \$top query option is mandatory to restrict resource consumption in the system.

### 5.6.3.1 Basic Concepts

The public `API_MKT_EXTERNAL_RECMDN_SRV` OData API upload (import) recommendations from external sources.

## Processing Information

The API can perform all supported operations either as a single operation or as a batch request. Batch requests are submitted as a single HTTP `POST` request to the `$batch` endpoint of a service as described in [\[OData-URI ↗\]](#). The batch request must contain a header parameter `content-type`, specifying the value `multipart/mixed` and `boundary=batch`.

### i Note

There are no eligibility or validity checks performed on offers. Offer content is determined using the following value help entities of the `CUAN_OFFER_DISCOVERY_SRV` API:

- Communication Medium
- Content Type
- Content Position
- Marketing Area
- Languages
- Coupon

For more information, see [Discover Offers \[page 1008\]](#).

## Error Messages

By default, data processing for external recommendation is synchronous and an OK response or error messages are returned as soon as data processing finishes. If the OData service isn't accessible, for example due to missing authorization, or because the system isn't available, a corresponding HTTP status code is returned.

You can change the default setting to asynchronous with the `Sap-Cuan-AsynchronousProcessing` property. Using asynchronous processing, an OK response is returned almost immediately. If data uploads contain severe errors, such as parse or format errors, they produce an error message and the data is placed in a staging area, where it's then further processed. To view the processing status and check for errors or success messages when data is processed asynchronously, you must launch the *Import Monitor* app. If errors occur, you can restart or discard the import in the app.

For more information, see [Structure of OData Service API\\_MKT\\_EXTERNAL\\_RECMDN\\_SRV \[page 956\]](#).

## Related Information

[Import Monitor \[page 404\]](#)

[External Algorithms](#)

## 5.6.3.2 Structure of OData Service API\_MKT\_EXTERNAL\_RECMDN\_SRV

This document describes the structure of the Public OData API service `API_MKT_EXTERNAL_RECMDN_SRV`. Be sure to read the Basic Concepts topic before you start.

### Request Header

The request header contains the additional header fields listed in the table.

Property	Example	Description	Edm Core Type	Max. Length	Mandatory
<code>Sap-cuan-asynchronousProcessing</code>	X	This property enables uploaded data to be processed asynchronously. For more information, see <a href="#">Basic Concepts [page 954]</a> .	Edm.Boolean	n/a	No
<code>Sap-cuan-SourceSystemId</code>	HYBRIS	This free text field identifies the source system.	Edm.String	255	No

**i Note**

This property is only useful if property `Sap-cuan-AsynchronousProcessing` is enabled.

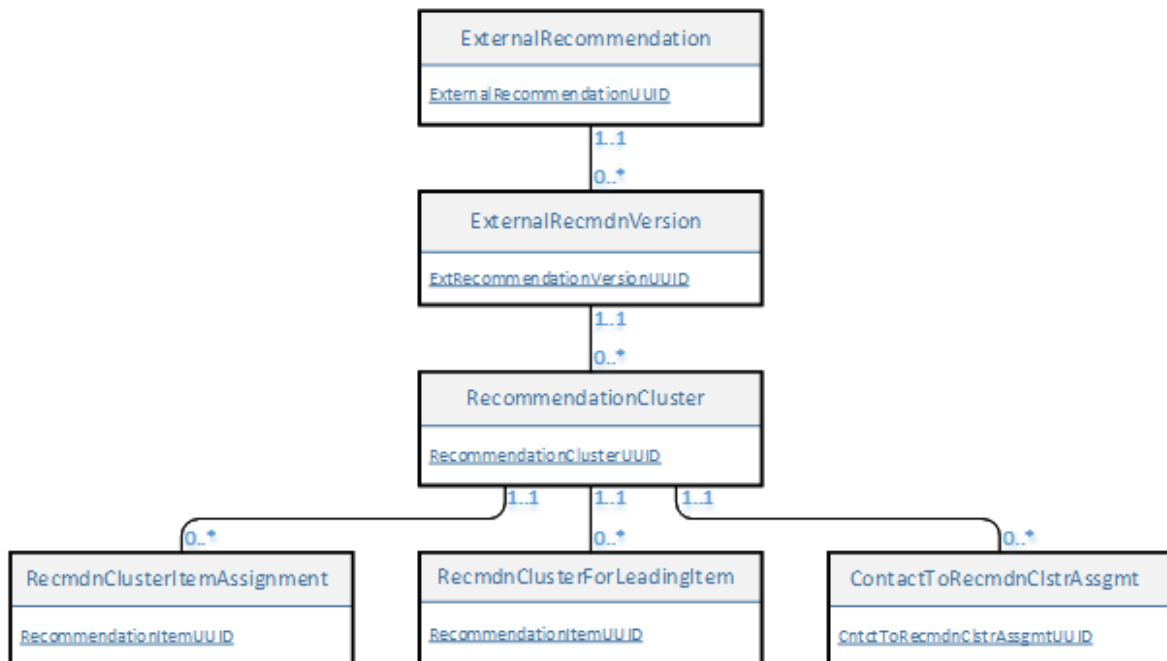


Property	Example	Description	Edm Core Type	Max. Length	Mandatory
Sap-cuan-SourceSystemType	EXTERNAL	This free text field identifies the source system type.	Edm.String	20	No

**Note**

This property is only useful if property Sap-cuan-AsynchronousProcessing is enabled.

## Entity Data Model



## Entity Sets

The External Recommendations OData API provides the following entity sets:

Entity Set	Description	Path
ExternalRecommendations	This entity set contains a list of external recommendation data sets.	/ExternalRecommendations
ExternalRecmdnVersions	This entity set contains a list of versions of an external recommendation data set.	/ExternalRecommendations(guid'<ExternalRecommendationUUID >')/to_ExternalRecmdnVersion
RecommendationClusters	This entity set contains a list of clusters of a version of an external recommendation data set.	/ExternalRecmdnVersions(guid'<ExtRecommendationVersionUUID>')/to_RecommendationCluster
ContactToRecmdnCistrAssgmts	This entity set contains a list of contacts that are assigned to a recommendation cluster.	/RecommendationClusters(guid'<RecommendationClusterUUID>')/to_CntctToRecmdnCistrAssgmt
RecmdnClusterItemAssignments	This entity set contains a list of ranked recommendation result items (products, offers).	/RecommendationClusters(guid'<RecommendationClusterUUID>')/to_RecmdnCistrItemAssgmt
RecmdnClusterForLeadingItems	This entity set contains a list of tuples of recommendation leading items (products, product categories, items of interest, offers) and result items (products, offers).	/RecommendationClusters(guid'<RecommendationClusterUUID>')/to_RecmdnCistrLeadingitm

### ExternalRecommendations

The `ExternalRecommendations` entity set represents the header of an external recommendations data set. Its data controls the behavior of the dependent subnodes in the data model.

**Resource Path:** `/ExternalRecommendations`

You can perform the following operations on the `ExternalRecommendations` entity set:

HTTP Method	Operation	URI
GET	Get a list of external recommendations. This method supports standard OData parameters, such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , <code>\$orderby</code> and <code>\$expand</code> .	<code>/ExternalRecommendations?\$top=1</code>

HTTP Method	Operation	URI
	Get the details of a specific external recommendation	<code>/ExternalRecommendations(guid '&lt;ExternalRecommendationUUID&gt;')</code>
POST	Create an external recommendation data set	<code>/ExternalRecommendations</code>
PATCH	Update an external recommendation data set	<code>/ExternalRecommendations(guid '&lt;ExternalRecommendationUUID&gt;')</code>

More information about the fields in the `ExternalRecommendations` entity set:

- The field `ExternalRecommendation` is a 32 character free-text field that identifies the data set.
- The field `ExternalRecommendationName` is a 120 character free-text field that describes the data set.
- The field `ExternalRecommendationType` is a 2 character field that must use the values '01' or '02'. This field controls whether the data set contains a simple ranked list of recommendation result items (see the entity set `RecmdnClusterItemAssignments`) or a list of ranked leading and result item tuples (see the entity set `RecmdnClusterForLeadingItems`).
- The field `RecommendationResultItemType` controls whether the recommendation result items are products (value '13') or offer content items (value '15'). Offers aren't subject to any eligibility or validity checks.

### Note

During the import process of external recommendations, enclose multiple changes within a single changeset. Doing so ensures a higher throughput in the import process.

## ExternalRecmdnVersions

The `ExternalRecmdnVersions` entity set represents a version of an external recommendations data set. Several versions of each data set can be uploaded to the system in parallel. The recommendation engine determines which version is used at runtime based on the value of the field `ValidityStartDateTime`. This timestamp controls the point in time at which the data within a version becomes active. The recommendation engine uses the version for which the value of `ValidityStartDateTime` is in the past and closest to the current point in time. If two versions have the same value for `ValidityStartDateTime`, the version with the most recent `LastChangedDate` timestamp is used at runtime.

The `IsDeleted` field indicates that an internal cleanup job deleted an obsolete data set version. The following criteria must be met for the data set to be deemed obsolete and to enable the internal job to delete the version:

- The version isn't active and won't be active later. For example, another version with a more recent `ValidityStartDateTime` is active.
- The version isn't used by a recommendation model that has a status of `Active` or `Activation Pending`. For more information, see [Understanding Model Statuses](#).

**Resource Path:** `/ExternalRecmdnVersions`

You can perform the following operations on the `ExternalRecmdnVersions` entity set:

HTTP Method	Operation	URI
GET	Get a list of external recommendation versions for an external recommendation. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , <code>\$orderby</code> and <code>\$expand</code> .	<pre>/ ExternalRecommendations(   guid'&lt;ExternalRecommendationUUID&gt;')/ to_ExternalRecmdnVersion</pre>
	Get the details of a specific external recommendation version	<pre>/ ExternalRecmdnVersions(   guid'&lt;ExtRecommendationVersionUUID&gt;')</pre>
POST	Create an external recommendation version for an external recommendation data set	<pre>/ ExternalRecommendations(   guid'&lt;ExternalRecommendationUUID&gt;')/ to_ExternalRecmdnVersion</pre>
PATCH	Update an external recommendation version	<pre>/ ExternalRecmdnVersions(   guid'&lt;ExtRecommendationVersionUUID&gt;')</pre>

## RecommendationClusters

The `RecommendationClusters` entity set represents a cluster of recommendation data in a version of an external recommendation data set. A cluster in the recommendation contains a list to contacts assigned to this cluster (entity set `ContactToRecmdnClstrAssgmts`), and either a list of ranked recommendation items (entity set `RecmdnClusterItemAssignments`) or a list of ranked leading and result item tuples (entity set `RecmdnClusterForLeadingItems`). See also the field [ExternalRecommendationType](#) in the `ExternalRecommendations` entity type.

**Resource Path:** `/RecommendationClusters`

You can perform the following operations on the `RecommendationClusters` entity set:

HTTP Method	Operation	URI
GET	Get a list of recommendation clusters for a recommendation version. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , <code>\$orderby</code> and <code>\$expand</code> .	<pre>/ ExternalRecmdnVersions(   guid'&lt;ExtRecommendationVersionUUID&gt;')/ to_RecommendationCluster</pre>

HTTP Method	Operation	URI
	Get the details of a specific recommendation cluster	<code>/ RecommendationClusters (guid'&lt;RecommendationClusterUUID&gt;')</code>
POST	Create a recommendation cluster	<code>/ ExternalRecmdnVersions (guid'&lt;ExtRecommendationVersionUUID&gt;') / to_RecommendationCluster</code>
PATCH	Update a recommendation cluster	<code>/ RecommendationClusters (guid'&lt;RecommendationClusterUUID&gt;')</code>
DELETE	Delete a recommendation cluster	<code>/ RecommendationClusters (guid'&lt;RecommendationClusterUUID&gt;')</code>

One cluster in the list of clusters of a version can be a fallback cluster. If the contact for which the recommendation was called was either anonymous or couldn't be found in any other cluster, the recommendation runtime uses the recommendation items from this cluster. Only one fallback cluster is allowed. It's identified by the *IsFallbackRecmdnCluster* field with the value `true`. Fallback clusters don't need any contacts in the `ContactToRecmdnClstrAssgmts` entity set.

## ContactToRecmdnClstrAssgmts

The `ContactToRecmdnClstrAssgmts` entity set represents the assignment of interaction contacts to a recommendation cluster.

**Resource Path:** `/ContactToRecmdnClstrAssgmts`

You can perform the following operations on the `ContactToRecmdnClstrAssgmts` entity set:

HTTP Method	Operation	URI
GET	Get a list of interaction contacts in a recommendation cluster. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , <code>\$orderby</code> and <code>\$expand</code> .	<code>/ RecommendationClusters (guid'&lt;RecommendationClusterUUID&gt;') / to_CntctToRecmdnClstrAssgmt</code>
	Get the details of a specific contact to cluster assignment	<code>/ ContactToRecmdnClstrAssgmts (guid'&lt;CntctToRecmdnClstrAssgmtUUID&gt;')</code>

HTTP Method	Operation	URI
POST	Create a contact to cluster assignment	<pre>/ RecommendationClusters(guid'&lt;RecommendationClusterUUID&gt;'/ to_CntctToRecmdnClstrAssgmt</pre>
DELETE	Delete a contact to cluster assignment	<pre>/ ContactToRecmdnClstrAssgmts(guid'&lt;CntctToRecmdnClstrAssgmtUUID&gt;')</pre>

When creating a `ContactToRecmdnClstrAssgmts` entity using a POST call, the interaction contact is provided using the fields `InteractionContactId` and `InteractionContactOrigin`. The system determines the corresponding `InteractionContactUUID` automatically.

The field `RecommendationCluster` is inherited from the `RecommendationClusters` entity. The field `RecmdnInteractionContactType` is determined internally.

### RecmdnClusterItemAssignments

The `RecmdnClusterItemAssignments` entity set represents the assignment of ranked result items to a recommendation cluster.

**Resource Path:** `/RecmdnClusterItemAssignments`

You can perform the following operations on the `RecmdnClusterItemAssignments` entity set:

HTTP Method	Operation	URI
GET	Get a list of recommendation items in a recommendation cluster. This method supports standard OData parameters such as <code>\$filter</code> , <code>\$select</code> , <code>\$top</code> , <code>\$skip</code> , <code>\$count</code> , <code>\$inlinecount</code> , <code>\$orderby</code> and <code>\$expand</code> .	<pre>/ RecommendationClusters(guid'&lt;RecommendationClusterUUID&gt;'/ to_RecmdnClstrItemAssgmt</pre>
	Get the details of a specific recommendation item to cluster assignment	<pre>/ RecmdnClusterItemAssignments(guid'&lt;RecommendationItemUUID&gt;')</pre>
POST	Create a recommendation item to cluster assignment	<pre>/ RecommendationClusters(guid'&lt;RecommendationClusterUUID&gt;'/ to_RecmdnClstrItemAssgmt</pre>

HTTP Method	Operation	URI
DELETE	Delete a recommendation item to cluster assignment	<code>/RecmdnClusterItemAssignments(guid'&lt;RecommendationItemUUID&gt;')</code>

When creating a recommendation item to cluster assignment using a POST call, the field *RecommendationResultItemType* is automatically inherited from the corresponding ExternalRecommendations entity. The field *RecommendationCluster* is inherited from the RecommendationClusters entity.

If the *RecommendationResultItemType* is "13" for products, recommendation result items can be identified using the combination of *RecommendationResultItem* and *RecommendationResultItemOrigin* or using *RecommendationResultItemUUID*. If all values are provided in the POST call, they're cross-checked.

If the *RecommendationResultItemType* is "15" for SAP Marketing Cloud Offer Content, result items are identified using *RecommendationResultItemUUID*. For more information, see [Read Offers \[page 1002\]](#).

The *RecommendationItemScore* value must be greater than 0.

## RecmdnClusterForLeadingItems

The *RecmdnClusterForLeadingItems* entity set represents the assignment of ranked leading and result item tuples to a recommendation cluster.

**Resource Path:** `/RecmdnClusterForLeadingItems`

You can perform the following operations on the *RecmdnClusterForLeadingItems* entity set:

HTTP Method	Operation	URI
GET	Get a list of recommendation leading item/result item tuples in a recommendation cluster. This method supports standard OData parameters such as \$filter, \$select, \$top, \$skip, \$count, \$inlinecount, \$orderby and \$expand.	<code>/RecommendationClusters(guid'&lt;RecommendationClusterUUID&gt;'/to_RecmdnClstrLeadingItm</code>
	Get the details of a specific recommendation leading item/result item tuple to cluster assignment	<code>/RecmdnClusterForLeadingItems(guid'&lt;RecommendationItemUUID&gt;')</code>
POST	Create a recommendation leading item/result item tuple to cluster assignment	<code>/RecommendationClusters(guid'&lt;RecommendationClusterUUID&gt;'/to_RecmdnClstrLeadingItm</code>
DELETE	Delete a leading item/result item tuple to cluster assignment	<code>/RecmdnClusterForLeadingItems(guid'&lt;RecommendationItemUUID&gt;')</code>

When creating a recommendation item to cluster assignment using a POST call, the field *RecommendationResultItemType* is automatically inherited from the corresponding ExternalRecommendations entity. The field *RecommendationCluster* is inherited from the RecommendationClusters entity.

If the RecommendationResultItemType is "13" for products, recommendation result items can be identified using the combination of *RecommendationResultItem* and *RecommendationResultItemOrigin* or using *RecommendationResultItemUUID*. If all values are provided in the POST call, they're cross-checked.

If the *RecommendationResultItemType* is "15" for SAP Marketing Cloud Offer Content, result items are identified using *RecommendationResultItemUUID*. For more information, see [Read Offers \[page 1002\]](#).

The *RecommendationItemScore* value must be greater than 0.

The following leading item types are supported:

- "11" for items of interest  
The POST call expects *RecommendationLeadingItem* and/or *RecommendationLeadingItemUUID* which are checked for correctness and eventually converted into each other.
- "13" for products  
The POST call expects *RecommendationLeadingItem* and *RecommendationLeadingItemOrigin* or *RecommendationLeadingItemUUID* which are checked for correctness and eventually converted into each other.
- "15" for SAP Marketing Cloud Offer Content  
The POST call expects *RecommendationLeadingItemUUID* which is checked for correctness.
- "19" for product categories  
The POST call expects the category UUID in *RecommendationLeadingItemUUID* or the category ID in *RecommendationLeadingItem* and hierarchy ID in *RecommendationLeadingItemOrigin*.

#### **General comment on create (POST):**

The system generates the key for the entities. The POST payload doesn't provide the key.

#### **General comment on update (PATCH):**

Certain fields (for example *RecommendationResultItemType* in the ExternalRecommendations entity set) can be updated while there are no child entities. As soon as child nodes exist, the fields can no longer be updated, and update requests return an error when trying to update these fields.

Entities from the ContactToRecmdnClstrAssgmts, RecmdnClusterItemAssignments, RecmdnClusterForLeadingItems entity sets, can only be created or deleted. The entities can't be updated.

#### **General comment on error handling:**

If errors occur when using the default synchronous data processing, they're directly reported using the response to the OData request. Any errors that occur during create or update operations are also recorded in the application log. The application log object is "Recommendations" (PRI) with the subobject "External Recommendations" (EXT\_DATA). A system administrator can analyze entries in the application log in the corresponding SAP Fiori UI.

If errors occur when using the asynchronous data processing feature, you must launch the *Import Monitor* app to view the processing status and check for errors or success messages. If errors occur, you can restart or discard the import using the app.



## 5.6.3.3 Payload Examples

The following are examples of how you can use the External Recommendations API.

### Example 1

External offer recommendation with one contact, the tuple of a leading product, and an offer (content) result item. The leading product and the offer (content) are identified by their UUIDs.

#### Sample Code

```
{
  "ExternalRecommendation": "EXTERNAL_OFFER",
  "ExternalRecommendationName": "External Offer",
  "ExternalRecommendationType": "02",
  "RecommendationResultItemType": "15",
  "to_ExternalRecmdnVersion": [
    {
      "ExtRecmdnExternalVersion": "EXTERNAL_OFFER_V1",
      "ValidityStartDateTime": "2017-02-12T09:00:00.0000000",
      "to_RecommendationCluster": [
        {
          "RecommendationCluster":
            "EXTERNAL_OFFER_C1",
          "IsFallbackRecmdnCluster": false,
          "to_CntctToRecmdnClstrAssgmt": [
            {
              "InteractionContactId":
                "test_user@test.test",
              "InteractionContactOrigin":
                "EMAIL"
            }
          ],
          "to_RecmdnClstrLeadingItm": [
            {
              "RecommendationLeadingItemUUID": "2DA602C04DD61C1516006102FF7A1A39",
              "RecommendationLeadingItemType": "13",
              "RecommendationResultItemUUID": "941882831C7D1ED88AF4C61864181AE7",
              "RecommendationItemScore": 1.0
            }
          ]
        }
      ]
    }
  ]
}
```

## Example 2

External offer recommendation with one contact, two tuples of leading products, and offer (content) result items. The offer (content) result item is identified by its UUID. The leading product is identified once by its origin and ID, and once by its UUID.

### Sample Code

```
{
  "ExternalRecommendation": "EXTERNAL_OFFER",
  "ExternalRecommendationName": "External Offer",
  "ExternalRecommendationType": "02",
  "RecommendationResultItemType": "15",
  "to_ExternalRecmdnVersion": [
    {
      "ExtRecmdnExternalVersion": "EXTERNAL_OFFER_V1",
      "ValidityStartDateTime": "2017-02-15T09:00:00.0000000",
      "to_RecommendationCluster": [
        {
          "RecommendationCluster":
            "EXTERNAL_OFFER_C1",
          "IsFallbackRecmdnCluster": false,
          "to_CntctToRecmdnClstrAssgmt": [
            {
              "InteractionContactId":
                "test_user@test.test",
              "InteractionContactOrigin":
                "EMAIL"
            }
          ],
          "to_RecmdnClstrLeadingItm": [
            {
              "RecommendationLeadingItem":
                "P-12345",
              "RecommendationLeadingItemOrigin": "SAP_PRODUCT",
              "RecommendationLeadingItemType": "13",
              "RecommendationResultItemUUID": "941882831C7D1ED88AF4C61864181AE7",
              "RecommendationItemScore": 1.0
            }
          ],
          "RecommendationLeadingItemUUID": "31A602C04DD61C1516006102FF7A1A39",
          "RecommendationLeadingItemType": "13",
          "RecommendationResultItemUUID": "941882831C7D1EE888AF07DF4D1A0094",
          "RecommendationItemScore": 1.0
        }
      ]
    }
  ]
}
```

### Example 3

External offer recommendation without contacts (uses the fallback cluster), two tuples of leading products, and offer (content) result items. The leading products and the offer (content) result item are identified by their UUIDs.

#### Sample Code

```
{
  "ExternalRecommendation": "EXTERNAL_OFFER",
  "ExternalRecommendationName": "External Offer",
  "ExternalRecommendationType": "02",
  "RecommendationResultItemType": "15",
  "to_ExternalRecmdnVersion": [
    {
      "ExtRecmdnExternalVersion": "EXTERNAL_OFFER_V1",
      "ValidityStartDateTime": "2017-02-15T09:00:00.000000",
      "to_RecommendationCluster": [
        {
          "RecommendationCluster":
            "EXTERNAL_OFFER_CFB",
          "IsFallbackRecmdnCluster": true,
          "to_RecmdnClstrLeadingItm": [
            {
              "RecommendationLeadingItemUUID": "2DA602C04DD61C1516006102FF7A1A39",
              "RecommendationLeadingItemType": "13",
              "RecommendationResultItemUUID": "941882831C7D1ED88AF4C61864181AE7",
              "RecommendationItemScore": 1.0
            },
            {
              "RecommendationLeadingItemUUID": "31A602C04DD61C1516006102FF7A1A39",
              "RecommendationLeadingItemType": "13",
              "RecommendationResultItemUUID": "941882831C7D1EE888AF07DF4D1A0094",
              "RecommendationItemScore": 1.0
            }
          ]
        }
      ]
    }
  ]
}
```

### Example 4

External offer recommendation without contact (uses the fallback cluster) and a single offer result item identified by its UUID.

#### Sample Code

```
{
  "ExternalRecommendation": "EXTERNAL_OFFER",
  "ExternalRecommendationName": "External Offer",
```

```

"ExternalRecommendationType": "01",
"RecommendationResultItemType": "15",
"to_ExternalRecmdnVersion": [
  {
    "ExtRecmdnExternalVersion": "EXTERNAL_OFFER_V1",
    "ValidityStartDateTime": "2017-11-01T00:00:00.0000000",
    "to_RecommendationCluster": [
      {
        "RecommendationCluster":
"EXTERNAL_OFFER_CFB",
        "IsFallbackRecmdnCluster": true,
        "to_RecmdnClstrItemAssgmt": [
          {
            "RecommendationResultItemUUID": "941882831C7D1ED88AF4C61864181AE7",
            "RecommendationItemScore": 1.0
          }
        ]
      }
    ]
  }
]
}

```

## Example 5

Same as example 4, but the request is redirected to asynchronous processing.

**POST** [https://server:port/sap/opu/odata/sap/API\\_MKT\\_EXTERNAL\\_RECMDN\\_SRV/ExternalRecommendations](https://server:port/sap/opu/odata/sap/API_MKT_EXTERNAL_RECMDN_SRV/ExternalRecommendations)

### Header

#### Sample Code

```

Content-Type:application/json
sap-cuan-asynchronousprocessing: x
sap-cuan-sourcesystemid: bat1
sap-cuan-sourcesystemtype: ERP

```

### Body

#### Sample Code

```

{
  "ExternalRecommendation": "EXTERNAL_OFFER",
  "ExternalRecommendationName": "External Offer",
  "ExternalRecommendationType": "02",
  "RecommendationResultItemType": "15",
  "to_ExternalRecmdnVersion": [
    {
      "ExtRecmdnExternalVersion": "EXTERNAL_OFFER_V1",
      "ValidityStartDateTime": "2017-02-15T09:00:00.0000000",
      "to_RecommendationCluster": [
        {
          "RecommendationCluster":
"EXTERNAL_OFFER_CFB",
          "IsFallbackRecmdnCluster": true,

```

```

        "to_RecmdnClstrLeadingItm": [
            {
                "RecommendationLeadingItemUUID": "2DA602C04DD61C1516006102FF7A1A39",
                "RecommendationLeadingItemType": "13",
                "RecommendationResultItemUUID": "941882831C7D1ED88AF4C61864181AE7",
                "RecommendationItemScore": 1.0
            },
            {
                "RecommendationLeadingItemUUID": "31A602C04DD61C1516006102FF7A1A39",
                "RecommendationLeadingItemType": "13",
                "RecommendationResultItemUUID": "941882831C7D1EE888AF07DF4D1A0094",
                "RecommendationItemScore": 1.0
            }
        ]
    }
}
]
}
}

```

## Example 6

A batch request using asynchronous processing. This sample request includes the following:

- A deletion of a cluster.
- A deep creation of a cluster.
- A patch to a leading item node.

**POST** [https://server:port/sap/opu/odata/sap/API\\_MKT\\_EXTERNAL\\_RECMDN\\_SRV/\\$batch](https://server:port/sap/opu/odata/sap/API_MKT_EXTERNAL_RECMDN_SRV/$batch)

### Header

#### Sample Code

```
Content-Type:multipart/mixed; boundary=batchtest
```

### Body

#### Sample Code

```

--batchtest
Content-Type: multipart/mixed; boundary=changeset_9970-5898-d67d

--changeset_9970-5898-d67d
Content-Type: application/http
Content-Transfer-Encoding: binary

PATCH RecmdnClusterForLeadingItems (guid'6c0b84b7-5523-1ee9-b8a4-
fa24838b7e95') HTTP/1.1
Content-Type: application/json
Content-Length: 168

```

```

sap-cuan-asynchronousprocessing: x
sap-cuan-sourcesystemid: bat1
sap-cuan-sourcesystemtype: ERP

{
  "RecommendationLeadingItem": "TEAMTD_PRD_06_01",
  "RecommendationLeadingItemType": "13",
  "RecommendationLeadingItemOrigin": "SAP_HYBRIS_PRODUCT",
  "RecommendationResultItem": "TEAMTD_PRD_03_02",
  "RecommendationResultItemType": "13",
  "RecommendationResultItemOrigin": "SAP_HYBRIS_PRODUCT",
  "RecommendationItemScore": 9
}

--changeset_9970-5898-d67d
content-type: application/http
content-transfer-encoding: binary

DELETE RecommendationClusters(guid'6c0b84b7-5523-1ed8-bf8d-belac978dc2d')?
saml2=disabled HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en-US
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
Content-Type: application/json
Content-Length: 728
sap-cuan-asynchronousprocessing: x
sap-cuan-sourcesystemid: bat1
sap-cuan-sourcesystemtype: ERP

--changeset_9970-5898-d67d
Content-Type: application/http
Content-Transfer-Encoding: binary

POST ExternalRecmdnVersions(guid'6c0b84b7-5523-1ed8-bf8d-belac92e7c2d')/
to_RecommendationCluster?saml2=disabled HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en-US
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
Content-Type: application/json
Content-Length: 728
sap-cuan-asynchronousprocessing: x
sap-cuan-sourcesystemid: bat1
sap-cuan-sourcesystemtype: ERP

{
  "RecommendationCluster": "100",
  "IsFallbackRecmdnCluster": false,
  "to_CntctToRecmdnClstrAssgmt": [
    {
      "InteractionContactId": "0000045000",
      "InteractionContactOrigin": "SAP_CRM_BUPA"
    }
  ],
  "to_RecmdnClstrItemAssgmt": [
    {
      "RecommendationResultItem": "0000000000000001510",
      "RecommendationResultItemOrigin": "SAP_ERP_MATNR",
      "RecommendationItemScore": 1
    }
  ]
}

```

```
--changeset_9970-5898-d67d--  
--batchtest--
```

## 5.6.4 Recommendations Interaction Data

OData service (PROD\_RECO\_RUNTIME\_SRV) for posting interactions to an SAP HANA database.

The PROD\_RECO\_RUNTIME\_SRV OData service enables host systems to post interactions to an SAP HANA database and then consume the information in a recommendation model. An interaction can be any event performed by a consumer on a Web shop.

### Prerequisites

- You have assigned the *Marketing - Recommendation Integration* communication scenario to your communication user in *Maintain Communication Users*.
- To post interactions, you must call the service using the deep insert functionality of OData. For more information about the deep insert functionality of OData, see <http://www.help.sap.com>. Choose *Technology SAP Gateway*. Choose a release and then *Application Help*. In SAP Library, choose [▶ SAP NetWeaver Gateway Developer Guide ▶ OData Channel ▶ Advanced Features ▶ Deep Insert ▶](#)

### Details of Service Entity

**Root URL:** `https://<Server>:<Port>/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/Interactions`

**Request Mode:** POST

**Entity Data Model:** Interaction

The following table contains the parameters of the Interaction Entity:

Name	Is Key	Edm Core Type	Max Length	Creatable	Updatable	Sortable	Nullable	Filterable	Complex Type Name
Scenario Id	TRUE	Edm.String	50	FALSE	FALSE	FALSE	TRUE	FALSE	n.a.
UserId	TRUE	Edm.String	255	FALSE	FALSE	FALSE	FALSE	FALSE	n.a.

Name	Is Key	Edm Core Type	Max Length	Creatable	Updatable	Sortable	Nullable	Filterable	Complex Type Name
InteractionType	TRUE	Edm.String	20	FALSE	FALSE	FALSE	FALSE	FALSE	n.a.
UserId	FALSE	Edm.String	20	FALSE	FALSE	FALSE	FALSE	FALSE	n.a.
SourceObjectId	FALSE	Edm.String	50	FALSE	FALSE	FALSE	TRUE	FALSE	n.a.
TimeStamp	TRUE	Edm.DateTimeOffset	n.a.	FALSE	FALSE	FALSE	TRUE	FALSE	n.a.

### Parameter Descriptions

- `ScenarioId`  
The recommendation scenario ID represents a model type and related usage information, for example, promotion model type and user type.
- `UserId`  
The ID of the user who performs the interaction, for example, customer ID or contact ID.
- `UserId`  
The type of the user who performs the interaction, for example, `COOKIE_ID` or `SAP_ERP_CONTACT`.
- `InteractionType`  
The interaction type, for example, click through and conversion.
- `SourceObjectId`  
The ID of the session the user performed the interaction in.
- `TimeStamp`  
The coordinated universal time (UTC) stamp of when the interaction happened.

### Entity Data Model: InteractionItems

The following table contains the parameters of the `InteractionItems` entity:

Name	Is Key	Edm Core Type	Max Length	Creatable	Updatable	Sortable	Nullable	Filterable	Complex Type Name
ItemType	TRUE	Edm.String	30	FALSE	FALSE	FALSE	TRUE	FALSE	n.a.
ItemId	TRUE	Edm.String	50	FALSE	FALSE	FALSE	FALSE	FALSE	n.a.
ItemNavUrl	FALSE	Edm.String	1333	FALSE	FALSE	FALSE	FALSE	FALSE	n.a.

### Parameter Descriptions

- `Item_type`  
One of the following standard delivery item types:
  - `SAP_CUAN_PRODUCT`



- SAP\_HYBRIS\_PRODUCT
- CUAN\_PROD\_CATEGORY\_HIERARCHY
- ItemId  
The ID of the item, for example, material number.
- ItemNavUrl  
The url to navigate to the item, for example, `https://<Server>:<Port>/yacceleratorstorefront/electronics/en/Open-Catalogue/Cameras/Digital-Cameras/Digital-Compacts/DSC-N1/p/M-10`

## Example

### Payload in JSON Format:

```
{
  "ScenarioId" : "INT_TEST",
  "UserType" : "COOKIE_ID",
  "UserId" : "ccef655202caec49",
  "InteractionType": "CLICK_THROUGH",
  "TimeStamp": "2015-11-23T01:00:00Z",
  "SourceObjectId": "17FE2EA62DB2154594CC1FCEEB58C691",
  "InteractionItems" : [{
    "ItemType" : "SAP_ERP_MATNR",
    "ItemId" : "M-10",
    "ItemNavUrl" : "https://localhost:9002/yacceleratorstorefront/electronics/en/Open-Catalogue/Cameras/Digital-Cameras/Digital-Compacts/DSC-N1/p/M-10"
  }]
}
```

The HTTP post response does not contain any entity.

## 5.6.5 Import Offers

Use the public OData API `CUAN_OFFER_IMPORT_SRV` to upload (import) offers from external sources.

### i Note

The offer import API supports the importation of offers with assigned object references, such as products, marketing locations and coupons. Furthermore, the service supports basic read functionality to read an imported offer when specifying the offer key (consisting of external id and external origin). It is not possible to query all offers or all imported offers with functionality such as search and filtering. For such usecases, please use the [Read Offers API \[page 1002\]](#). Technically, HTTPS GET operations on entities, such as Offers, ProductAssignments, and TargetGroupAssignments, only return data when providing the fully qualified key.

## Overview

Offer data can be maintained using the corresponding maintenance app in the system, but it can also be imported from other systems using this public OData application programming interface (API). You can use the

OData service `CUAN_OFFER_IMPORT_SRV` to upload (import) external offers and offer content with extensibility, assign dependent objects like marketing locations, products, and product categories and read the offer information. Imported offers are assigned an external reference and origin and initially have the status In Preparation. You cannot change the offer data, but you can, for example, change the contents and the status. The ability to import the offer content entity allows for a complete end-to-end integration without any manual steps in SAP Marketing Cloud.

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/</code> <code>CUAN_OFFER_IMPORT_SRV/</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/</code> <code>CUAN_OFFER_IMPORT_SRV//\$metadata</code>
<b>Authorizations</b>	The following business catalog role is required: <code>SAP_COM_CSR_0020</code>
<b>Communication Scenario ID</b>	<code>SAP_COM_0020</code>
Component for Incidents	CEC-MKT-OFM (Offers)

#### **i Note**

Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

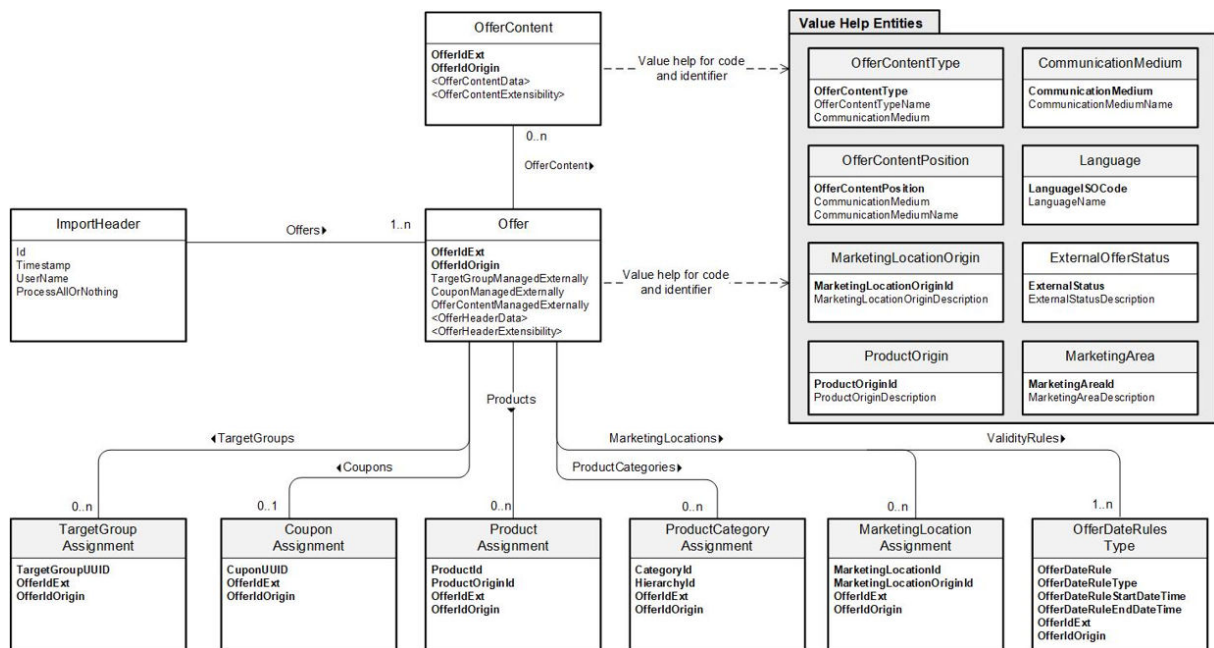
## **Support of OData Features**

<b>Feature</b>	<b>Support</b>
Query options for value help entities	The current implementation of the value help entities supports the following query options, which can be passed as query or path parameters: <ul style="list-style-type: none"> <li>• <code>\$top</code> and <code>\$skip</code></li> <li>• <code>\$select</code></li> <li>• <code>\$orderby</code></li> <li>• <code>\$count</code> and <code>\$inlinecount</code></li> </ul>
Bulk processing using deep-create on entity <code>ImportHeader</code>	The service supports both bulk processing using deep-create on the <code>ImportHeader</code> entity as well as single access to the entities.  If the <code>ImportHeader</code> is used, an application log protocol entry is written.

## Entity Data Model

The following figure shows the entity data model (EDM) for the offer import service:

Service Metadata URI: `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/$metadata?sap-documentation=all`



## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV;v=0003/\$metadata?sap-documentation=all</code>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing Offer Import Details Page</a>	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ul style="list-style-type: none"> <li>1. On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>2. Specify which application you want to use to open the EDMX file type.</li> </ul>
<a href="#">Import Offers API</a>	General access link takes you directly to the <i>Import Offers</i> metadata file. One-time registration or logon is required.

## i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Resources

The service consists of the following types of resources:

- **Read-Only Value Help Entities**

Value help entities to provide values for certain code and identifiers used in other entities. These entities are read-only and support HTTP GET operation to read the values defined in the system.

Resource	Description	Path
ExternalOfferStatus	Read-only value help entity to retrieve all possible external offer status values.	/ExternalOfferStatus
MarketingArea	Read-only value help entity to retrieve all active marketing areas in the system.	/MarketingAreas
ProductOrigin	Read-only value help entity to retrieve all active product origins in the system.	/ProductOrigins
MarketingLocationOrigin	Read-only value help to retrieve all active marketing location origins defined in the system.	/MarketingLocationOrigins
OfferContentType	Read-only value help entity to retrieve all active offer content types in the system.	/OfferContentTypes
CommunicationMedium	Read-only value help entity to retrieve all active communication mediums in the system.	/CommunicationMediums

Resource	Description	Path
Language	Read-only value help entity to retrieve all languages defined in the system.	/Languages
OfferContentPosition	Read-only value help entity to retrieve all positions currently defined in any offer content.	/OfferContentPositions

- **Import Entities**

Entities used for importing offers with their assignment. These entities support multiple operations and are described in more detail in the OData operations sections below.

Resource	Description	Path
<a href="#">ImportHeader [page 978]</a>	Starting point when importing offers using deep-create.	/ImportHeaders
<a href="#">Offer [page 979]</a>	Supports basic CRUD with single read, create, update, and delete of imported offers	/Offers
<a href="#">OfferContent [page 981]</a>	Supports basic CRUD with single read, expanded read of all content for an imported offer, create, update and deletion of offer content.	/OfferContents
<a href="#">OfferDateRulesType [page 982]</a>	Supports basic CRUD with single read, expanded read of all validity rules for an imported offer, create, update and deletion of validity rules.	/OfferDateRules
<a href="#">MarketingLocationAssignment [page 983]</a>	Assign or remove the assignment of marketing locations to offers.	/MarketingLocationAssignments
<a href="#">ProductAssignment [page 984]</a>	Assign or remove the assignment of products to offers.	/ProductAssignments
<a href="#">ProductCategoryAssignment [page 985]</a>	Assign or remove the assignment of product categories to offers.	/ProductCategoryAssignments
<a href="#">TargetGroupAssignment [page 986]</a>	Assign or remove the assignment of target groups to offers.	/TargetGroupAssignments
<a href="#">CouponAssignment [page 987]</a>	Assign or remove the assignment of coupons to offers.	/CouponAssignments

## OData Resource: ImportHeader

Helper entity representing import metadata, such as the importing system.

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/ImportHeaders`

**Permissions:** Business Role `SAP_COM_CSR_0020`

### Operations

Only deep-create and single read is support on this helper resource.

HTTP Method	Operation	URI
POST	Bulk import multiple offers with their assignments to Marketing Locations, Products, and Product Categories and Validity Rules.	<code>/ImportHeaders</code>
GET	Read import header metadata of a specific import run, identified by its Id.	<code>/ImportHeaders(&lt;id&gt;)</code>

### Properties

- **Id:** A technical ID of one import service execution. If no value is provided by the caller, an ID is generated by the system.
- **Timestamp:** Timestamp of the import run. If no value is provided by the caller, a timestamp is generated by the system.
- **UserName:** Name of the user who started the import. If no value is provided by the caller, the system uses the system name.
- **SourceSystemType:** The type of source system (can be freely defined, could be, for example, CRM or ERP).
- **SourceSystemId:** The ID of the source system. Can be freely defined.
- **ImportMode:** Mode in which the offers are imported.

The following status values are available:

- "U" for Upsert: Non-existing offers are created with dependent child objects and already existing ones are updated. For existing ones, the offer header properties not given in the request or not changeable are ignored (works like a PATCH request). We recommend to always send the whole offer with all assigned objects and all offer header properties. For updating single values or assignments, the "Offer" entity or the offer assignment entitysets can be used.  
The offer child entities (like products, marketing locations, etc.) are replaced with the new entered ones (works like a PUT request). Assigned marketing locations or content will be deleted if not in the payload available. For Example: An offer has two validity rules. The request for update contains only offer header data with start date and end date. The two validity rules will be deleted and a new validity rule with the start date and end date of the offer header data will be created.
- "F" (Default) for Full: Only creates offers. Does not update.
- **ProcessAllOrNothing:** If an error occurs, this flag defines, whether all imported offers are discarded (response code 201) or only the ones that contain an error (response code 400, the error messages can be found in the response header). If the import mode is "U", the flag is always considered to have the value "true". If the import mode is "F", the flag can have values "true" and "false" (default).

## OData Resource: Offer

Represents an imported offer.

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/Offers`

### Operations

The Offer resource provides basic offer header attributes that can be imported, for example offer name and validity start and end date.

#### i Note

If an `OfferDateRuleType` entity is assigned, then no start or end date are allowed in the payload. Otherwise the execution will fail and an error message will be produced.

If you have enabled extension fields for the import service using the [Custom Fields](#) app, these extension attributes are also available to the offer resource to be imported.

HTTP Method	Operation	URI
GET	Single read an imported offer and its assigned objects.	<code>/Offers(&lt;key&gt;)</code> <code>/Offers(&lt;key&gt; / MarketingLocations</code> <code>/Offers(&lt;key&gt; / Products</code> <code>/Offers(&lt;key&gt; / ProductCategories</code> <code>/Offers(&lt;key&gt; / Rules</code>
PUT MERGE PATCH	Update an already imported offer, for example change the external status.	<code>/Offers(&lt;key&gt;)</code>
DELETE	Delete an imported offer	<code>/Offers(&lt;key&gt;)</code>

### Properties

- `OfferIdExt`: The unique offer ID provided by the external system that is used as part of the internal offer key in the import scenario.
- `OfferIdOrigin`: A unique identifier of the origin of the external offer. This origin ID is also used as part of the internal offer key in the import scenario. It should logically match the `SourceSystemId` and `SourceSystemType` of the `ImportHeader` entity type.
- `Name`: Name of the external offer (freetext).
- `Description`: Description of the external offer (freetext).
- `MarketingAreaDescription`: The description of the marketing area, which must be known to the system. Based on the description, the import system determines the ID of the marketing area. Note that either the marketing area description or the marketing area ID has to be provided.

- **MarketingAreaID:** The ID of the marketing area, which must be known to the system. Note that either the marketing area description or the marketing area ID has to be provided.
- **StartDate:** The validity start date of the offer (timestamp with timezone offset). Not allowed together with `OfferDateRuleType` entity. Either offer dates or validity rules must be filled.
- **EndDate:** The validity end date of the offer (timestamp with timezone offset). Not allowed together with `OfferDateRuleType` entity. Either offer dates or validity rules must be filled.
- **ExternalStatus:** Status of the offer that can be defined by the external system (optional). Admissible values can be retrieved by the `ExternalOfferStatus` entity.

If the `ExternalStatus` property is used, it is possible to control the internal lifecycle status of the offer under certain conditions. If the conditions are met, the internal offer status corresponds with the external status. In an integrated environment, it is then no longer necessary to release the imported offer manually in the SAP Marketing Cloud system. The release can instead be triggered by the importing system. The conditions under which the external status is mapped to the internal status are met if the offer is completely managed externally. This is the case if:

- `TargetGroupManagedExternally` was set to `true` during offer creation and
- `CouponManagedExternally` was either set to 'E' or left blank when the offer was imported and
- `ContentManagedExternally` was set to `true` during offer creation.

The following status values are available:

- 00 for *In Preparation*
- 01 for *Released*
- 02 for *Paused*

The following status transitions are possible:

- From 00 to 01.
- From 01 to 00 if the offer start date is in the future.
- From 01 to 02 if the offer start date is in the past.
- From 02 to 01.

It is possible to create or import an offer with the status *Released* directly without having to import it with the *In Preparation* status first and then update it to *Released*. However, this will only succeed if all prerequisites for releasing an offer are met (such as offer content must be available; if it is an offer with coupon, a released coupon must be assigned; and so on).

- **OfferContentManagedExternally:** Defines whether the offer content is managed by the consumer of this API (=true) or by the *Manage Offers* application in SAP Marketing Cloud (=false). If you want to import offer content, you must set the property to true. The value for this property cannot be changed during a subsequent update to the offer and retains its original value.
- **ExternalStatusDescription:** Status description of the offer that can be defined by the external system (optional). Admissible values can be retrieved by the `ExternalOfferStatus` entity.
- **TargetGroupManagedExternally:** Defines if the target group assignments to externally created offers is managed by the import service (= `true`) or by the Manage Offers app (= `false`). In case you want to import target group assignments with this OData Service, this property must be set to `true` for the respective offer. If you later update the offer, the value for this property cannot be changed and will keep its original value.
- **CouponManagedExternally:** Defines if the coupon assignment to externally created offers is managed by the import service (= `E`) or by the Manage Offers app (= blank (`I`)) or if there is no coupon assignment to the offer at all (= ' '). In the latter case, the offer will be created without the coupon feature. In case you want to import a coupon assignment with this OData Service, this property must be set to `E` for the respective offer. If you later update the offer, the value for this property cannot be changed and will keep its original value.



- `OfferIsFundedBySupplier`: A boolean flag that indicates if the offer is funded by a supplier or vendor of the offered products.
- `ProjectedGrossMarginInPercent`: The projected or calculated gross margin in percent, for example, for an offer.

## i Note

### Marketing Area

When assigning additional objects to an offer like target groups or coupons, the corresponding marketing area of the assigned objects need to match the marketing area of the offer. Keep in mind that also related marketing areas can taken into account when the enhancement option *Allowed Marketing Areas* has been implemented.

For more information, see [Allowed Marketing Areas](#).

## OData Resource: OfferContent

Represents an imported offer content.

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/OfferContents`

### Operations

The `OfferContent` resource provides offer content attributes that can be imported, such as the image URL and target link URL for offer content with the type *Image*. If you have enabled offer content extension fields for the import service using the *Custom Fields* app, these extension attributes are also available to the offer content resource to be imported.

HTTP Method	Operation	URI
GET	Single read a specific offer content or read all content defined for a given offer.	<code>/OfferContents(&lt;key&gt;)</code> <code>/Offers(&lt;key&gt;)/OfferContent</code>
PUT MERGE PATCH	Update an offer content entity that has already been imported, for example change the URL of an image.	<code>/OfferContents(&lt;key&gt;)</code>
DELETE	Delete an offer content instance.	<code>/OfferContents(&lt;key&gt;)</code>

### Properties

- `OfferIdExt`: The unique offer ID provided by the external system that is used as part of the internal offer key in the import scenario.
- `OfferIdOrigin`: A unique identifier of the origin of the external offer. This origin ID is also used as part of the internal offer key in the import scenario. It should logically match the `SourceSystemId` and `SourceSystemType` of the `ImportHeader` entity type.

- `MarketingOfferContent`: A consecutive number generated in the backend to identify a single offer content instance of an offer, such as `00001`.
- `LanguageISOCode`: The language ISO code defining the language of the offer content, such as `EN` or `DE`.
- `OfferContentType`: The identifier of the offer content type, such as `01`.
- `OfferContentTypeName`: The language-dependent name of the offer content type, such as `Image`.
- `CommunicationMedium`: The identifier of a communication medium to which the offer content is to apply, such as `EMAIL`.
- `CommunicationMediumName`: The language-dependent name of communication medium, such as `Email`.
- `OfferContentPosition`: The position of the offer content. This is an additional key field to define different offer contents of the same type for the same language and communication medium. In the case of offer content displayed on a webpage of an online shop, positions might be `TOP` or `BOTTOM`. The position can be freely defined by the consumer of the API.
- `OfferContentSourceURL`: The URL of an image.
- `OfferContentSourceURLDesc`: A description for the image URL.
- `OfferContentTargetURL`: The URL of a target link mostly used to define the target of a click action on the image.
- `OfferContentTargetURLDesc`: A description for the target link URL.

## OData Resource: OfferDateRulesType

Represents an imported validity rule.

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/OfferContents`

### Operations

The `OfferDateRulesType` resource provides validity rule attributes that can be imported, such as visibility, recurrence, start date, and end date. If you have enabled extension fields for the import service using the [Custom Fields](#) app, these extension attributes are also available to the `OfferDateRulesType` resource to be imported.

#### i Note

When importing Validity Rules, please familiarize yourself with the [Special behavior for importing and updating validity rules \[page 996\]](#) at the end of this page.

HTTP Method	Operations	URI
GET	Single read a specific offer validity rule or read all validity rules defined for a given offer.	<code>/OfferDateRules (&lt;key&gt;)</code> <code>/Offer (&lt;key&gt;) /Rules</code>

HTTP Method	Operations	URI
PUT MERGE PATCH	Update an offer validity rule that has already been imported. For example, change the start date or end date of the validity rule.	<code>/OfferDateRules (&lt;key&gt;)</code>
POST	Creates one or more additional offer validity rules entities for an offer that has already been imported.	<code>/OfferDateRules</code>
Delete	Delete an offer validity rule instance.	<code>/OfferDateRules (&lt;key&gt;)</code>

## Properties

- `OfferIdExt`: The unique offer ID provided by the external system that is used as part of the internal offer key import scenario.
- `OfferIdOrigin`: A unique identifier of the origin of the external offer. This origin ID is also used as part of the internal offer key in the import scenario. It should logically match the `SourceSystemId` and `SourceSystemType` of the `ImportHeader` entity type.
- `OfferDateRule`: A consecutive number generated in the backend to identify a single offer validity rule instance of an offer, such as `00001`.
- `OfferDateRuleType`: Use the offer validity rule type to define the rule visibility and validity in the specified time period, e.g. `01` to make the rule only visible for the contact in this period, `02` to make the rule visible for the contact and valid for business processes in this period, and `03` to make the rule only valid for business processes in this period.
- `OfferDateRuleStartDateTime`: The start date of the validity rule (timestamp with timezone offset).
- `OfferDateRuleEndDateTime`: The end date of the validity rule (timestamp with timezone offset).

## OData Resource: MarketingLocationAssignment

Represents a marketing location assignment to an offer. The resource only contains the key fields for the offer and the marketing location. To import and read marketing locations, please use the corresponding import service. For more information, see [Marketing Locations \[page 710\]](#).

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/MarketingLocationAssignments`

### Operations

The `MarketingLocationAssignment` resource provides the necessary attributes to add, remove, and read marketing location assignments from existing offers.

HTTP Method	Operation	URI
GET	Query of all marketing locations assigned to a specific offer.	<code>/Offers (&lt;key&gt;) / MarketingLocations / MarketingLocationAssignments (&lt;key&gt;)</code>
POST	Add new assignment of a marketing location to an existing offer. The location must not be obsolete.	<code>/MarketingLocationAssignments</code>
DELETE	Delete existing marketing location assignment from an existing offer.	<code>/ MarketingLocationAssignments (&lt;key&gt;)</code>

## Properties

- `OfferIdExt`: The unique offer ID provided by the external system that is used as part of the internal offer key in the import scenario.
- `OfferIdOrigin`: A unique identifier of the origin of the external offer. This origin ID is also used as part of the internal offer key in the import scenario. It should logically match the `SourceSystemId` and `SourceSystemType` of the `ImportHeader` entity type.
- `MarketingLocationId`: Unique marketing location ID provided by the external system. With the `MarketingLocationOriginId`, it is the external identifier of the master data object `MarketingLocation`. In the import scenario, it is part of the key used to assign marketing locations to an offer.
- `MarketingLocationOriginId`: A unique identifier of the origin of the external marketing location. In the import scenario, this origin ID is part of the key used to assign marketing locations to an offer.

## OData Resource: ProductAssignment

Represents the product assignment to an offer. The resource only contains the key fields for the offer and the product. To import and read products, please use the corresponding import service. For more information, see [Products \[page 582\]](#).

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/ProductAssignment`

## Operations

The `ProductAssignment` resource provides the necessary attributes to add, remove, and read products from existing offers.

HTTP Method	Operation	URI
GET	Query of all products assigned to a specific offer.	<code>/Offers (&lt;key&gt;) /Products /ProductAssignments (&lt;key&gt;)</code>
POST	Add new assignment of a product to an existing offer.	<code>/ProductAssignments</code>
DELETE	Delete existing product assignment from an existing offer.	<code>/ProductAssignments (&lt;key&gt;)</code>

## Properties

- `offerIdExt`: The unique offer ID provided by the external system that is used as part of the internal offer key the import scenario.
- `offerIdOrigin`: A unique identifier of the origin of the external offer. This origin ID that is also used as part of the internal offer key in the import scenario. It should logically match the `SourceSystemId` and `SourceSystemType` of the `ImportHeader` entity type.
- `productId`: Unique product ID provided by the external system. With the `ProductOriginId`, it is the external identifier of the master data object Product. In the import scenario it is part of the key used to assign a product to an offer.
- `ProductOriginId`: A unique identifier of the origin of the external product. In the import scenario, this origin ID is part of the key used to assign a product to an offer.

## OData Resource: ProductCategoryAssignment

Represents the product category assignment to an offer. The resource only contains the key fields for the offer and the product category. To import and read product categories, please use the corresponding import service. For more information, see [Product Hierarchies and Categories \[page 604\]](#).

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/ProductCategoryAssignments`

## Operations

The `ProductCategoryAssignment` resource provides the necessary attributes to add, remove, and read product categories from existing offers.

HTTP Method	Operation	URI
GET	Query to get all product categories assigned to a specific offer.	<code>/Offers (&lt;key&gt;) /ProductCategories /ProductCategoryAssignments (&lt;key&gt;)</code>

HTTP Method	Operation	URI
POST	Add new assignment of a product category to an existing offer.	/ProductCategoryAssignments
DELETE	Delete existing product category assignment from an existing offer.	/ProductCategoryAssignments (<key>)

## Properties

- **OfferIdExt:** The unique offer ID provided by the external system that is used as part of the internal offer key in the import scenario.
- **OfferIdOrigin:** A unique identifier of the origin of the external offer. This origin ID that is also used as part of the internal offer key in the import scenario. It should logically match the `SourceSystemId` and `SourceSystemType` of the `ImportHeader` entity type.
- **CategoryId:** The unique category ID provided by the external system. Together with the `HierarchyId`, it is the external identifier of the master data object Product Category. In the import scenario, it is part of the key used to assign a product category to an offer.
- **HierarchyId:** A unique identifier external identifier of the product category Hierarchy. In the import scenario, this `HierarchyID` is part of the key used to assign a product category to an offer.

## OData Resource: TargetGroupAssignment

Represents a target group assignment to an offer. The resource only contains the key fields for the offer and the target group. To import and read target groups, please use the corresponding import service. For more information, see [Target Groups \[page 755\]](#).

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/TargetGroupAssignments`

## Operations

The `TargetGroupAssignment` resource provides the necessary attributes to add, remove, and read target group assignments from existing offers.

HTTP Method	Operation	URI
GET	Query of all target groups assigned to a specific offer.	/Offers (<key>) /TargetGroups /TargetGroupAssignments (<key>)
POST	Add new assignment of a target group to an existing offer. The target group must be in status <i>Released</i> and the category must not be <i>Live</i> . Only static or dynamic target groups are allowed.	/TargetGroupAssignments

HTTP Method	Operation	URI
DELETE	Delete existing target group assignment from an existing offer.	/TargetGroupAssignments (<key>)

## Properties

- **OfferIdExt:** The unique offer ID provided by the external system that is used as part of the internal offer key in the import scenario.
- **OfferIdOrigin:** A unique identifier of the origin of the external offer. This origin ID is also used as part of the internal offer key in the import scenario. It should logically match the `SourceSystemId` and `SourceSystemType` of the `ImportHeader` entity type.
- **TargetGroupKey:** Unique target group key provided by the external system. Note that this key is a GUID which must be known to the caller.

## odata Resource: CouponAssignment

Represents a coupon assignment to an offer. The resource only contains the key fields for the offer and the coupon. Note that in difference to all other assignments, the coupon represents a 1:1 relationship to the offer, for example, only a single coupon can be assigned to an offer. To import and read coupons, please use the corresponding import service. For more information, see [Coupons \[page 1026\]](#).

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/CouponAssignments`

## Operations

The `CouponAssignment` resource provides the necessary attributes to add, remove, and read coupon assignments from existing offers.

HTTP Method	Operation	URI
GET	Query of the coupon assigned to a specific offer.	/Offers (<key>) /Coupons /CouponAssignments (<key>)
POST	Add new assignment of a coupon to an existing offer and implicitly delete the old one. The coupon must not be assigned to another offer already.	/CouponAssignments
DELETE	Delete existing coupon assignment from an existing offer.	/CouponAssignments (<key>)

## Properties

- **OfferIdExt:** The unique offer ID provided by the external system that is used as part of the internal offer key in the import scenario.

- `offerIdOrigin`: A unique identifier of the origin of the external offer. This origin ID is also used as part of the internal offer key in the import scenario. It should logically match the `sourceSystemId` and `sourceSystemType` of the `ImportHeader` entity type.
- `CouponUUID`: Unique coupon identifier provided by the external system. Note that this key is a GUID which must be known to the caller.
- `Coupon`: Unique user-assigned coupon identifier. Can be used to identify which coupon should be assigned to the offer. If both `CouponUUID` and `Coupon` are provided, the system will use `CouponUUID` and ignore `Coupon`.

## Common HTTP Headers

Common request and response headers used by the service operations.

### Common Request Headers

Header	Required	Description
Content-Type	No	Describes the format of the request body, for example, <code>application/json</code> . All examples in this document use JSON format for the payloads.
X-CSRF-Token	Yes	A security token that must be passed with every request.

### Common Response Headers

Header	Required	Description
SAP-Messages	No	If a request is successful, messages can be returned to the service consumer in this HTTP header. For example, in the case of deep-create offers with the property <code>ProcessAllOrNothing</code> set to false, this header contains potential errors occurred during the creation. However, the response shows the actual data created in the system.

### Common Status and Error Codes

Code	Reason
400	Bad request, for example, an offer with the same key already exists.
404	Not found, for example, an offer with the given key cannot be found in the system.



Code	Reason
201	Offer successfully imported.

## OData Operation: Bulk Import Offers

Import offers using the resource `ImportHeader` to create multiple offers with their assignments collectively (OData deep-create).

### Request

**URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/ImportHeaders`

**HTTP Method:** POST

### Request Example:

[POST] `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/ImportHeaders`

### Request Payload Example

Import exactly one offer using the `ImportHeader` resource.

#### Sample Code

```
{
  "UserName" : "IMPORT_USER",
  "SourceSystemType" : "SGC",
  "SourceSystemId" : "Gateway Client",
  "ProcessAllOrNothing" : true,
  "Offers" : [{
    "OfferIdExt" : "0000000001",
    "OfferIdOrigin" : "SAP_PMR",
    "Name" : "PMR Offer Name",
    "Description" : "Offer Description ",
    "MarketingAreaDescription" : "Global",
    "StartDate" : "\/Date(1432634400000)\/",
    "EndDate" : "\/Date(1441872000000)\/"
  }]
}
```

### Request Example:

[POST] `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/ImportHeaders`

### Request Payload Example

Complex payload example of creating two offers, each offer including:

- Two Marketing Locations
- Two Validity Rules
- Two Products
- Two Product Categories
- Two Target Groups
- One Offer

## Sample Code

```
{
  "UserName" : "IMPORT_USER",
  "SourceSystemType" : "SGC",
  "SourceSystemId" : "Gateway Client",
  "ProcessAllOrNothing" : true,
  "CouponManagedExternally" : "E",
  "Offers" : [{
    "OfferIdExt" : "OFFER_0057",
    "OfferIdOrigin" : "SAP_PMR",
    "Name" : "PMR Offer 0057",
    "Description" : "Offer 0057 from Gateway Test Client",
    "MarketingAreaDescription" : "Global",
    "MarketingLocations" : [{
      "MarketingLocationOriginId" : "SAP_HYBRIS_COMMERCE_POS",
      "MarketingLocationId" : "99998"
    },
    {
      "MarketingLocationOriginId" : "SAP_HYBRIS_COMMERCE_POS",
      "MarketingLocationId" : "99985"
    }
  ]
},
  "Rules" : [
    {
      "OfferDateRule": "0001",
      "OfferDateRuleType": "01",
      "OfferDateRuleStartDateTime": "\/Date(1530900780000)\/",
      "OfferDateRuleEndDateTime": "\/Date(1531772000000)\/",
    },
    {
      "OfferDateRule": "0002",
      "OfferDateRuleType": "01",
      "OfferDateRuleStartDateTime": "\/Date(1531000780000)\/",
      "OfferDateRuleEndDateTime": "\/Date(1531572000000)\/",
    }
  ],
  "Products" : [{
    "ProductOriginId" : "SAP_ERP_MATNR",
    "ProductId" : "887749052850"
  },
  {
    "ProductOriginId" : "SAP_ERP_MATNR",
    "ProductId" : "887749052848"
  }
],
  "ProductCategories" : [{
    "HierarchyId" : "GENERATED_HIERARCHY_ID",
    "CategoryId" : "Fleece2"
  },
  {
    "HierarchyId" : "GENERATED_HIERARCHY_ID",
    "CategoryId" : "Fleece3"
  }
],
  "TargetGroups" : [{
    "TargetGroupKey": "6C0B84B7-5523-1ED7-8BFB-CFE77A316EC7"
  },
  {
    "TargetGroupKey": "6C0B84B7-5523-1ED7-8AB7-D828EE609B8D"
  }
],
  "Coupons":
  {
    "OfferIdExt": "OFFER_0057",
    "OfferIdOrigin": "SAP_PMR",
    "CouponUUID": "6C0B84B7-5523-1ED7-8BFB-CFE77A316EC7",
  }
},
```

```

{
  "OfferIdExt" : "OFFER_0058",
  "OfferIdOrigin" : "SAP_PMR",
  "Name" : "PMR Offer 0058",
  "Description" : "Offer 0058 from Gateway Test Client",
  "MarketingAreaDescription" : "Global",
  "MarketingLocations" : [{
    "MarketingLocationOriginId" : "SAP_HYBRIS_COMMERCE_POS",
    "MarketingLocationId" : "99998"
  },
  {
    "MarketingLocationOriginId" : "SAP_HYBRIS_COMMERCE_POS",
    "MarketingLocationId" : "99985"
  }
],
  "Rules": [
    {
      "OfferDateRule": "0001",
      "OfferDateRuleType": "01",
      "OfferDateRuleStartDateTime": "\\Date(1530900780000)\\",
      "OfferDateRuleEndDateTime": "\\Date(1531772000000)\\",
    },
    {
      "OfferDateRule": "0002",
      "OfferDateRuleType": "03",
      "OfferDateRuleStartDateTime": "\\Date(1531000780000)\\",
      "OfferDateRuleEndDateTime": "\\Date(1531572000000)\\",
    }
  ],
  "Products" : [{
    "ProductOriginId" : "SAP_ERP_MATNR",
    "ProductId" : "887749052850"
  },
  {
    "ProductOriginId" : "SAP_ERP_MATNR",
    "ProductId" : "887749052848"
  }
],
  "ProductCategories" : [{
    "HierarchyId" : "GENERATED_HIERARCHY_ID",
    "CategoryId" : "Fleece2"
  },
  {
    "HierarchyId" : "GENERATED_HIERARCHY_ID",
    "CategoryId" : "Fleece3"
  }
],
  "TargetGroups": [{
    "TargetGroupKey": "6C0B84B7-5523-1ED7-8BFB-CFE77A316EC7"
  },
  {
    "TargetGroupKey": "6C0B84B7-5523-1ED7-8AB7-D828EE609B8D"
  }
],
  "Coupons":
  {
    "CouponKey": "6C0B84B7-5523-1ED7-8BFB-CFE77A316EC7",
  }
}
]
}

```

## OData Operation: Update an Offer That Has Already Been Imported

Update field values in an existing offer. The values of the following properties can be changed:

- Name
- Description

- StartDate
- EndDate
- ExternalStatus
- ExternalStatusDescription

System response is defined by the HTTP method used:

- **MERGE or PATCH:** It is possible to update a single value, for example the value for the description. All other fields won't be changed. No mandatory fields.
- **POST:** It is possible to add an additional value, for example TargetGroupAssignments. All other fields won't be changed. No mandatory fields.
- **PUT:** All fields will be updated, fields not mentioned in the request payload will be initialized. Update won't be performed if not all mandatory fields are included in the request payload.

### Request

**URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/Offers`

**HTTP Method:** POST, MERGE, PATCH, or PUT.

### Request Example:

Example of changing the external status of an already imported offer to "released".

[PATCH] `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/Offers(<key>`

### Request Payload Example

#### Sample Code

```
{
  "ExternalStatus": "01"
  "ExternalStatusDescription": ""
}
```

### Request Example:

Example of changing an offer using HTTP PUT. Properties not included in the payload will be set to their initial value.

[PUT] `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/Offers(<key>`

### Request Payload Example

#### Sample Code

```
{
  "Name" : "Offer_0120_new",
  "Description" : "brand new Text for Offer 120",
  "StartDate" : "\/Date(1432634400000)\/",
  "EndDate" : "\/Date(1432734400000)\/",
  "ExternalStatus" : "02"
}
```

## OData Operation: Delete Existing Offer

An existing offer will be deleted including all assigned marketing location, products, and product categories.

## Request

**URI:** https://<Server>:<Port>/sap/opu/odata/sap/CUAN\_OFFER\_IMPORT\_SRV/Offers

**HTTP Method:** DELETE

### Request Example:

```
[DELETE] https://<Server>:<Port>/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/Offers(<key>)
```

## OData Operation: Read Offer Details

Read offer details of an existing offer.

## Request

**URI:** https://<Server>:<Port>/sap/opu/odata/sap/CUAN\_OFFER\_IMPORT\_SRV/Offers(<offer key>)

**HTTP Method:** GET

### Request Example:

Read the offer header data only.

```
[GET] https://<Server>:<Port>/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/Offers(OfferIdExt='EXT_ID_0001', OfferIdOrigin='SAP_PMR')
```

### Response Payload Example

#### Sample Code

```
{
  "OfferIdExt" : "EXT_ID_0001",
  "OfferIdOrigin" : "SAP_PMR",
  "Name" : "PMR Offer Name",
  "Description" : "PMR Offer Description ",
  "MarketingAreaId" : "Global",
  "MarketingAreaDescription" : "Global",
  "StartDate" : "\/Date(1432634400000)\/",
  "EndDate" : "\/Date(1441872000000)\/",
  "ExternalStatus" : "01",
  "ExternalStatusDescription" : "Externally Released"
}
```

### Request Example:

Read the offer header data and all of the assignment information in one request.

```
[GET] https://<Server>:<Port>/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/Offers(OfferIdExt='EXT_ID_0001', OfferIdOrigin='SAP_PMR')?
$expand=Prodcuts,ProductCategories,MarketingLocations
```

### Response Payload Example

#### Sample Code

```
{
  "OfferIdExt" : "EXT_ID_0001",
  "OfferIdOrigin" : "SAP_PMR",
  "Name" : "PMR Offer Name",
```

```

"Description" : "PMR Offer Description ",
"MarketingAreaId" : "Global",
"MarketingAreaDescription" : "Global",
"StartDate" : "\/Date(1432634400000)\/",
"EndDate" : "\/Date(1441872000000)\/",
"ExternalStatus" : "01",
"ExternalStatusDescription" : "Externally Released",
"Products" : [{
  "ProductOriginId" : "SAP_ERP_MATNR",
  "ProductId" : "887749052850"
}],
"ProductCategories" : [{
  "HierarchyId" : "GENERATED_HIERARCHY_ID",
  "CategoryId" : "Fleece2"
}],
"MarketingLocations" : [{
  "MarketingLocationOriginId" : "SAP_HYBRIS_COMMERCE_POS",
  "MarketingLocationId" : "99998"
}]
}

```

## OData Operation: Assign Marketing Location for an Existing Offer

Create single marketing location assignment to an existing offer.

### Request

**URI:** `https://<Server>:<Port>/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/MarketingLocationAssignments`

**HTTP Method:** POST

### Request Example:

[POST] `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_IMPORT_SRV/MarketingLocationAssignments`

### Response Payload Example

Add exactly one marketing location to an existing offer.

#### Sample Code

```

{
  "OfferIdExt" : "OFFER_0020",
  "OfferIdOrigin" : "SAP_PMR",
  "MarketingLocationOriginId" : "SAP_HYBRIS_COMMERCE_POS",
  "MarketingLocationId" : "99998"
}

```

## OData Operation: Remove Marketing Location Assignment from an Existing Offer

Remove assignment of a single marketing location assignment from an existing offer.

### Request

**URI:** `https://<Server>:<Port>/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/MarketingLocationAssignments(<offer_key><mkt_location_key>`

**HTTP Method:** DELETE

### Request Example:

```
[DELETE] https://<Server>:<Port>/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/MarketingLocationAssignments(OfferIdExt='OFFER_0020',OfferIdOrigin='SAP_PMR',MarketingLocationOriginId='SAP_HYBRIS_COMMERCE_POS',MarketingLocationId='99998')
```

### OData Operation: Create, Remove Assignment, and Get Operation for Products, Product Categories, Target Groups, Coupons, and Validity Rules.

Create, remove assignment, and get operations for products, product categories, target groups, coupons, and validity rules (only create and get, remove is different) are similar to the operations for marketing locations described above.

#### i Note

When removing validity rules, the offer must contain at least one remaining validity rule.

For create and remove assignment operations replace `MarketingLocationAssignments` in the URI with `ProductAssignments`, `ProductCategoryAssignments`, `TargetGroupAssignments`, `CouponAssignments`, or `OfferDateRules`.

Within the create request payload example replace the marketing location specific properties with product-, product-category-, target-group-, coupon-, or validity-rules-specific properties.

To perform the GET operation, replace `MarketingLocations` with `Products`, `ProductCategories`, `TargetGroups`, `Coupons`, or `OfferDateRules`.

### OData Operation: Post New Offer with 2 Validity Rules

Post a single offer with two validity rules.

#### Request

**URI:** `https://<Server>:<Port>/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/ImportHeaders`

**HTTP Method:** POST

#### Request Example:

```
[POST] https://<Server>:<Port>/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/ImportHeaders
```

#### Request Payload Example

##### Sample Code

```
{
  "Id": "",
  "Timestamp": null
  "UserName": "",
  "SourceSystemType": "SGC",
  "SourceSystemId": "Gateway Client",
  "ProcessAllOrNothing": true,
  "ImportMode": "U",
  "Offers": [
    { "OfferIdExt": "EXT_OFFER_W_RULES",
      "OfferIdOrigin": "GENERIC",
      "Name": " EXT_OFFER_W_RULES ",
      "Description": "Offer From Gateway Test Client",
      "MarketingAreaDescription": "Global",
```

```

"StartDate": "\/Date(1530800780000)\/",
"EndDate": "\/Date(1531872000000)\/",
"TargetGroupManagedExternally": true,
"OfferContentManagedExternally": true,
"OfferContent": [
  {
    "MarketingOfferContent": "00001",
    "LanguageISOCode": "EN",
    "OfferContentType": "01",
    "CommunicationMedium": "EMAIL",
    "OfferContentPosition": "HOME",
    "OfferContentSourceURL": "https://bild1.jpg",
    "OfferContentSourceURLDesc": "SourceBild1",
    "OfferContentTargetURL": "https://bild1.jpg",
    "OfferContentTargetURLDesc": "TargetBild1",
    "__metadata": {"type": "CUAN_OFFER_IMPORT_SRV.OfferContent"}
  }
],
"Rules": [
  {
    "OfferDateRule": "0001",
    "OfferDateRuleType": "01",
    "OfferDateRuleStartDateTime": "\/Date(1530900780000)\/",
    "OfferDateRuleEndDateTime": "\/Date(1531772000000)\/",
    "__metadata": {"type": "CUAN_OFFER_IMPORT_SRV.OfferDateRulesType"}
  },
  {
    "OfferDateRule": "0002",
    "OfferDateRuleType": "03",
    "OfferDateRuleStartDateTime": "\/Date(1531000780000)\/",
    "OfferDateRuleEndDateTime": "\/Date(1531572000000)\/",
    "__metadata": {"type": "CUAN_OFFER_IMPORT_SRV.OfferDateRulesType"}
  }
],
"__metadata": {"type": "CUAN_OFFER_IMPORT_SRV.Offer"}
}],
"__metadata": {"type": "CUAN_OFFER_IMPORT_SRV.ImportHeader"}
}

```

## Special behavior for importing and updating Validity Rules

### Import Validity Rules

	Start Date Included	End Date Included	Number of Validity Rules in Request	Result
ImportHeaders Entityset	No	No	0	"Either offer dates or validity rules must be filled."
	No	No	>0	Success
	Yes	No	0	"End date must be filled."
	Yes	No	>0	"Not allowed to import offer dates and rules."



## Import Validity Rules

	No	Yes	0	"Start date must be filled."
	No	Yes	>0	"Not allowed to import offer dates and rules."
	Yes	Yes	0	Success
	Yes	Yes	>0	"Not allowed to import offer dates and rules."
<b>Offers Entityset - Patch</b>	No	No	-	Success
	Yes	No	-	If offer has only 1 rule assigned, change the start date of the rule and the offer. If >1 rule is assigned: "Start date of offers with more than one rule assigned cannot be changed."
	No	Yes	-	If offer has only 1 rule assigned, change the end date of the rule and the offer. If >1 rule is assigned: "End date of offers with more than one rule assigned cannot be changed."
	Yes	Yes	-	If offer has only 1 rule assigned, change the start and end dates of the rule and the offer. If >1 rule is assigned: "Start and end dates of offers with more than one rule assigned cannot be changed."
<b>Offers Entityset - Put</b>	No	No	-	"Offer start and end dates are not filled."
	Yes	No	-	"Offer end date is not filled."
	No	Yes	-	"Offer start date is not filled."

### Import Validity Rules

Yes	Yes	-	If offer has only 1 rule assigned, change the start and end dates of the rule and the offer. If >1 rule is assigned: "Start and end dates of offers with more than one rule assigned cannot be changed."
-----	-----	---	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Update Offer Validity

	Start Date	End Date	Number of Validity Rules in the Request	Offer Status	Result
<b>ImportHeaders Entityset</b>	No	No	0	In Preparation	"Either offer dates or validity rules must be filled."
	No	No	>0	In Preparation	Replace the existing rules with new ones.
	Yes	No	0	In Preparation	If only one rule is assigned, update start date, keep the end date. If >1 rule is assigned: "Start and end dates of offers with more than one rule assigned cannot be changed."
	Yes	No	>0	In Preparation	"Not allowed to import offer dates and rules."

## Update Offer Validity

	No	Yes	0	In Preparation	If only one rule is assigned, update end date, keep the start date. If >1 rule is assigned: "Start and end dates of offers with more than one rule assigned cannot be changed."
	No	Yes	>0	In Preparation	"Not allowed to import offer dates and rules."
	Yes	Yes	0	In Preparation	Delete all assigned rules and create a new one with the given start and end dates.
	Yes	Yes	>0	In Preparation	"Not allowed to import offer dates and rules."
<b>ImportHeaders Entityset</b>	No	No	0	Paused	"Either offer dates or validity rules must be filled."
	No	No	>0	Paused	<p>If any stored or imported rules have any dates in the past, then an error is raised.</p> <p>If all stored and imported rules have all dates in the future, then replace the existing rules with new ones.</p>

## Update Offer Validity

Yes	No	0	Paused	<p>If only one rule is assigned and start date is in the future, then update the start date and keep the end date. Otherwise, an error message is produced: "Start date is in the past and cannot be changed."</p> <p>If &gt;1 Rule is stored, then an error message is produced: "Start and end dates of offers with more than one rule assigned cannot be changed."</p>
Yes	No	>0	Paused	"Not allowed to import offer dates and rules."
No	Yes	0	Paused	<p>If only one rule is stored and end date is in the future, then update the end date and keep the start date (also if it is in the past). Otherwise, an error message is produced: "End date is in the past and cannot be changed."</p> <p>If &gt;1 rule is stored, then an error message is produced: "Start and end dates of offers with more than one rule assigned cannot be changed."</p>

## Update Offer Validity

No	Yes	>0	Paused	"Not allowed to import offer dates and rules."
Yes	Yes	0	Paused	<p>If &gt;1 rule is stored and any date is in the past, then an error message is produced: "Start and end dates of offers with more than one rule assigned cannot be changed."</p> <p>If all stored rules start in the future, then delete all assigned rules and create a new one with the given start and end dates.</p> <p>If only one rule is assigned, then send dates to BOPF (tests), change existing rule and same behavior &lt; 1902.</p>
Yes	Yes	>0	Paused	"Not allowed to import offer dates and rules."

## Related Information

[Custom Fields](#)

## 5.6.6 Read Offers

Public OData API (API\_MKT\_OFFER\_SRV) for Offers

### Technical Data

The public API for Offers supports operations on the `Offers` Business Object.



<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_OFFER_SRV</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_OFFER_SRV/\$metadata</code>
<b>Authorizations</b>	The following business catalog is required: SAP_CEC_BC_MKT_API_OFM_PC
Communication Scenario ID	SAP_COM_0306
Component for Incidents	CEC-MKT-OFM (Offers)
	<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"><p><b>i Note</b></p><p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p></div>
Field Extensibility Supported	Yes

You can view sample payloads and test the API at <https://api.sap.com>.

### Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_OFFER_SRV/\$metadata?sap-documentation=all</code>	Only for internal access. You need to provide the server and port names.

Access Link	Remarks
<a href="#">Marketing - Offers Details Page</a> 	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Read Offers API</a> 	General access link takes you directly to the <i>Offers</i> metadata file. One-time registration or logon is required.

### Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Entity Sets

The Offers OData API provides the following entities:

Entity Set	Description	Path
Offers	This entity contains a list of offers.	<code>/Offers</code>
OfferContents	This entity contains the contents of an offer.	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferContents</code>
OfferCoupons	This entity contains offer coupons.	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferCoupons</code>
OfferMarketingLocations	This entity contains the marketing locations of offers.	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferMarketingLocations</code>
OfferProducts	This entity contains the products that are on offer.	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferProducts</code>
OfferProductCategories	This entity contains the categories of products that are on offer.	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferProductCategories</code>
OfferTargetGroups	This entity contains the target groups to which you want to send the offers.	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferTargetGroups</code>

Entity Set	Description	Path
OfferFeatures	This entity contains the features of offers.	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferFeatures</code>
OfferFacets	This entity contains external offers.	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferFacets</code>

## Offers

**Resource Path:** `/Offers`

You can perform the following operations on the `Offers` entity set:

Operations on `Offers` entity set

HTTP Method	Description	Path
<a href="#">GET</a>	Get a list of offers	<code>/Offer?\$top</code>
	Get the details of a specific offer	<code>/Offers(guid'&lt;Offer UUID&gt;')</code>

## OfferContents

**Resource Path:** `/Offers(guid'<Offer UUID>')/OfferContents`

You can perform the following operations on the `OfferContents` entity set:

Operations on `OfferContents` entity set

HTTP Method	Description	Path
<a href="#">GET</a>	Get all contents of an offer	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferContents</code>
	Get a specific offer content	<code>/OfferContents(guid'&lt;Offer Contents UUID&gt;')</code>
	Get the offer or offers in which a specific offer content is used	<code>/OfferCoupons(CouponUUID=guid'&lt;Coupon UUID&gt;',MarketingOfferUUID=guid'&lt;Marketing Offer UUID&gt;')/Offer</code>

## OfferCoupons

**Resource Path:** `/Offers(guid'<Offer UUID>')/OfferCoupons`

You can perform the following operations on the `OfferCoupons` entity set:

Operations on `OfferCoupons` entity set

HTTP Method	Description	Path
<a href="#">GET</a>	Get all coupons of an offer	<code>/Offers(guid'&lt;Offer UUID&gt;')/OfferCoupons</code>



HTTP Method	Description	Path
	Get a specific coupon of an offer	/ OfferCoupons (MarketingOfferUUID=guid '<Marketing Offer UUID>', CouponUUID=guid '<Coupon UUID>')
	Get the offer or offers in which a specific coupon is used	/ OfferCoupons (CouponUUID=guid '<Coupon UUID>', MarketingOfferUUID=guid '<Marketing Offer UUID>')/Offer

## OfferMarketingLocations

**Resource Path:** /Offers (guid '<Offer UUID>')/OfferMarketingLocations

You can perform the following operations on the OfferMarketingLocations entity set:

Operations on OfferMarketingLocations entity set

HTTP Method	Description	Path
<i>GET</i>	Get all marketing locations of an offer	/Offers (guid '<Offer UUID>') / OfferMarketingLocations
	Get a specific marketing location of an offer	/ OfferMarketingLocations (MarketingOfferUUID=guid '<Marketing Offer UUID>', MarketingLocationUUID=guid '<Marketing Location UUID>')
	Get the offer or offers in which a specific marketing location is used	/ OfferMarketingLocations (MarketingLocationUUID=guid '<Marketing Location UUID>', MarketingOfferUUID=guid '<Marketing Offer UUID>')/Offer

## OfferProducts

**Resource Path:** /Offers (guid '<Offer UUID>')/OfferProducts

You can perform the following operations on the OfferProducts entity set:

Operations on OfferProducts entity set

HTTP Method	Description	Path
<i>GET</i>	Get all products of an offer	/Offers (guid '<Offer UUID>') / OfferProducts

HTTP Method	Description	Path
	Get a specific product of an offer	/ OfferProducts (MarketingOfferUUID=guid'<Marketing Offer UUID>',ProductUUID=guid'<Product UUID>')
	Get the offer or offers in which a specific product is used	/ OfferProducts (ProductUUID=guid'<Product UUID>',MarketingOfferUUID=guid'<Marketing Offer UUID>')/Offer

## OfferProductCategories

**Resource Path:** /Offers (guid'<Offer UUID>')/OfferProductCategories

You can perform the following operations on the OfferProductCategories entity set:

Operations on OfferProductCategories entity set

HTTP Method	Description	Path
<i>GET</i>	Get all product categories of an offer	/Offers (guid'<Offer UUID>') / OfferProductCategories
	Get a specific product category of an offer	/ OfferProductCategories (MarketingOfferUUID=guid'<Marketing Offer UUID>',ProductCategoryUUID=guid'<Product Category UUID>')
	Get the offer or offers in which a specific product category is used	/OfferProductCategories (Product Category UUID=guid'<ProductCategoryUUID>',MarketingOfferUUID=guid'<Marketing Offer UUID>')/Offer

## OfferTargetGroups

**Resource Path:** /Offers (guid'<Offer UUID>')/OfferTargetGroups

You can perform the following operations on the OfferTargetGroups entity set:

Operations on OfferTargetGroups entity set

HTTP Method	Description	Path
<i>GET</i>	Get all target groups of an offer	/Offers (guid'<Offer UUID>') / OfferTargetGroups

HTTP Method	Description	Path
	Get a specific target group of an offer	<code>/OfferTargetGroups (MarketingOfferUUID=guid'&lt;Marketing Offer UUID&gt;',TargetGroupUUID=guid'&lt;Target Group UUID&gt;')</code>
	Get the offer or offers in which a specific target group is used	<code>/OfferTargetGroups (TargetGroupUUID=guid'&lt;Target Group UUID&gt;',MarketingOfferUUID=guid'&lt;Marketing Offer UUID&gt;')/Offer</code>

## OfferFeatures

**Resource Path:** `/Offers (guid'<Offer UUID>')/OfferFeatures`

You can perform the following operations on the OfferFeatures entity set:

Operations on OfferFeatures entity set

HTTP Method	Description	Path
<i>GET</i>	Get all features of an offer	<code>/Offers (guid'&lt;Offer UUID&gt;')/OfferFeatures</code>
	Get a specific feature of an offer	<code>/OfferFeatures (MarketingOfferUUID=guid'&lt;Marketing Offer UUID&gt;',OfferFeature='&lt;Offer Feature&gt;')</code>
	Get the offer or offers in which a specific feature is used	<code>/OfferFeatures (MarketingOfferUUID=guid'&lt;Marketing Offer UUID&gt;',OfferFeature=guid'&lt;Offer Feature&gt;')/Offer</code>

## OfferFacets

**Resource Path:** `/Offers (guid'<Offer UUID>')/OfferFacets`

You can perform the following operations on the OfferFacets entity set:

Operations on OfferFacets entity set

HTTP Method	Description	Path
<i>GET</i>	Get all external offers	<code>/Offers (guid'&lt;Offer UUID&gt;')/OfferFacets</code>

HTTP Method	Description	Path
	Get a specific external offer	/ OfferFacets (ExternalOffer=guid'<External Offer>', ExternalOfferOrigin=guid'<External Offer Origin>')
	Get the offer or offers for a given offer facet	/ OfferFacets (ExternalOffer=guid'<External Offer>', ExternalOfferOrigin=guid'<External Offer Origin>')/Offer

## 5.6.7 Discover Offers

Use the API OData service `CUAN_OFFER_DISCOVERY_SRV` for SAP Marketing Cloud Offers to find suitable offers for a consumer.

### Overview

The public OData service `CUAN_OFFER_DISCOVERY_SRV` can be used to retrieve suitable offer content to consumers for a specific context. For example in a Web shop, such as SAP Commerce, the service can be used to determine offer content, such as a banner, to be displayed on a Web page in the shop. To find the most relevant offer content, a number of context parameters can be passed to the service. Possible contexts for a Web shop scenario include the following:

- Current user logged on to the Web shop to show personalized offers
- Browser language to determine offer content (such as images) in the correct language

When requesting the OData with an Offer Recommendation Scenario ID, the rule based Offer Recommendation Intelligence will include recommended offers. Without the Scenario ID a simple solution will recommend offers based on solely on the eligibility.

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/ CUAN_OFFER_DISCOVERY_SRV</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/ CUAN_OFFER_DISCOVERY_SRV/\$metadata</code>
<b>Authorizations</b>	The following business catalog role is required: <code>SAP_COM_CSR_0021</code>

**i Note**

Not to be used for HTTP errors. For more information, see [HTTP Response Status Codes \[page 408\]](#).

You can view sample payloads and test the API at <https://api.sap.com>.

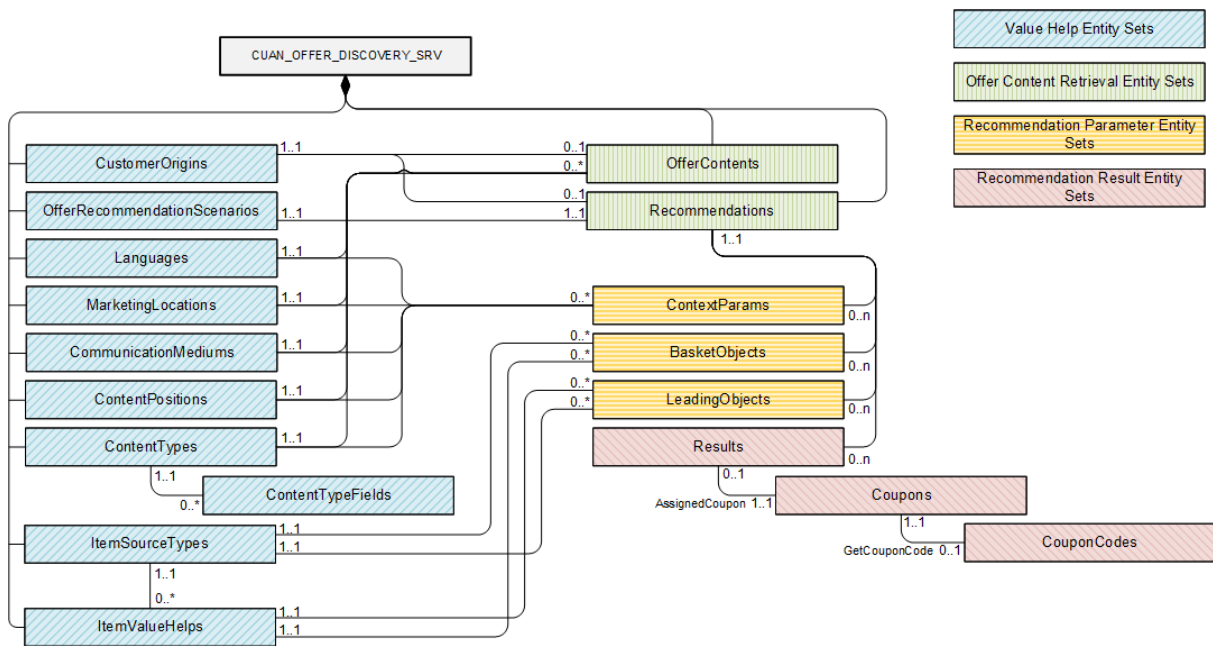
**Support of OData Features**

See the following chapters for implementation details and search behavior of the OData services.

Feature	Support
Query options for value help entities	<p>The current implementation of the value help entities supports the following query options, which can be passed as query or path parameters:</p> <ul style="list-style-type: none"> <li>• \$top and \$skip</li> <li>• \$select</li> <li>• \$orderby</li> <li>• \$count and \$inlinecount</li> </ul> <p>Exception: the entity set <code>ItemSourceTypes</code> supports only \$orderby, the entity set <code>ItemValueHelps</code> does not support \$inlinecount.</p>
http GET on OfferContents entity set	<p>This API can be used to retrieve Offer Content objects from the system using the input parameters provided. This API supports the retrieval of an offer without Offer Recommendation Intelligence (ORI) and with ORI with a subset of the available ORI functionality.</p>
http POST with deep-create on Recommendations entity set	<p>This API is the preferred method to retrieve Offer Content Recommendations using Offer Recommendation Intelligence (ORI). It supports enhanced capabilities for use with the ORI.</p>
http GET on function GetCouponCode	<p>This API endpoint can be used to retrieve coupon codes.</p>

**Entity Data Model**

The following graphic shows the available entity sets and their relationships to each other.



**Service Metadata URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/`  
`CUAN_OFFER_DISCOVERY_SRV/$metadata`

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<a href="https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV;v=0003/\$metadata?sap-documentation=all">https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV;v=0003/\$metadata?sap-documentation=all</a>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing Offer Discovery Service Details Page</a>	General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users. <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Discover Offers API</a>	General access link takes you directly to the <i>Offer Discovery Service</i> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Resources

The service consists of the following resource:

- Value help entities to provide values for certain code and identifiers used in other entities. These entities are read-only and support HTTP GET operation on the corresponding entity set to read the values defined in the system.
- Navigational entities, that cannot be called directly, but are used in combination with another entity. The API hub documentation shows these entities only as collections in the model description of the corresponding request of its parent entity set.
- Entities to retrieve offer content objects from the system. These entities are described in more detail in the OData Resource section of this document.

Read-Only Value Help Entities

Resource	Description	Path
CommunicationMedium	Value help entity to retrieve available communication mediums.	/CommunicationMediums
ContentPosition	Value help entity to retrieve available content position values.	/ContentPositions
ContentType	Value help entity to retrieve available offer content type values	/ContentTypes
CustomerOrigin	Value help entity to retrieve available customer origins	/CustomerOrigins
ItemSourceType	Value help entity to retrieve available item source types for BasketObject and LeadingObject.	/ItemSourceTypes
ItemValueHelp	Value help entity to retrieve available items for BasketObject and LeadingObject.	/ItemValueHelps
Language	Value help entity to retrieve available language ISO codes	/Languages
MarketingLocation	Value help entity to retrieve available marketing location UUIDs.	/MarketingLocations

Resource	Description	Path
OfferRecommendationScenario	Value help entity to retrieve available offer recommendation scenario IDs.	/OfferRecommendationScenarios

#### Read-Only Navigational Property Entities

Resource	Description	Path
BasketObject	Navigational property of the Recommendation entity to provide BasketObjects in the deep-create call.	/Recommendations/BasketObjects
ContentTypeField	Navigational property of the ContentType entity to read the available fields of offer content types.	/ContentTypes/ContentTypeFields
ContextParam	Navigational property of the Recommendation entity to provide free context parameters in the deep-create call.	/Recommendations/ContextParams
LeadingObject	Navigational property of the Recommendation entity to provide LeadingObjects in the deep-create call.	/Recommendations/LeadingObjects
Result	Navigational property of the Recommendation entity to return results in the deep-create call.	/Recommendations/Results
AssignedCoupon	Navigational property of the Result entity to provide Coupon data in the deep-create call.	/Recommendations/Results/AssignedCoupon

#### Offer Content Retrieval Entities

Resource	Description	Path
OfferContent	Entity to retrieve offer content objects from the system using a GET call on the corresponding entity set OfferContents with search parameters provided via the complex type OfferContentSearch.	/OfferContents



Resource	Description	Path
Recommendation	Entity to retrieve offer content objects from the system using a POST call on the corresponding entity set Recommendations and providing the input parameters in a deep-create call using the navigational properties LeadingObjects, BasketObjects and ContextParams of the entity. The results are returned using the navigational property Results.	/Recommendations

## OData Resource: OfferContent

The entity to retrieve offer content objects from the system using a GET call on the corresponding entity set.

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/OfferContents`

**Permissions:** Business catalog role SAP\_COM\_CSR\_0021

### Operations

#### CRUD Operations

The entity supports only GET on its entity set.

HTTP Method	Operation	URI
<a href="#">GET</a>	Read a list of offer content objects that adhere to the provided search parameters.	/OfferContents

The retrieval of `OfferContent` entities in the OData service `CUAN_OFFER_DISCOVERY_SRV` is done by requesting the entity set `OfferContents` using a GET operation, using a `$filter` operator with the available parameters for the complex type `OfferContentSearch`. A direct retrieval of individual `OfferContent` entities using the key fields is not implemented. The following request parameters can be used as filter properties in the `$filter`-clause:

- `CustomerId`: The user or customer or consumer ID of the user logged on to the Web shop. If this parameter is not used in `$filter` or used with an empty string, the parameter `CustomerOriginId` is ignored.
- `CustomerOriginId`: The origin ID for interaction contacts defined in Customizing. If this parameter is not used in `$filter` or used with an empty string and the `CustomerId` is also used, the parameter value is defaulted to the delivered value `SAP_HYBRIS_CONSUMER` internally.
- `CommunicationMediumId`: The ID of a communication medium. If a `CommunicationMediumId` is passed to the OData service as a filter, only offer contents for that communication medium are retrieved.

- `LanguageId`: The ISO language code of the offer content. In a Web shop, this might correspond to the user's logon language. If no language is passed to the OData service, the result contains all available languages.
- `Position`: Position in the Web shop where offers are to be displayed, such as **Top** or **Bottom**. This information must have been entered for the offer content.
- `RecommendationScenarioId`: This field is not supported.
- `ContentMediumTypeId`: The ID of a content type, such as "01" for content type "Image". If this parameter is not used in `$filter` the value is defaulted to "01" internally. Only offer contents matching the specified content medium type ID are retrieved.
- `MarketingLocationKey`: One or more marketing location identifiers (UUIDs) to search for offers that either have no marketing location assigned or have one of the specified marketing locations assigned. If offers without marketing locations are requested, a filter on `MarketingLocationKey` for example "" must be supplied in the query. When not using the `MarketingLocationKey` property as filter, the returned offers can have zero to multiple marketing locations assigned.

## Extensibility and Offer Content Types

The OData service `CUAN_OFFER_DISCOVERY_SRV` supports both "Offer Header Data" and "Offer Content Data" extensibility using the *Custom Fields* app. For more information, see [Custom Fields for Offer Header and Offer Content](#). It is possible to read the values of extensibility fields when reading `OfferContent` entities. However, it is not possible to define filters for these extension fields.

For any content type defined in the app *Manage your Solution*, associated fields can be defined in the app *User Interface Adaptation*. For more information, see [User Interface Adaptation](#). The service provides the information about which fields belong to which offer content type in the following two ways:

- The value help entity type `ContentType` has a navigational property `AvailableFields` to read the list of available fields in the entity type `OfferContent`.
- Field-control fields are part of the entity type `OfferContent`. These fields control the visibility of the corresponding data field in the following way:
  - If the corresponding data field is part of the returned content type the value is 1 (meaning read-only)
  - If the corresponding data field is not part of the returned content type the value is 0 (meaning hidden)

## Search Behavior

The `CustomerId` and `CustomerOriginId` parameters can only accept a single value for the filter operation `$filter` with the operator `EQ`. Additional filters using these parameters are ignored. Other operators are ignored and set to `EQ`. If a range operator (for example `BT`) is used, the lower boundary value is used. The higher boundary value is ignored.

The search result contains 0-n `OfferContent` entities which are active on the date and time of the actual request and have only valid marketing location assignments. If a coupon is assigned to the offer, an offer is only returned if the redemption limit for the coupon has not been reached. The *Redemption Limit* and the *Redemption Limit for Each Contact* are taken into account. For more information, see [Manage Coupons](#).

An offer is active at a particular point in time if the status is *Released* and if the point in time is found during one of the offer visibility (valid or visible) ranges.

A marketing location assigned to an offer is valid with respect to this offer if, and only if the following is true:

- The marketing location is not closed (deleted).
- The offer end date, in UTC, is found during the validity period of the location.

The result set is sorted internally using a ranking that is based on the filterable properties `CustomerId`: Offers are assigned the highest ranking if the `CustomerId` is part of any assigned target group that is assigned to the offer. Conversely, offers with no target group are assigned the lowest ranking. These offers are valid for any customer.

Additional sorting parameters can be passed to the service using `$orderby`, but are applied to the internal sorting after the logic described above.

## OData Operation: GET OfferContent

### Request

**URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/OfferContents`

**Operation Type:** R (Read)

**HTTP Method:** *GET*

**Permissions:** Business catalog role SAP\_COM\_CSR\_0021

The OData API is only to be called using `$batch`, so that the query can be encrypted in the HTTP request body and to avoid URL overflows. A request to the OData service to retrieve up to ten `OfferContent` entities with filters on `CustomerId`, `CommunicationMedium Language` and `Position` could then be as follows:

### Example HTTP request for offer content retrieval

#### i Note

To improve readability, the following example HTTP requests and responses do not show all the details. Some metadata information is for example omitted in the JSON responses and URLs are shown without encoding. For example, spaces are not replaced by `%20`.

```
POST /sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/$batch

Request Headers:
accept          application/json
content-type    multipart/mixed;boundary=batch_1e29-6867-0e8e
Request Body:
--batch_1e29-6867-0e8e Content-Type: application/http Content-Transfer-Encoding:
binary
  GET OfferContents?$top=10&$filter=Search/CustomerId eq 'demo@hybris.com' and
  Search/CommunicationMediumId eq 'ONLINE_SHOP' and Search/LanguageId eq 'FR' and
  Search/Position eq 'TOP'
HTTP/1.1
Accept: application/json
Accept-Language: en-US
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0

--batch_1e29-6867-0e8e--
```

### Response

**Format:** JSON

```
{
```

```

"d": {
  "results": [{
    "Search": {
      "CustomerId": "",
      "CustomerOriginId": "",
      "CommunicationMediumId": "",
      "ContentMediumTypeId": "",
      "LanguageId": "",
      "Position": "",
      "RecommendationScenarioId": ""
    }
    "MarketingLocationKey": ""
  },
  "OfferId": "0000004711",
  "Offer": "Offer 4711",
  "CustomerId": "demo@hybris.com",
  "ContentId": "00001",
  "CommunicationMediumId": "ONLINE_SHOP",
  "CommunicationMedium": "Online-shop",
  "LanguageId": "FR",
  "Language": "French",
  "Position": "TOP",
  "ContentMediumTypeId": "01",
  "ContentMediumType": "Image",
  "ContentDescription": "Buy 2 get 1 free",
  "ContentSource": "http://assets.mycompany.com/offer4711.png",
  "TargetDescription": "Link for offer 4711",
  "TargetLink": "http://www.mycompany.com/offer4711.html"
},
{
  "Search": {
    "CustomerId": "",
    "CustomerOriginId": "",
    "CommunicationMediumId": "",
    "ContentMediumTypeId": "",
    "LanguageId": "",
    "Position": "",
    "RecommendationScenarioId": ""
  }
  "MarketingLocationKey": ""
},
  "OfferId": "0000004712",
  "Offer": "Offer 4712",
  "CustomerId": "",
  "ContentId": "00001",
  "CommunicationMediumId": "ONLINE_SHOP",
  "CommunicationMedium": "Online-shop",
  "LanguageId": "FR",
  "Language": "French",
  "Position": "TOP",
  "ContentMediumTypeId": "01",
  "ContentMediumType": "Image",
  "ContentDescription": "20% off for order values > 100€",
  "ContentSource": "http://assets.mycompany.com/offer4712.png",
  "TargetDescription": "Link for offer 4712",
  "TargetLink": "http://www.mycompany.com/offer4712.html"
}
  ]
}

```

This response contains two `OfferContent` entities. The first entity was found because the customer (with id `demp@hybris.com`) was found in a target group that is assigned to offer 4711 and both the language, the communication medium and the position of the entity match the search query. The second `OfferContent`

entity was found because the offer has no target group assigned and the language, the communication medium and the position match the corresponding filter values of the entity. The content of offer 4711 is ranked first according to the sorting rules described above.

## OData Resource: Recommendations

As of Release 1705, a new API has been introduced in the offer discovery service that provides enhanced capabilities to use in combination with the Offer Recommendation Intelligence (ORI).

This API is based on the new Recommendation entity in the offer discovery service and allows the caller to specify leading items, basket items and free context parameters to be used in the defined rules in an ORI model. Since this functionality could not be provided using the existing GET OfferContent API, this new API was introduced which uses a POST call on the Recommendations entity set. All input parameters are provided in the HTTP body of the request, similar to a deep create of an OData entity.

Remark: The GET OfferContent continues to work with the release 1705, with some minor changes regarding the search behavior when using a MarketingLocationKey filter and the sorting behavior of the result set.

**Resource Path:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/Recommendations`

**Permissions:** Business catalog role SAP\_COM\_CSR\_0021

### Operations

#### CRUD Operations

The entity supports only PUT on its entity set.

HTTP Method	Operation	URI
<i>POST</i>	Deep-create on the Recommendations entity set to retrieve offer content objects process by the ORI	<code>/Recommendations</code>

Input and output of the API is defined by the structure of the Recommendation entity (see also the \$metadata XML definition of the offer discovery service and the API hub documentation).

The Recommendation entity has three properties (CustomerId, CustomerOriginId and RecommendationScenarioId) that are used as input values when calling the API. To call the recommendation API for anonymous contacts, provide the CustomerID and CustomerOriginId parameters and use blank/empty ("") values. Additional input values for leading items, basket items and free context parameters are provided using navigation properties with the respective entity types LeadingObject, BasketObject and ContextParam.

The results are returned in the navigation property results. The entity result uses the same properties as the OfferContent entity for the GET OfferContent call and adds the property Score, which contains the score value for the offer content object as determined by the ORI and the external ID and origin for imported offer objects (InboundOriginIdExt and InboundOriginId).

The service contains two additional value help entities ItemSourceType and ItemValueHelp for the properties of the LeadingObject and BasketObject entities. These are defined in the service metadata using ValueList annotations, similar to the value help entities for the OfferContent/Search in the GET call of the API.

No value help entity is available for the the ContextParam entity. For a list of possible context parameter names, see the “Search Behavior” chapter of this section.

#### Example of the Recommendation entity as JSON structure:

##### Sample Code

```
Recommendation: {
  // -> IN: fields that were previously used in complex type
  OfferContentSearch to perform the ORI call.
  // CustomerId is now UserId
  // CustomerOriginId is now UserOriginId
  // CommunicationMediumId -> is derived from the recommendation scenario!
  // ContentMediumTypeId, LanguageId, Position, MarketingLocationKey ->
  moved into the ContextParams navigation property
  UserId: "",
  UserOriginId: "",
  RecommendationScenarioId: "", // value list annotation to
  OfferRecommendationScenarios

  // -> IN: new LeadingObject entity type (same internal structure as
  BasketObject)
  LeadingObjects: [{
    LeadingObjectType: "", // -> value list annotation to ItemSourceTypes
    LeadingObjectId: "" // -> value list annotation to ItemValueHelps
    (LeadingObjectType as input)
  }],

  // -> IN: new BasketObject entity type (same structure as LeadingObject)
  BasketObjects: [{
    BasketObjectType: "", // -> value list annotation to ItemSourceTypes
    BasketObjectId: "" // -> value list annotation to ItemValueHelps
    (BasketObjectType as input)
  }],

  // -> IN: new ContextParam entity type
  ContextParams: [{
    ContextId: 0 // Integer key
    Name: "",
    Value: ""
  }],

  // -> OUT: new Result entity type
  Results: [{
    OfferId: "",
    Offer: "",
    InboundOriginId: "",
    InboundOriginIdExt: "",
    ContentId: "",
    CommunicationMediumId: "",
    CommunicationMedium: "",
    LanguageId: "",
    Language: "",
    Position: "",
    ContentMediumTypeId: "",
    ContentMediumType: "",
    ContentDescription: "",
    ContentSource: "",
    TargetDescription: "",
    TargetLink: "",
    FC_ContentPosition: "",
    FC_ContentMediumTypeId: "",
    FC_ContentDescription: "",
    FC_ContentSource: "",
    FC_TargetDescription: "",
    FC_TargetLink: "",
    Score: "",
```

```
} ]  
}
```

## Extensibility and Result Entity Type

The Result entity responds in the same way as the OfferContent entity with regards to extensibility (see also the corresponding chapter above).

## Search Behavior

The POST Recommendations API responds similarly to the GET OfferContents API, with the following differences:

- The call is performed using an HTTP POST, providing all input parameters in the request body.
- To call the API, a valid X-CSRF-Token must be supplied in the request header (this token can be retrieved by sending a GET request to /sap/opu/odata/sap/CUAN\_OFFER\_DISCOVERY\_SRV with the HTTP header containing the attribute X-CSRF-Token with the value 'fetch').
- \$skip/\$top operations are not supported. The number of returned offer content objects in the Result property is restricted by the available number of valid objects in the system and the maximum number of results defined in the recommendation model.
- The API can only be used for use with ORI. Therefore, the RecommendationScenarioId is a mandatory input parameter.
- The chosen recommendation scenario must have a communication medium assigned. The communication medium is used during the selection of the relevant offer content. Using a recommendation scenario without a communication medium will lead to an error.
- LeadingObjects, BasketObjects and ContextParams are optional input parameters.
- The Results navigational property must be requested in the API call (see below for an example of calling the API)
- The ContentMediumTypeId, LanguageId, Position, MarketingLocationKey and CommunicationMediumId input parameters from the GET OfferContent API are provided using the ContextParams navigation property.
- When providing CommunicationMediumId in the ContextParams, the values are used in addition to the communication medium derived from the recommendation scenario.
- The available context parameters names are:
  - P\_LANGUAGE: for OfferContent/Search/LanguageId
  - P\_POSITION: for OfferContent/Search/Position
  - P\_COMM\_MEDIUM: for OfferContent/Search/CommunicationMediumId
  - P\_CONT\_MEDIUM\_TYPE: for OfferContent/Search/ContentMediumTypeId
  - P\_ML\_UUID32: for OfferContent/Search/MarketingLocationKey
  - P\_CURRENT\_LONGITUDE: parameter no longer supported
  - P\_CURRENT\_LATITUDE: parameter no longer supported
  - P\_WITH\_COUPON: parameter to filter offers with assigned coupons
    - Parameter value = 'X' delivers only offers with assigned coupons
    - Parameter value = '' delivers only offers without assigned coupons
    - Requests without parameter P\_WITH\_COUPON deliver offers with and without assigned coupons
  - P\_MARKETING\_AREA: parameter to restrict the offers based on their assigned marketing areaContext parameters that are not explicitly listed are ignored.

## OData Operation: POST Recommendations

**URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/Recommendations`

**Operation Type:** C (Create)

**HTTP Method:** *POST*

**Permissions:** Business catalog role SAP\_COM\_CSR\_0021

### Request

As previously mentioned, the Recommendation entity can only be called using an HTTP POST call. The call mimics an OData deep-create call. Other calls to the Recommendation entity are not supported. (such as GET entity/set, and so on).

#### i Note

To improve readability, the following example HTTP requests and responses do not show all the details. Some metadata information is for example omitted in the JSON responses and URLs are shown without encoding. For example, spaces are not replaced by %20

### Example

#### Sample Code

```
POST /sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/Recommendations

Request Headers:
accept          application/json
content-type    application/json
X-CSRF-Token   _oHVBAS9m4E95kbxqVV2ww==
Request Body:
{
  "UserOriginId": "ONLINE_SHOP",
  "UserId": "demo@hybris.com",
  "RecommendationScenarioId": "ORI_SCENARIO",
  "BasketObjects": [],
  "LeadingObjects": [],
  "ContextParams": [{
    "ContextId": 1,
    "Value": "EN",
    "Name": "P_LANGUAGE"
  }, {
    "ContextId": 2,
    "Value": "TOP",
    "Name": "P_POSITION"
  }],
  "Results": []
}
```

### Response

#### Sample Code

```
{
  "d": {
    "UserId": "",

```



```

    "UserOriginId": "",
    "RecommendationScenarioId": "",
    "BasketObject": null,
    "LeadingObjects": null,
    "ContextParams": null,
    "Results": {
      "results": [{
        "OfferId": "0000004711",
        "Offer": "Offer 4711",
        "CustomerId": "demo@hybris.com",
        "ContentId": "00001",
        "CommunicationMediumId": "ONLINE_SHOP",
        "CommunicationMedium": "Online-shop",
        "LanguageId": "FR",
        "Language": "French",
        "Postion": "TOP",
        "ContentMediumTypeId": "01",
        "ContentMediumType": "Image",
        "ContentDescription": "Buy 2 get 1 free",
        "ContentSource": "http://assets.mycompany.com/
offer4711.png",
        "TargetDescription": "Link for offer 4711",
        "TargetLink": "http://www.mycompany.com/offer4711.html",
        "Score": "1.00"
      }, {
        "OfferId": "0000004712",
        "Offer": "Offer 4712",
        "CustomerId": "",
        "ContentId": "00001",
        "CommunicationMediumId": "ONLINE_SHOP",
        "CommunicationMedium": "Online-shop",
        "LanguageId": "FR",
        "Language": "French",
        "Postion": "TOP",
        "ContentMediumTypeId": "01",
        "ContentMediumType": "Image",
        "ContentDescription": "20% off for order values > 100€",
        "ContentSource": "http://assets.mycompany.com/
offer4712.png",
        "TargetDescription": "Link for offer 4712",
        "TargetLink": "http://www.mycompany.com/offer4712.html",
        "Score": "0.80"
      }
    ]
  }
}

```

## Sample Code

Example request with coupon-parameter: "P\_WITH\_COUPON"

```
POST /sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/Recommendations
```

Request Headers:

```

accept          application/atom+xml
content-type    application/json
X-CSRF-Token   _oHVBAS9m4E95kxqVV2ww==
Request Body:

```

```

{
  "UserOriginId": "ONLINE_SHOP",
  "UserId": "demo@hybris.com",
  "RecommendationScenarioId": "ORI_SCENARIO",
  "BasketObjects": [],
  "LeadingObjects": [],
  "ContextParams": [

```

```

    "ContextId": 1,
    "Value": "EN",
    "Name": "P_LANGUAGE"
  }, {
    "ContextId": 2,
    "Value": "TOP",
    "Name": "P_POSITION"
  }, {
    "ContextId": 3,
    "Value": "X",
    "Name": "P_WITH_COUPON"
  }
],
"Results": [
  {
    "AssignedCoupon": {}
  }
]
}

```

Example response with coupon data

```

...
<title type="text">Coupons('COUPON_FOR_MOBILE')</title>
<updated>2017-09-08T14:29:37Z</updated>
<category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/
scheme" term="CUAN_OFFER_DISCOVERY_SRV.CouponEntityType"/>
<link title="CouponEntityType" rel="self"
href="Coupons('COUPON_FOR_MOBILE')"/>
-<content type="application/xml">
-<m:properties>
<d:Coupon>COUPON_FOR_MOBILE</d:Coupon>
<d:CouponType>SINGLE</d:CouponType>
<d:CouponName/>
<d:CouponDescription/>
</m:properties>
</content>
</entry>
</m:inline>
</link>
-<content type="application/xml">
-<m:properties>
<d:OfferId>0000000709</d:OfferId>
<d:Offer>Test_Coupon_Offer</d:Offer>
<d:InboundOriginIdExt/>
<d:InboundOriginId/>
<d:ContentId>00001</d:ContentId>
<d:CommunicationMediumId>MOBILE_APP</d:CommunicationMediumId>
<d:CommunicationMedium>Mobile Anwendung</d:CommunicationMedium>
<d:LanguageId>EN</d:LanguageId>
<d:Language>Englisch</d:Language>
<d:Position>Bottom</d:Position>
<d:ContentMediumTypeId>01</d:ContentMediumTypeId>
<d:ContentMediumType>Bild</d:ContentMediumType>
<d:ContentDescription>http://i.imgur.com/crDskHz.png</d:ContentDescription>
<d:ContentSource>http://i.imgur.com/crDskHz.png</d:ContentSource>
<d:TargetDescription/>
<d:TargetLink/>
<d:FC_ContentPosition>0</d:FC_ContentPosition>
<d:FC_ContentMediumTypeId>0</d:FC_ContentMediumTypeId>
<d:FC_ContentDescription>0</d:FC_ContentDescription>
<d:FC_ContentSource>0</d:FC_ContentSource>
<d:FC_TargetDescription>0</d:FC_TargetDescription>
<d:FC_TargetLink>0</d:FC_TargetLink>
<d:Score>1.00000</d:Score>
</m:properties>
</content>
</entry>
...

```

## Request Headers

### Recommendations Request Headers

Header	Required	Description
Content-Type	Yes	Recommended value: application/json
Accept	Yes	Recommended value: application/json
X-CSRF-Token	Yes	To be retrieved by the caller before calling the Recommendations API, for example using a GET call on <code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV</code> using the HTTP Header "X-CSRF-Token" with the value "fetch"

### Recommendations Response Headers

Header	Description
Content-Type	Returned value: application/json

### Recommendations Status and Error Codes

Code	Reason
201	Recommendations call was successfully processed.
400	There was an error processing the request. See the response for detailed error information.

## Discover Coupons

## OData Operation: GET GetCouponCode

**URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/GetCouponCode`

**Operation Type:** R (Read)

**HTTP Method:** *GET*

**Permissions:** Business catalog role SAP\_COM\_CSR\_0021

When Discovery and Recommendation returns Offer with Coupon information, then the user receives just the Coupon root information. To get a redeemable coupon code a second request is necessary. Therefore, this OData Function Import has been defined.

This OData API endpoint is used to fetch coupon codes for a coupon that was retrieved by a POST call on the Recommendations entity set using the "AssignedCoupon" navigational property in the request (see above).

In order to retrieve coupon codes for anonymous contacts, the parameters `UserId` and `UserOriginId` must be provided with empty values (`&UserId=""&UserOriginId=""`).

## i Note

Anonymous contacts can only receive coupon codes of coupons with contact relationship type "No Contact Assigned". For more information, see [Manage Coupons](#).

## Example Request

```
GET /sap/opu/odata/sap/CUAN_OFFER_DISCOVERY_SRV/GetCouponCode?
Coupon='COUPON_FOR_MOBILE'&UserId='demo@hybris.com'&UserOriginId='ONLINE_SHOP'
```

## Example Response (JSON)

### Sample Code

```
{
  "d" : {
    "CouponCode" : "CODE_1234",
    "EANCodeImageUrl" : "https://yourdomain.com/url_to_eancode/1234",
    "QRCodeImageURL" : "https://yourdomain.com/url_to_qrcode/1234",
    "ValidityStartDateTime" : "\/Date(1527235508173)\7",
    "ValidityEndDateTime" : "\/Date(1527717599000)\/",
    "CouponCodeSerialNumber" : "1234"
  }
}
```

## Request Headers

### Recommendations Request Headers

Header	Required	Description
Accept	No	Recommended value: application/json

### Recommendations Response Headers

Header	Description
Content-Type	Returned value: application/json

### Recommendations Status and Error Codes

Code	Reason
200	Recommendations call was successfully processed.
4xx	There was an error processing the request. See the response for detailed error information.

## OData Operation: POST GetCouponCodes

**URI:** `https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/GetCouponCodes`

**Operation Type:** R (Read)

**HTTP Method:** *POST*

**Permissions:** Business catalog role SAP\_COM\_CSR\_0021

This mass-enabled OData Coupon Code API is used to fetch redeemable coupon codes for multiple users. The number of users can be adjusted or may have to be reduced according to the system load, computing and storage capacity.

#### **i** Note

Anonymous contacts can only receive coupon codes of coupons with contact relationship type *No Contact Assigned*. For more information, see [Manage Coupons](#)

#### **Example Request**

POST /sap/opu/odata/sap/CUAN\_OFFER\_DISCOVERY\_SRV/GetCouponCodes

#### **Example Request Body (JSON)**

##### Sample Code

```
{
  "Coupon": "Coupon_1",
  "to_CouponCodeContacts": [
    {
      "UserId": " name.surname@company.com",
      "UserOriginId": "EMAIL"
    }
  ]
}
```

#### **Example Response (JSON)**

##### Sample Code

```
{
  "d" : {
    "Coupon" : "Coupon_1",
    "to_CouponCodeContacts" : {
      "results" : [
        {
          "Coupon" : "Coupon_1",
          "CouponCodeSerialNumber" : "SERIAL-NUMBER-007",
          "ValidityEndDateTime" : "\/Date(1610578799000)\/",
          "ValidityStartDateTime" : "\/Date(1606728581000)\/",
          "QRCodeImageUrl" : "https://yourdomain.com/url_to_qrcode/007",
          "EANCodeImageUrl" : "https://yourdomain.com/url_to_eancode/007",
          "CouponCode" : "SAMPLE-CODE-007",
          "UserOriginId" : "EMAIL",
          "UserId" : "name.surname@company.com"
        }
      ]
    }
  }
}
```

#### **Request Headers**

##### **Recommendations Request Headers**

Header	Required	Description
Accept	No	Recommended value: application/json

### Recommendations Response Headers

Response Header contains error messages with information about users which could not receive coupon codes. There could be many reasons, for example a user is not in the offer target group, a user has already redeemed the coupon code, the coupon, the coupon code or the offer is no longer valid.

Header	Description
Content-Type	Returned value: application/json

### Recommendations Status and Error Codes

Code	Reason
200	Coupon codes were retrieved successfully.
4xx	There was an error processing the request. See the response for detailed error information.

## 5.6.8 Coupons

Public OData API (API\_MKT\_COUPON\_SRV) for Coupons.

[Overview \[page 1026\]](#)

Entity Sets

- [Coupons \[page 1028\]](#)
- [CouponCodes \[page 1029\]](#)
- [CouponTexts \[page 1030\]](#)
- [CouponCodeUsages \[page 1030\]](#)
- [CouponCodeUsagelntactnCntcts \[page 1031\]](#)

### Overview



The public API for Coupons supports operations on the `Coupons` Business Object.

<b>OData Version</b>	2.0
<b>Root URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_COUPON_SRV</code>
<b>Service Metadata URI</b>	<code>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/API_MKT_COUPON_SRV/\$metadata</code>

<b>Authorizations</b>	The following business catalog is required: SAP_CEC_BC_MKT_API_COP_PC
<b>Communication Scenario ID</b>	SAP_COM_0317
Component for Incidents	CEC-MKT-OFM-CPM (Coupons)
	<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>i Note</b></p> <p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p> </div>
Field Extensibility Supported	Yes

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

Access Link	Remarks
<pre>https:// &lt;Server&gt;:&lt;Port&gt;/sap/opu/ odata/SAP/ API_MKT_CONTACT_SRV;v=0003/\$me tadata?sap-documentation=all</pre>	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Coupons Details Page</a> 	<p>General access to the <a href="#">Details</a> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>On the <a href="#">Details</a> page, click <a href="#">Download Specification</a> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Coupons API</a> 	General access link takes you directly to the <a href="#">Coupons</a> metadata file. One-time registration or logon is required.

### i Note

You can convert the XML file to an XML table to make it easier to read.

Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## Entity Sets

The Coupons OData API provides the following entities:

Entity Set	Description	Path
Coupons	This entity contains coupon data	/Coupons
CouponCodes	This entity contains the coupon codes of coupons	/CouponCodes
CouponTexts	This entity contains the coupon texts of coupons	/CouponTexts
CouponCodeUsages	This entity contains the coupon code usage of coupons	/CouponCodeUsages
CouponCodeUsageIntactnCntcts	This entity contains the contact and contact origin of a coupon code usage	/CouponCodeUsageIntactnCntcts

The service supports OData deep create functionality to create a tree of related entities in a single POST request. The service also supports `$batch` to group any number of arbitrary requests into one request.

You can view sample payloads and test the API at <https://api.sap.com>.

## Coupons

**Resource Path:** /Coupons

You can perform the following operations on the `Coupons` entity set:

Operations on Coupons entity set

HTTP Method	Description	Path
<i>GET</i>	Get a list of coupons	/Coupons? \$top=<Number_Of_Objects>
<b>i Note</b> <ul style="list-style-type: none"><li>The <code>\$top</code> parameter is mandatory.</li><li>You can get only 100 coupons with each request.</li></ul>		
	Get the details of a specific coupon	/Coupons(guid'<Coupon UUID>')
<i>POST</i>	Create a coupon	/Coupons



HTTP Method	Description	Path
<i>PUT</i>	Update a coupon	/Coupons (guid '<Coupon UUID>')
	<b>i Note</b> The CouponCanExceedOfferPeriod property can be set to TRUE only for a Released offer.	
<i>DELETE</i>	Delete a coupon	/Coupons (guid '<Coupon UUID>')

## CouponCodes

**Resource Path:** /CouponCodes

You can perform the following operations on the CouponCodes entity set:

Operations on CouponCodes entity set

HTTP Method	Description	Path
<i>GET</i>	Get all the coupon codes for a specific coupon	/Coupons (guid '<Coupon UUID>') / CouponCodes
	<b>i Note</b> <ul style="list-style-type: none"> <li>The <i>\$top</i> parameter is mandatory.</li> <li>You can get only 100 coupons codes with each request.</li> </ul>	
	Get the coupon codes of a specific coupon	/CouponCodes (guid '<Coupon Code UUID>')
	Get a list of coupon codes	/CouponCodes? \$top=<Number_Of_Objects>
	Get the coupon of a specific coupon code	/CouponCodes (guid '<Coupon Code UUID>') /to_Coupon
<i>POST</i>	Create a coupon code	/Coupons (guid '<Coupon UUID>') / to_CouponCode
<i>PUT</i>	Update a coupon code	/CouponCodes (guid '<Coupon Code UUID>')
<i>DELETE</i>	Delete a coupon code	/CouponCodes (guid '<Coupon Code UUID>')

## CouponTexts

**Resource Path:** /CouponTexts

You can perform the following operations on the CouponTexts entity set:

Operations on CouponTexts entity set

HTTP Method	Description	Path
<i>GET</i>	Get all the coupon texts	/Coupons(guid'<Coupon UUID>')/CouponTexts
	<b>i Note</b> <ul style="list-style-type: none"><li>If the path is /CouponTexts, the <i>\$top</i> parameter is mandatory.</li><li>You can get only 100 coupons texts with each request.</li></ul>	
	Get the coupon texts of a specific coupon	/CouponTexts(guid'<Coupon Text UUID>')
	Get a list of coupon texts	/CouponTexts?\$top=<Number_Of_Objects>
	Get the coupon of a specific coupon text	/CouponTexts(CouponUUID=guid'<Coupon UUID>',Language=guid'<Language>')/to_Coupon
<i>POST</i>	Create a coupon text	/Coupons(guid'<Coupon UUID>')/to_CouponText
<i>PUT</i>	Update a coupon text	/CouponTexts(guid'<Coupon Text UUID>')
<i>DELETE</i>	Delete a coupon text	/CouponTexts(guid'<Coupon Text UUID>')

## CouponCodeUsages

**Resource Path:** /CouponCodeUsages

You can perform the following operations on the CouponCodeUsages entity set:

## Operations on CouponCodeUsages entity set

HTTP Method	Description	Path
<i>GET</i>	Get all the coupon code usages for a coupon code	<code>/CouponCodes (guid'&lt;Coupon Code UUID&gt;')/to_CouponCodeUsage</code>
<div style="background-color: #f0f0f0; padding: 5px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>The <i>\$top</i> parameter is mandatory.</li> <li>You can get only 100 coupon code usages with each request.</li> </ul> </div>		
	Get a specific coupon code usage	<code>/CouponCodeUsages (guid'&lt;Coupon Code Usage UUID&gt;')</code>
	Get a list of coupon code usages	<code>/CouponCodeUsages? \$top=&lt;Number_Of_Objects&gt;</code>
	Get the coupon code for a specific coupon code usage	<code>/CouponCodeUsages (guid'&lt;Coupon Code Usage UUID&gt;')/to_CouponCode</code>

## CouponCodeUsageIntactnCntcts

**Resource Path:** `/CouponCodeUsageIntactnCntcts`

You can perform the following operations on the `CouponCodeUsageIntactnCntcts` entity set:

### Operations on CouponCodeUsageIntactnCntcts entity set

HTTP Method	Description	Path
<i>GET</i>	Get all the contacts and contact origins for a specific coupon code usage	<code>/CouponCodeUsages (guid'&lt;Coupon Code UUID&gt;')/to_Contacts</code>
<div style="background-color: #f0f0f0; padding: 5px;"> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>The <i>\$top</i> parameter is mandatory.</li> <li>You can get only 100 contacts and contact origins with each request.</li> </ul> </div>		
	Get the contact and contact origin for a specific interaction contact	<code>/ CouponCodeUsageIntactnCntcts (InteractionContactFacetUUID=guid'&lt;Interaction Contact Facet UUID&gt;', InteractionContactUUID=guid'&lt;Interaction Contact UUID&gt;')</code>
	Get a list of interaction contacts and contacts origins used in coupon code origins	<code>/CouponCodeUsageIntactnCntcts? \$top=&lt;Number_Of_Objects&gt;</code>

## Implementation Hints

When calling this API, please observe the following recommendations:

- It is not possible to update a specific coupon or any of its child objects (codes, texts) using parallel requests because each update request locks the coupon object and all its child objects. Therefore it is not possible to upload coupon codes for the same coupon in parallel requests. The requests must be serialized by the caller.
- The recommended maximum package size for uploading multiple coupon codes in a single \$batch request is 1,000. Larger package sizes can have a negative influence on other processes on the system. The preferred package size is 500 codes per \$batch request.
- The expected average throughput for uploading coupon codes using these package sizes is between 1,200 and 1,800 codes/minute.

## Function Imports

Function imports are used to perform custom operations on an entity, in addition to typical OData operations. This section contains function import payload examples for some functions of the process automation.

HTTP Method	Description	Path
<i>POST</i>	Release a coupon. Sets the status from <i>In Process</i> (01) to <i>Released</i> (02).	/ReleaseCoupon? CouponUUID=guid'<Coupon UUID>'
	Reject a coupon. Sets the status from <i>Released</i> (02) to <i>In Process</i> (01).	/RejectCoupon? CouponUUID=guid'<Coupon UUID>'
	Deletes all codes of a coupon. To be used for efficient deletion of all codes in large multi-code coupons.	/DeleteCodes? CouponUUID=guid'<Coupon UUID>'
	Trigger the replication of the coupon to an external system and request a number of coupon codes from the external system. 0 is a valid value for NumberOfCouponCodes and the value to be used for single-code coupons.	/ReplicateCoupon? CouponUUID=guid'<Coupon UUID>' &NumberOfCouponCodes=<Number of codes to be requested from external system>

## Related Information

<https://api.sap.com> 

## 5.7 Marketing Analytics

## 5.7.1 Import Analytical Data for Marketing Executive KPI

You can import analytical data from an external system for *Marketing Executive* for specific KPIs, for example, market share, net promoter score, or brand awareness, which come from external marketing research and survey institutes.

Note that the *Marketing Executive Dashboard* is obsolete as of release 2011. However you can use an OData API as described in the following sections to import the data and build custom CDS views, operational reports, or analytical stories..

### Using the Web Service

For the import, you can use the ODATA service `CUAN_ANALYTIC_FND`. The data model is structured in entity types as follows:

- `KPIImport`  
Provides administrative and logging information about the import
- `MarketingEffectivenessData`  
Enter the KPI data for the following KPIs:
  - Brand Awareness
  - Market Share
  - Net Promoter Score (NPS)
  - Leads
  - Return on Marketing Investment (ROMI)
  - Sales Forecast, Revenue
  - Opportunities
  - Sales Pipeline
  - Converted Pipeline
  - Pipeline Acceleration
- `WebDownloadData`  
Use it for KPI data about web downloads
- `WebVisits`  
Use it for KPI data about web visits

For the import of KPI Data, specify the parent entity `KPIImport`. Use the depending entity of `MarketingEffectivenessData`, `WebDownloads`, or `WebVisits` to list your KPI Data.

### Using the OData API

Use the OData API as follows:

- Request URI: `/sap/opu/odata/sap/CUAN_ANALYTIC_FND_SRV/KPIImports`
- Use HTTP Method: POST

In addition, consider the following:

- See the next section for an example of an HTTP request body in the JSON format. To use this example in the POST request, set the HTTP header parameter Content-Type = application/json and Accept = application/json. You can also use different supported body formats.
- If the CSRF Token is necessary, you can request it from the web service using the HTTP method GET with the HTTP request header parameter X-CSRF-Token = Fetch set.
- Provide the user, and the password in the HTTP request header parameters.

See the following source code in the JSON format about how to create two records for the analysis type "ROMI" – Return on Marketing Investment:

```
{
  "KPIImportGuid": "",
  "AnalysisTypeDescription": "",
  "FileName": "",
  "FileDescription": "",
  "ImportModeCode": "D",
  "ImportModeDescription": "",
  "ImportedById": "",
  "ImportedByName": "IMPORTUSER",
  "ImportedOn": "2015-06-10T00:00:00",
  "NumberOfImportedRecords": 0,
  "MarketingEffectivenessData": [
    {
      "AnalysisType": "ROMI",
      "Brand": "EST",
      "Audience": "Financial Services",
      "Country": "US",
      "Market": "",
      "Competitor": "",
      "ProductCategory": "",
      "Region": "",
      "Program": "",
      "Campaign": "",
      "SpendType": "",
      "CustomDimension1": "",
      "CustomDimension2": "",
      "CustomDimension3": "",
      "CustomDimension4": "",
      "CustomDimension5": "",
      "CustomDimension6": "",
      "CustomDimension7": "",
      "CustomDimension8": "",
      "CustomDimension9": "",
      "CustomDimension10": "",
      "KPIDate": "2015-04-01T00:00:00",
      "CurrentValue": "0.50",
      "TargetValue": "0",
      "Currency": "",
      "Factor": "1"
    },
    {
      "AnalysisType": "ROMI",
      "Brand": "MON",
      "Audience": "Financial Services",
      "Country": "FR",
      "Market": "",
      "Competitor": "",
      "ProductCategory": "",
      "Region": "",
      "Program": "",
      "Campaign": "",
      "SpendType": "",
      "CustomDimension1": "",

```

```

"CustomDimension2":"","
"CustomDimension3":"","
"CustomDimension4":"","
"CustomDimension5":"","
"CustomDimension6":"","
"CustomDimension7":"","
"CustomDimension8":"","
"CustomDimension9":"","
"CustomDimension10":"","
"KPIDate":"2015-04-01T00:00:00",
"CurrentValue":"0.50",
"TargetValue":"0",
"Currency":"","
"Factor":"1"
}
]
}

```

## Entity Type KPIImport

The entity type `KPIImport` describes the technical header of an import of marketing effectiveness data. The properties `KPIImportGuid`, `ImportedById`, `ImportedByName`, `ImportedOn`, and `NumberOfImportedRecords` are used for logging the external data request. Provide a new `KPIImportGuid` for every POST service request. Fill the `ImportedByName` property with a user ID. The `ImportedOn`, and `NumberOfImportedRecords` properties are maintained by default. The properties `FileName`, and `FileDescription` are ignored.

See the following table for the details of the structure of the entity type:

Property	Description	Edm Core Type	Max Length	Key	Nullable
<code>KPIImportGuid</code>	Guid created for every import request	Edm.String	32	X	
<code>AnalysisTypeId</code>	KPI specific Analysis Type	Edm.String	50		
<code>AnalysisTypeDescription</code>	To be ignored	Edm.String	50		
<code>FileName</code>	To be ignored	Edm.String	255		
<code>FileDescription</code>	To be ignored	Edm.String			
<code>ImportModeCode</code>	Fill in the Code "D" for Delta Mode(Update + Insert) and "F" for Full Mode(Delete + Insert)	Edm.String	1		
<code>ImportModeDescription</code>	To be ignored	Edm.String	60		
<code>ImportedById</code>	User Id	Edm.String	12		

Property	Description	Edm Core Type	Max Length	Key	Nullable
ImportedByName	User name	Edm.String	80		
ImportedOn	Will be defaulted - Date and time which should be logged for the current import. Example 1: 2002-10-10T17:00:00Z	Edm.DateTimeOffset			
NumberOfImportedRecords	Number of imported records will be calculated at import. Within the request, set it to "0" as it cannot be empty	Edm.Int32			FALSE

## Entity Type Marketing Effectiveness Data

See the following table for the details of the structure of the marketing effectiveness data entity type:

Property	Description	Edm Core Type	Max Length	Key
AnalysisType	KPI specific Analysis Type	Edm.String		X
Brand	Brand of the data record	Edm.String		X
Audience	Audience of the data record	Edm.String		X
Country	Country of the data record	Edm.String		X
Market	Market of the data record	Edm.String		X
Competitor	Competitor of the data record	Edm.String		X
ProductCategory	Product Category of the data record	Edm.String		X
Region	Region of the data record	Edm.String		X
Program	Program of the data record	Edm.String		X
Campaign	Campaign of the data record	Edm.String		X
SpendType	Spend Type of the data record	Edm.String		X
CustomDimension1	These attributes can be used for Customer specific dimensions which are not part of the standard delivery	Edm.String		X



Property	Description	Edm Core Type	Max Length	Key
CustomDimension2		Edm.String		X
CustomDimension3		Edm.String		X
CustomDimension4		Edm.String		X
CustomDimension5		Edm.String		X
CustomDimension6		Edm.String		X
CustomDimension7		Edm.String		X
CustomDimension8		Edm.String		X
CustomDimension9		Edm.String		X
CustomDimension10		Edm.String		X
KPIDate	Date being used to assign the record to the time dimension, for example, calendar quarter. Example 1: 2002-10-10T17:00:00Z	Edm.DateTime		X
CurrentValue	Represents numeric values with fixed precision and scale. Example 1: 2,345. Cannot be empty or null	Edm.Decimal	18	
TargetValue	Represents numeric values with fixed precision and scale. Example 1: 2,345. Cannot be empty or null	Edm.Decimal	18	
Currency	Should match the currency configured for SAP Marketing Cloud or it can be left empty	Edm.String		

Property	Description	Edm Core Type	Max Length	Key
Factor	Factor to quantify the weight of the data records for the calculation of the average value. A large value implies that the value of this record has a high impact on the average. Cannot be empty or null	Edm.Decimal		

## Entity Type Web Download Data

See the following table for the details of the structure of the web download data entity type:

Property	Description	Edm Core Type	Max Length	Key
Country	Country	Edm.String		X
Brand	Brand	Edm.String		X
KPIDate	Date being used to assign the record to the time dimension, for example, calendar quarter. Example 1: 2002-10-10T17:00:00Z	Edm.DateTime		X
PDFs	Number of PDFs	Edm.Int32		
Videos	Number of Videos	Edm.Int32		
Audios	Number of Audios	Edm.Int32		

## Entity Type Web Visit Data

See the following table for the details of the structure of the web visit data entity type:

Property	Description	Edm Core Type	Max Length	Key
Country	Country	Edm.String		X
Brand	Brand	Edm.String		X

Property	Description	Edm Core Type	Max Length	Key
KPIDate	Date being used to assign the record to the time dimension, for example, calendar quarter. Example 1: 2002-10-10T17:00:00Z	Edm.DateTime		X
Visits	Number of Visits	Edm.Int32		
UniqueVisitory	Number of unique visits	Edm.Int32		
PageViews	Number of Page views	Edm.Int32		

## AnalysisType

See the following table for the values of the `AnalysisType` property that correlates to each KPI.

KPI	AnalysisType Property	Other Recommended Properties
Brand Awareness	BRAND_AWARENESS	Country, Market, CustomDimensions, KPIDate, CurrentValue, Factor
Brand Awareness Competitor	BRAND_AWARENESS_COMPETITOR	Country, Market, Competitor, CustomDimensions, KPIDate, CurrentValue, Factor
Market Share	MARKET_SHARE	Country, Market, Brand, CustomDimensions, KPIDate, CurrentValue, Factor
Market Share Competitor	MARKET_SHARE_COMPETITOR	Country, Market, Brand, Competitor, CustomDimensions, KPIDate, CurrentValue, Factor
Net Promoter Score (NPS)	NET_PROMOTER_SCORE	Country, Market, CustomDimensions, KPIDate, CurrentValue, Factor
Return on Marketing Investment (ROMI)	ROMI	Country, Market, Audience, Brand, CustomDimensions, KPIDate, CurrentValue, Factor
Leads	DEMAND_GENERATION_LEADS	Country, Market, Audience, Brand, CustomDimensions, KPIDate, CurrentValue, TargetValue
Opportunities	DEMAND_GENERATION_OPPORTUNITIES	Country, Market, Audience, Brand, CustomDimensions, KPIDate, CurrentValue, TargetValue

KPI	AnalysisType Property	Other Recommended Properties
Pipeline Acceleration	DEMAND_ACCELERATION	Country, Market, Audience, Brand, CustomDimensions, KPIDate, CurrentValue, TargetValue
Sales Pipeline	PIPELINE_BUILD	Country, Market, Audience, Brand, CustomDimensions, KPIDate, CurrentValue, Targetvalue
Converted Pipeline	PIPELINE_COVERAGE	Country, Market, Audience, Brand, CustomDimensions, KPIDate, CurrentValue, TargetValue
Revenue	REVENUE	Country, Market, Audience, Brand, CustomDimensions, KPIDate, CurrentValue, TargetValue
Sales Forecast	SALES_FORECAST	Country, Market, Audience, Brand, CustomDimensions, KPIDate, CurrentValue, TargetValue
Web Downloads	WEB_DOWNLOADS_2	Brand, Country, KPIDate, PDFs, Videos, Audios
Web Visits	WEB_VISITS_2	Brand, Country, KPIDate, Visits, UniqueVisits, PageViews

In addition to the property `AnalysisType`, provide the data for the properties relevant for the KPIs.

The dimension Market is derived from the dimension Country. As an alternative, you can provide the dimension Market without an assignment to the dimension Country.

You can specify 10 custom attributes as `CUSTOM_FIELD1 ... CUSTOM_FIELD10`, to enable custom dimensions for the drill-down and filtering in analytical reporting. Specify the values with a particular date. The date is relevant for the year-to-date calculation. For example, if the current date is 30.6, the analysis time frame of the year-to-date in 2015 is 01.01.2015 - 30.6.2015. In this case, only values with a date before or on 30.6. are considered.

## 5.8 Marketing Planning and Performance

## 5.8.1 Actual and Committed Spend Data

You can upload actual and committed spend data from an external ERP system into SAP Marketing Cloud using the `CUAN_ACTUAL_IMPORT_SRV` OData service.

### Prerequisites

- You have assigned the *Marketing - Business Data Integration* communication scenario to your communication user in *Maintain Communication Users*.
- You have maintained the SICF node for the `CUAN_ACTUAL_IMPORT_SRV` external service name in SAP Marketing Cloud (which is either a back-end system or a remote gateway system, depending on your setup).

### Mass Import

The OData protocol allows the import or update of one object record (one spend item) only. To achieve the mass create and mass update of records, a dummy entity (import header) is created by deep insert. You perform an insert on the `ImportHeader` entity and create actual spend items as subnodes of the import header. The metadata of the service is read by means of the OData call:

- Request URL: `/sap/opu/odata/sap/CUAN_ACTUAL_IMPORT_SRV/$metadata`
- HTTP Method: Get

### Structure of `CUAN_ACTUAL_IMPORT_SRV` OData Service

The `CUAN_IMPORT_SRV` OData service consists of the following entity sets and entity types:

Entity Set	Entity Type	Entity Type Description
CampaignActual	CampaignActual	Actual Spends
ImportHeader	ImportHeader	Technical Import Message Header

#### ImportHeader Entity Type

The `ImportHeader` entity type describes the technical header of an import of actual spends. The property `ID` is used as an external reference number to identify the associated application log.

#### ImportHeader Meta Information

Property	Description	Edm Core Type	Maximum Length	Mandatory	Key
ID	Used as an external reference number in the application log	Edm.String	100	No	Yes

#### CampaignActual Entity Type

The `CampaignActual` entity type contains all attributes that are required to upload actual spend data. If the values for a combination of source ID, campaign ID, spend type, spend item ID, and reference date are being uploaded for the first time, the corresponding values for this combination (amounts and currency) from the HTTP request are uploaded. If they are being uploaded a subsequent time, the corresponding values for this combination are updated with the values from the HTTP request.

Actual and committed spend can be created at any level. You are responsible to create data at the level that is relevant for your campaign and your business processes. For example, you can create data at the following levels:

- Campaign level
- Campaign and spend level
- Campaign and spend item level

#### CampaignActual Meta Information

Property	Description	Edm Core Type	Maximum Length	Mandatory	Key
SourceID	The source ID indicates the origin of the spend information.	Edm.String	30	No	Yes
CampaignID	Campaigns have been created in the <a href="#">Campaigns</a> , <a href="#">Programs</a> , or <a href="#">Marketing Plans</a> applications.	Edm.String	10	Yes	Yes
SpendType	Spend types have been defined in the <a href="#">Spend Type</a> configuration application.	Edm.String	10	No	Yes

Property	Description	Edm Core Type	Maximum Length	Mandatory	Key
SpendItemID	If specified, the spend item ID must exist in the campaign and the spend type must match. If no spend type is specified, the spend type will be derived from the spend item of the campaign.	Edm.String	10	No	Yes
ReferenceDate	The date that is used for currency conversion. The reference date must be a valid date in the following format "YYYY-MM-DDT00:00:00".	Edm.DateTime		Yes	Yes
WBSElementID	The WBS Element ID is only used internally by SAP	Edm.String	24	No	No
Currency	Currency for the spend amount.	Edm.String	5	Yes	No
ActualSpend	Actual costs that have been incurred from marketing activities.	Edm.Decimal	15,2	No	No
CommittedSpend	The amount of already known spend based on existing requests and orders for an item, for example, from a purchasing system.	Edm.Decimal	15,2	No	No

## Importing Actual Spend Data Using OData Service

For the input file, the following applies:

- If the values for a combination of source ID, campaign ID, spend type, spend item ID, and reference date are being uploaded for the first time, the corresponding values for this combination (amounts and currency) from the local file are uploaded. If they are being uploaded a subsequent time, the corresponding values for this combination are updated with the values from the new local file.
- A period (.) must be used to separate decimals in amounts.
- If a field is optional and you do not want to include a value for it in the file, you still need to insert a comma (,) in place of the excluded value.
- Actual and committed spend can be uploaded at any level. You are responsible for uploading data at the level that is relevant for your campaign and your business processes. For example, you can upload data at the following levels:
  - Campaign level
  - Campaign and spend level
  - Campaign and spend item level

If there is invalid data in the local file, no actual and committed spend amounts are uploaded and saved.

To upload actual spend data, the `ImportHeader` and `CampaignActual` entity types are required.

### Example

- Request URL: `/sap/opu/odata/sap/cuan_actual_import_srv/ImportHeaders`
- HTTP Method: Post
- Example request:

```
{
  "Id": "Example-01",
  "CampaignActuals": [
    {
      "SourceId": "",
      "CampaignId": "40144",
      "SpendType": "",
      "SpendItemId": "1",
      "ReferenceDate": "2015-11-30T00:00:00",
      "Currency": "USD",
      "ActualSpend": "50000.00",
      "CommittedSpend": "30000.00"
    },
    {
      "SourceId": "",
      "CampaignId": "40144",
      "SpendType": "",
      "SpendItemId": "2",
      "ReferenceDate": "2015-11-30T00:00:00",
      "Currency": "USD",
      "ActualSpend": "10000.00",
      "CommittedSpend": "10000.00"
    }
  ]
}
```

### Success Message



After a successful upload of the actual spend data, the status of the HTTP response is 201 Created and the following success message is provided:

```
<?xml version="1.0" encoding="utf-8"?>
<entry xml:base="https://wdciw1.wdf.sap.corp:11100/sap/opu/odata/sap/
cuan_actual_import_srv/" xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:d="http://
schemas.microsoft.com/ado/2007/08/dataservices">
  <id>https://wdciw1.wdf.sap.corp:11100/sap/opu/odata/sap/
cuan_actual_import_srv/ImportHeaders('Example-01')</id>
  <title type="text">ImportHeaders('Example-01')</title>
  <updated>2015-12-01T13:44:21Z</updated>
  <category term="CUAN_ACTUAL_IMPORT_SRV.ImportHeader" scheme="http://
schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <link href="ImportHeaders('Example-01')" rel="self" title="ImportHeader"/>
  <link href="ImportHeaders('Example-01')/CampaignActuals" rel="http://
schemas.microsoft.com/ado/2007/08/dataservices/related/CampaignActuals"
type="application/atom+xml;type=feed" title="CampaignActuals">
    <m:inline/>
  </link>
  <content type="application/xml">
    <m:properties>
      <d:Id>Example-01</d:Id>
      <d:Timestamp>2015-12-01T13:44:21.5450340</d:Timestamp>
      <d:UserName>USERNAME</d:UserName>
      <d:SourceSystemType/>
      <d:SourceSystemId/>
    </m:properties>
  </content>
</entry>
```

## Error Handling

If the request fails due to some errors, the complete HTTP request is rejected and errors must be corrected before uploading again. In case of errors, the status of the HTTP response is 400 Bad Request.

You can also find all messages using the [Application Log](#) application, entering CUAN\_IMPORT as a category and CUAN\_ACTUAL\_IMPORT as a subcategory.

## Component for Incidents

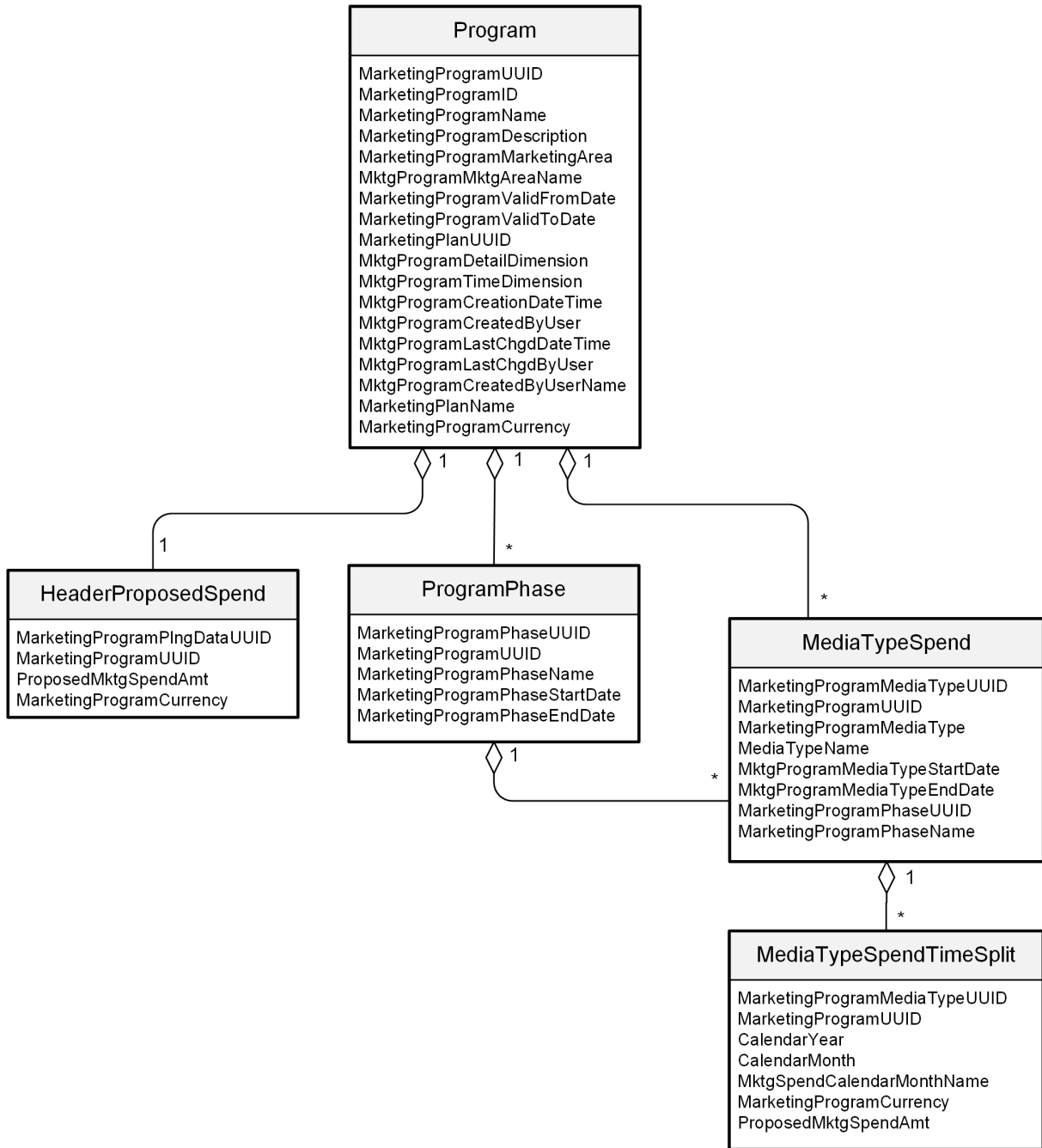
CEC-MKT-MSM

## 5.8.2 Marketing Programs

Public OData API (`API_MKT_PROGRAM`) for reading, updating and creating marketing program data.

## Entity Data Model

The following diagram shows the entity data model for program and its media type spend.



## Technical Data

Name of the Service

API\_MKT\_PROGRAM

Authorizations

The following business catalog role is required:

SAP\_BCR\_CEC\_MKT\_API\_PGM\_PC



Communication Scenario ID	SAP_COM_0320
Component for Incidents	CEC-MKT-PGM
	<div style="background-color: #f0f0f0; padding: 5px;"> <p><b>i Note</b></p> <p>Not to be used for HTTP errors. For more information, see <a href="#">HTTP Response Status Codes [page 408]</a>.</p> </div>
OData Version	2.0
Root URI	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_PROGRAM_SRV/
Service Metadata URI	https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_PROGRAM_SRV/\$metadata
Field Extensibility Supported	Only the Program entity is enabled for extension.

## Technical Field Documentation

You can access detailed technical documentation for the API fields by downloading a metadata file in one of the following ways:

### i Note

You can convert the XML file to an XML table to make it easier to read.

Access Link	Remarks
https://<Server>:<Port>/sap/opu/odata/SAP/API_MKT_PROGRAM_SRV;v=0002/\$metadata?sap-documentation=all	Only for internal access. You need to provide the server and port names.
<a href="#">Marketing - Programs Details Page</a> 	<p>General access to the <i>Details</i> page of the service on SAP API Hub. One-time registration is required for first-time users.</p> <ol style="list-style-type: none"> <li>On the <i>Details</i> page, click <i>Download Specification</i> and download as EDMX.</li> <li>Specify which application you want to use to open the EDMX file type.</li> </ol>
<a href="#">Marketing Programs API</a> 	General access to the <i>Marketing Programs</i> metadata file. One-time registration or logon is required.


Please note the meaning of the following values in the file:

Metadata Value in XML	Meaning When FALSE
sap:updatable	read-only field
sap:creatable	read-only field
nullable	mandatory field

## 5.8.2.1 Basic Concepts

The public API for Marketing Programs `API_MKT_PROGRAM_SRV` supports operations on the `Marketing Program Business Object`.

### Processing Info

Batch requests are submitted as a single `HTTP POST` request to the `$batch` endpoint of a service as described in [OData-URI ]. The batch request must contain a header parameter `content-type`, specifying the value `multipart/mixed` and `boundary=batch`.

A `PATCH (MERGE)` request updates only the properties indicated in the request body and leaves everything untouched that was not mentioned. All properties that are not to be changed can be omitted. The transmitted properties are merged with the data already stored in SAP Marketing Cloud.

### Best Practices

You can view sample payloads and test the API at <https://api.sap.com/> .

### Error Messages

If the OData service is not accessible, for example due to missing authorization, or because the system is not available, a corresponding HTTP status code is returned.

If the OData service is accepted by the gateway component in SAP Marketing Cloud, the HTTP status code 201 or 204 is returned.

### Field Extensibility

In addition to the pre-delivered attributes, you can add customer-specific fields using the *Custom Fields* app. The `Program` entity is the only entity that is enabled for extension. For more information, see [Custom Fields](#).

Please enable the Data Source under UIs and Reports: `API_MKT_PROGRAM_SRV_0002`.

## 5.8.2.2 Structure of OData Service `API_MKT_PROGRAM`

This document describes the structure of the Public OData API `API_MKT_PROGRAM_SRV`. Make sure you read the Basic Concepts topic before you start.

### Entity Sets

The Programs OData API provides the following entities:

Entity	Description	Path
Program	This entity contains program header data. Read and create functionalities are supported.	<code>/Programs</code>
HeaderProposedSpend	This entity provides information about the proposed spend entered at the program plan level. Only read functionalities are supported.	<code>/HeaderProposedSpends</code>
ProgramPhase	This entity provides information about the name and the validity period of the program phase. Only read functionalities are supported.	<code>/ProgramPhases</code>
MediaTypeSpend	This entity provides information about the different media types used for the program, their validity period, and their possible assignment to a program phase. Create, read, update functionalities are supported.	<code>/MediaTypeSpends</code>
MediaTypeTimeSplit	This entity provides the proposed spend for each media type on a monthly level. Read and update functionalities are supported.	<code>/MediaTypeTimeSplits</code>

## Program

**Resource Path:** /Programs

You can perform the following operations on the Program entity set:

Operations on Program Entity Set

HTTP Method	Description	Path
<i>GET</i>	Get a list of programs.	/Programs
	Get specific program information.	/Programs(<MarketingProgramUUID>)
<i>POST</i>	Create a program.	/Programs
<i>PATCH</i>	Edit program entities.	/Programs(<MarketingProgramUUID>)

## HeaderProposedSpend

**Resource:** /HeaderProposedSpend

You can perform the following operations on the HeaderProposedSpend entity set:

Operations on HeaderProposedSpend Entity Set

HTTP Method	Description	Path
<i>GET</i>	Get a list of header proposed spend.	/HeaderProposedSpends
	Get specific header proposed spend information.	/HeaderProposedSpends(<MarketingProgramPngDataUUID>)
	Get header proposed spend for a specific program.	/Programs(<MarketingProgramUUID>)/ProgramHeaderProposedSpend
<i>PATCH</i>	Edit header proposed spend entities.	/HeaderProposedSpends(<MarketingProgramPngDataUUID>)

## ProgramPhase

**Resource Path:** /ProgramPhases

You can perform the following operations on the ProgramPhases entity set:

#### Operations on ProgramPhases Entity Set

HTTP Method	Description	Path
<i>GET</i>	Get a list of program phases.	/ProgramPhase
	Get specific program phase information.	/ProgramPhases(<MarketingProgramPhaseUUID>)
	Get program phases for a specific program.	/Programs(<MarketingProgramUUID>)/Phase

## MediaTypeSpend

**Resource Path:** /MediaTypeSpends

You can perform the following operations on the `MediaTypeSpends` entity set:

#### Operations on MediaTypeSpends Entity Set

HTTP Method	Description	Path
<i>GET</i>	Get a list of media type spends.	/MediaTypeSpends
	Get specific media type spend information.	/MediaTypeSpends(<MarketingProgramMediaTypeUUID>)
	Get media type spends for a specific program.	/Programs(<MarketingProgramUUID>)/ProgramMediaTypeSpend
<i>PUT / PATCH / MERGE</i>	Update specific media type spend.	/MediaTypeSpends(<MarketingProgramMediaTypeUUID>)
<i>POST</i>	Create media type spend for a specific program.	/MediaTypeSpends

## MediaTypeTimeSplit

**Resource Path:** /MediaTypeTimeSplits

You can perform the following operations on the `MediaTypeTimeSplits` entity set:

#### Operations on MediaTypeTimeSplits Entity Set

HTTP Method	Description	Path
<i>GET</i>	Get a list of media type time splits.	/MediaTypeTimeSplits

HTTP Method	Description	Path
	Get media type time split information for a specific year and month.	/MediaTypeTimeSplits(<MarketingProgramMediaTypeUUID>,<CalendarYear>,<CalendarMonth>)
	Get media type time splits for a specific media type.	/MediaTypeSpend(<MarketingProgramMediaTypeUUID>)/TimeSplit
<i>PUT/PATCH/MERGE</i>	Update media type time splits for a specific media type.	/MediaTypeTimeSplits(<MarketingProgramMediaTypeUUID>,<CalendarYear>,<CalendarMonth>)

## 5.8.2.3 Payload Examples

The following examples show how you can use the Programs API.

### Update Monthly Proposed Spend

#### MERGE and PATCH

The following example is a request without batch:

##### Sample Code

```
MediaTypeTimeSplits(MarketingProgramMediaTypeUUID=guid'6C0B84B7-5523-1EE7-A7B9-2CDC0AF8FECB',CalendarYear='2041',CalendarMonth='01')
```

```
{
  "ProposedMktgSpendAmt"      : "3000.00",
  "MarketingProgramCurrency"  : "USD"
}
```

The following example is a request with batch:

##### Sample Code

```
--batch_d3de-e5db-e865
Content-Type: multipart/mixed; boundary=changeset_5cda-58b3-eea0
--changeset_5cda-58b3-eea0
Content-Type: application/http
Content-Transfer-Encoding: binary
MERGE
MediaTypeTimeSplits (MarketingProgramMediaTypeUUID=guid'6C0B84B7-5523-1EE7-A7B9-2CDC0AF8FECB',CalendarYear='2041',CalendarMonth='01') HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
```



```

sap-cancel-on-close: true
Content-Type: application/json
Content-Length: 250
{
  "ProposedMktgSpendAmt"      : "10000.00",
  "MarketingProgramCurrency"  : "USD"
}
--changeset_5cda-58b3-eea0--
--batch_d3de-e5db-e865--

```

## MERGE

The following example is a request with batch:

### Sample Code

```

--batch_d3de-e5db-e865
Content-Type: multipart/mixed; boundary=changeset_5cda-58b3-eea0
--changeset_5cda-58b3-eea0
Content-Type: application/http
Content-Transfer-Encoding: binary
MERGE
MediaTypeTimeSplits (MarketingProgramMediaTypeUUID=guid'6C0B84B7-5523-1EE7-
A7B9-2CDC0AF8FECB', CalendarYear='2041', CalendarMonth='01') HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
Content-Type: application/json
Content-Length: 250
{
  "ProposedMktgSpendAmt"      : "60000.00",
  "MarketingProgramCurrency"  : "USD"
}
--changeset_5cda-58b3-eea0
Content-Type: application/http
Content-Transfer-Encoding: binary
MERGE
MediaTypeTimeSplits (MarketingProgramMediaTypeUUID=guid'6C0B84B7-5523-1EE7-
A7B9-2CDC0AF8FECB', CalendarYear='2041', CalendarMonth='02') HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
Content-Type: application/json
Content-Length: 250
{
  "ProposedMktgSpendAmt"      : "8000.00",
  "MarketingProgramCurrency"  : "USD"
}
--changeset_5cda-58b3-eea0--
--batch_d3de-e5db-e865--

```

## PUT

The following example is a request without batch:

### Sample Code

```

{

```

```
"MarketingProgramMediaTypeUUID" : "6c0b84b7-5523-1ee7-a7b9-2cdc0af8fecb",
"MarketingProgramUUID" : "6c0b84b7-5523-1ee7-a7b9-2cdc0af89ecb",
"CalendarYear" : "2041",
"CalendarMonth" : "01",
"ProposedMktgSpendAmt" : "30000.00",
"MarketingProgramCurrency" : "USD"
}
```

## Create Program

### POST

The following example is a request without batch:

#### Sample Code

```
{
  "MarketingProgramName" : "New Product Launch A",
  "MarketingProgramMarketingArea" : "GLOBAL",
  "MarketingProgramValidFromDate" : "2041-01-01T00:00:00",
  "MarketingProgramValidToDate" : "2041-02-06T00:00:00",
  "MarketingProgramCurrency" : "USD"
}
```

## Create MediaTypeSpend

### POST

The following example is a request without batch:

#### Sample Code

```
{
  "MarketingProgramUUID" : "6c0b84b7-5523-1ed8-acc-2c4f6f612f18",
  "MarketingProgramMediaType" : "TV",
  "MktgProgramMediaTypeStartDate" : "2018-09-08T00:00:00",
  "MktgProgramMediaTypeEndDate" : "2018-09-20T00:00:00"
}
```

## Update MediaTypeSpend

### MERGE and GET

The following example is a request with batch:

#### Sample Code

```
--batch_d3de-e5db-e865
```

```

Content-Type: multipart/mixed; boundary=changeset_5cda-58b3-eea0
--changeset_5cda-58b3-eea0
Content-Type: application/http
Content-Transfer-Encoding: binary
MERGE MediaTypeSpends (MarketingProgramMediaTypeUUID=guid'6C0B84B7-5523-1EE7-
A7B9-2CDC0AF8FECB') HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
Content-Type: application/json
Content-Length: 250
{
  "MarketingProgramMediaType" : "PRINT",
  "MktgProgramMediaTypeStartDate" : "2041-01-02T00:00:00",
  "MktgProgramMediaTypeEndDate" : "2041-02-08T00:00:00"
}
--changeset_5cda-58b3-eea0--
--batch_d3de-e5db-e865
Content-Type: application/http
Content-Transfer-Encoding: binary
GET MediaTypeSpends (guid'6C0B84B7-5523-1EE7-A7B9-2CDC0AF8FECB') HTTP/1.1
HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
--batch_d3de-e5db-e865--

```

## MERGE and PATCH

The following example is a request without batch:

### Sample Code

```

{
  "MarketingProgramMediaType" : "RADIO",
  "MktgProgramMediaTypeStartDate" : "2041-01-03T00:00:00",
  "MktgProgramMediaTypeEndDate" : "2041-02-07T00:00:00"
}

```

## PUT

The following example is a request without batch:

### Sample Code

```

{
  "MarketingProgramMediaTypeUUID" : "6c0b84b7-5523-1ee7-a7b9-2cdc0af8fecb",
  "MarketingProgramUUID" : "6c0b84b7-5523-1ee7-a7b9-2cdc0af89ecb",
  "MarketingProgramMediaType" : "RADIO",
  "MktgProgramMediaTypeStartDate" : "2041-01-03T00:00:00",
  "MktgProgramMediaTypeEndDate" : "2041-02-07T00:00:00",
  "MarketingProgramPhaseUUID" : "6c0b84b7-5523-1ee7-a7b9-2cdc0af8decb"
}

```

## Get MediaTypeSpend

### GET

The following example is with batch:

#### Sample Code

```
--batch_d3de-e5db-e865
Content-Type: application/http
Content-Transfer-Encoding: binary
GET MediaTypeSpends(guid'6C0B84B7-5523-1EE7-A7B9-2CDC0AF8FECB') HTTP/1.1
HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
--batch_d3de-e5db-e865
Content-Type: application/http
Content-Transfer-Encoding: binary
GET MediaTypeSpends(guid'6C0B84B7-5523-1EE7-A7B9-2CF31AC53ECB') HTTP/1.1
HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
--batch_d3de-e5db-e865
Content-Type: application/http
Content-Transfer-Encoding: binary
GET MediaTypeSpends(guid'6C0B84B7-5523-1EE7-A7B9-2CF31ABADECB') HTTP/1.1
HTTP/1.1
sap-contextid-accept: header
Accept: application/json
Accept-Language: en
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
sap-cancel-on-close: true
--batch_d3de-e5db-e865--
```

## Get all MediaTypeSpend for a Phase

### GET

The following is an example without batch:

#### Sample Code

```
{
  "d": {
    "results": [
      {
        "MarketingProgramMediaTypeUUID": "01234567-89ab-
cdef-0123-456789abcdef",
        "MarketingProgramUUID": "01234567-89ab-cdef-0123-456789abcdef",
        "MarketingProgramMediaType": "string",
```



**Objects.** Once you have a communication user assigned to this communication arrangement, the required data can be imported into the custom business object.

### Call OData Service

You find the URL of your Custom Business Object's OData service in your communication arrangement. You can call the metadata document by adding `$metadata` through the URL. You insert data by using the POST method. To update data use the PATCH method and to delete data use the DELETE method. If you want to upload multiple data sets send a batch request.

## Related Information

<https://blogs.sap.com/2017/05/12/usage-of-odata-service-of-custom-business-object/>

## 5.10 Business Users

The following synchronous inbound SOAP services are provided for setting up business users in SAP Marketing Cloud.

- `MANAGEBUSINESSUSERIN`
- `QUERYBUSINESSUSERIN`

### 5.10.1 Business User

Technical name: `MANAGEBUSINESSUSERIN`

This synchronous inbound SOAP service enables you to create, update, and delete business users from your external data sources, such as an identity management system. Deleting business users doesn't mean you've actually deleted them yet. The user assigned to the business user is deleted and the `MarkedForArchivingIndicator` has been set. This is the prerequisite for the ILM process that physically deletes business users.

You can assign business role IDs to the users at the node `Role`.

We recommend processing blocks of 10 users to a maximum of 100 users. Otherwise, the target system may time out.

This service supports the business users Employee (BUP003) and Agency User (AGC001).

#### Caution

This service directly influences the data and authorizations of business users. Changes are effective immediately in the target system.

Make sure to maintain only those authorizations that are intended for what a user needs to do in the system. Not doing so can cause security issues.

## Service Request

The service is structured into the following two top-level nodes:

### Message Header (`MessageHeader`)

The service message header is not in use in this service.

### Business User (`BusinessUser`)

The service nodes contain the service's business data.

#### Note

In the following table, attributes are marked in blue.

Nodes and Fields for the BusinessUser Node

Node or Field	Description	Maximum Field Length	Cardinality
<code>PersonExternalID</code>	Person External ID  Mandatory for business partner category role BUP003 (Employee) at creation.	60	0..1
<code>PersonID</code>	Person ID  At least one of the person IDs is mandatory.	10	0..1
<code>PersonUUID</code>	Person UUID  At least one of the person IDs is mandatory.	36	0..1
<code>BusinessPartnerRoleCode</code>	Business Partner Role Code  Only business partner role code BUP003 (Employee) is supported.  This field is mandatory.	6	0..1

Node or Field	Description	Maximum Field Length	Cardinality	
MarkedForArchivingIndicator	Mark for Archiving  Set to <b>True</b> : <ul style="list-style-type: none"> <li>The business user will be archived</li> <li>The <code>actionCode [1] for User</code> must be set to 02</li> </ul> Set to <b>False</b> : <ul style="list-style-type: none"> <li>The business user will be re-activated (Undo Archive)</li> <li>The <code>actionCode [1] for User</code> must be set to 02</li> </ul>		0..1	
ValidityPeriod	StartDate  Format:  YYYY-MM-DD  By default, the system date is set.		0..1	
	EndDate  Format:  YYYY-MM-DD  By default, 9999-12-31 is set.		0..1	
PersonalInformation	FormOfAddress	Form of address	4	0..1
Cardinality: 0..1	FirstName	First name	40	0..1
	LastName	Last name	40	0..1
		This field is mandatory.		
	PersonFullName	Person full name	80	0..1



Node or Field	Description	Maximum Field Length	Cardinality	
AcademicTitle	Academic title	4	0..1	
CorrespondenceLanguage	Correspondence language	9	0..1	
MiddleName	Middle name	40	0..1	
AdditionalLastName	Additional last name	40	0..1	
BirthName	Birth name	40	0..1	
NickName	Nick name	40	0..1	
Initials	Initials	10	0..1	
AcademicSecondTitle	Academic second title	4	0..1	
LastNamePrefix	Last name prefix	4	0..1	
LastNameSecondPrefix	Last name second prefix	4	0..1	
NameSupplement	Name supplement	4	0..1	
actionCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• 01 - Create</li> <li>• 02 - Update</li> <li>• 03 - Delete</li> </ul> <p>Mandatory if [2] is not set and personal information data are given.</p>	2	optional	
<b>User (only for Cloud)</b> Cardinality: 0..1	UserName	User name/Alias	40	0..1
	LogonLanguageCode	Logon language	9	0..1

Node or Field	Description	Maximum Field Length	Cardinality
DateFormatCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• 1 - DD.MM.YYYY (Gregorian Date)</li> <li>• 2 - MM/DD/YYYY (Gregorian Date)</li> <li>• 3 - MM-DD-YYYY (Gregorian Date)</li> <li>• 4 - YYYY.MM.DD (Gregorian Date)</li> <li>• 5 - YYYY/MM/D D (Gregorian Date)</li> <li>• 6 - YYYY-MM-DD (Gregorian Date, ISO 8601)</li> <li>• 7 - GYY.MM.DD (Japanese Date)</li> <li>• 8 - GYY/MM/DD (Japanese Date)</li> <li>• 9 - GYY-MM-DD (Japanese Date)</li> <li>• A - YYYY/MM/D D (Islamic Date 1)</li> <li>• B - YYYY/MM/D D (Islamic Date 2)</li> </ul>	2	0..1

Node or Field	Description	Maximum Field Length	Cardinality
	<ul style="list-style-type: none"> <li>C - YYYY/MM/D (Iranian Date)</li> </ul>		
DecimalFormatCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>1.234.567,89</li> <li>X - 1,234,567.89</li> <li>Y - 1 234 567,89</li> </ul>	2	0..1
TimeZoneCode	Time zone	10	0..1
TimeFormatCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>0 - 24 Hour Format (Example: 12:05:10)</li> <li>1 - 12 Hour Format (Example: 12:05:10 PM)</li> <li>2 - 12 Hour Format (Example: 12:05:10 pm)</li> <li>3 - Hours from 0 to 11 (Example: 00:05:10 PM)</li> <li>4 - Hours from 0 to 11 (Example: 00:05:10 pm)</li> </ul>	2	0..1
LockedIndicator	Locked indicator	5	0..1

Node or Field		Description	Maximum Field Length	Cardinality
ValidityPeriod	StartDate	Format: YYYY-MM-DD If no start date is maintained for the User, the StartDate for the BusinessUser is entered.		1
Cardinality: 1				
	EndDate	Format: YYYY-MM-DD If no EndDate is maintained, it is set to 9999-12-31.		1
Role	RoleName	Role name	40	1
Cardinality: 0..unbounded	actionCode	You can use the following values: <ul style="list-style-type: none"> <li>01 - Create</li> <li>03 - Delete</li> </ul> Mandatory if [6] is not set and role name data is given.	2	optional
	actionCode	You can use the following values: <ul style="list-style-type: none"> <li>01 - Create</li> <li>02 - Update</li> <li>03 - Delete</li> </ul> Mandatory if [3] is not set and user data (UserName and Role) are given.	2	optional
	[6] roleListCompleteTransmissionIndicator	CTI for the Role node		optional

Node or Field		Description	Maximum Field Length	Cardinality
UserAssignment	UserID	User ID	12	1
<b>(only for on-premise)</b>	actionCode	You can use the following values: <ul style="list-style-type: none"> <li>• 01 - Create</li> <li>• 02 - Update</li> <li>• 03 - Delete</li> </ul> Mandatory if [4] is not set and User ID data are given.	2	optional
Cardinality: 0..1				
WorkplaceInformation	EmailAddress	Email address	241	0..1
Cardinality: 0..1	PhoneInformation	Phone type <ul style="list-style-type: none"> <li>• B - Business</li> <li>• C - Cell</li> </ul> One set of phone information per phone type supported.	1	1
Cardinality: 0..2	CountryDialingCode	Country dialing code <p>Used for both phone types.</p>	10	0..1
	PhoneNumberAreaID	Phone number area code <p>Used for phone type B only.</p>	10	0..1
	PhoneNumberSubscriberID	Phone number subscriber ID <p>Used for both phone types.</p>	30	0..1
	PhoneNumberExtension	Phone number extension <p>Used for phone type B only.</p>	10	0..1

Node or Field	Description	Maximum Field Length	Cardinality
actionCode	You can use the following values: <ul style="list-style-type: none"> <li>• 01 - Create</li> <li>• 02 - Update</li> <li>• 03 - Delete</li> </ul> Mandatory if [7] is not set and phone data is given.	2	optional
FunctionalTitleName	Functional title name	40	0..1
Department	Department name	40	0..1
RoomNumber	Room number	10	0..1
Building	Building name	10	0..1
actionCode	You can use the following values: <ul style="list-style-type: none"> <li>• 01 - Create</li> <li>• 02 - Update</li> <li>• 03 - Delete</li> </ul> Mandatory if [5] is not set and workplace information data is given.	2	optional
[7] phoneInformationListCompleteTransmissionIndicator	CTI for the PhoneInformation node		optional
[1] actionCode	You can use the following values: <ul style="list-style-type: none"> <li>• 01 - Create</li> <li>• 02 - Update</li> <li>• 03 - Delete</li> </ul> This attribute is mandatory.	2	optional
[2] personalInformationListCompleteTransmissionIndicator	CTI for the PersonalInformation node		optional

Node or Field	Description	Maximum Field Length	Cardinality
[3] userListCompleteTransmissionIndicator	CTI for the User node		optional
[4] userAssignmentListCompleteTransmissionIndicator	CTI for the UserAssignment node		optional
[5] workplaceInformationListCompleteTransmissionIndicator	CTI for the WorkplaceInformation node		optional

## Sample Payload

### Sample Code

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aba="http://sap.com/xi/ABA">
  <soapenv:Header/>
  <soapenv:Body>
    <aba:BusinessUserBundleMaintainRequest_sync>
      <!--1 or more repetitions:-->
      <BusinessUser actionCode="01"
personalInformationListCompleteTransmissionIndicator="false"
userListCompleteTransmissionIndicator="false"
userAssignmentListCompleteTransmissionIndicator="false"
workplaceInformationListCompleteTransmissionIndicator="false">
        <PersonExternalID>Muster01</PersonExternalID>
        <BusinessPartnerRoleCode>BUP003</BusinessPartnerRoleCode>
        <PersonalInformation actionCode="01">
          <FormOfAddress>0002</FormOfAddress>
          <FirstName>Max</FirstName>
          <LastName>Muster</LastName>
          <PersonFullName>Prof. Dr. Max Muster</PersonFullName>
          <AcademicTitle>0002</AcademicTitle>

          <CorrespondenceLanguage>D</CorrespondenceLanguage>
          <MiddleName>Michael</MiddleName>
          <AcademicSecondTitle>0001</AcademicSecondTitle>
          <BirthName>Milli</BirthName>
          <NickName>Maxi</NickName>
          <LastNamePrefix>0001</LastNamePrefix>
        </PersonalInformation>
        <User actionCode="01"
roleListCompleteTransmissionIndicator="false">
          <!--Optional:-->
          <UserName>MAXMUSTER01</UserName>
          <LogonLanguageCode>DE</LogonLanguageCode>
          <LockedIndicator>>false</LockedIndicator>
          <Role actionCode="01">
            <RoleName>SAP_BR_MANAGER</RoleName>
          </Role>
          <Role actionCode="01">
            <RoleName>SAP_BR_BPC_EXPERT</RoleName>
          </Role>
        </User>
        <WorkplaceInformation actionCode="01"
phoneInformationListCompleteTransmissionIndicator="true">
          <EmailAddress>Max.Muster01@Test.com</EmailAddress>

```

```

        <PhoneInformation actionCode="01">
          <PhoneType>C</PhoneType>
          <CountryDialingCode>+49</CountryDialingCode>
          <PhoneNumberSubscriberID>0160123456</
PhoneNumberSubscriberID>
        </PhoneInformation>
        <PhoneInformation actionCode="01">
          <PhoneType>B</PhoneType>
          <CountryDialingCode>+49</CountryDialingCode>
          <PhoneNumberAreaID>06227</PhoneNumberAreaID>
          <PhoneNumberSubscriberID>7</PhoneNumberSubscriberID>
          <PhoneNumberExtension>12345</PhoneNumberExtension>
        </PhoneInformation>
        <FunctionalTitleName>TESTER</FunctionalTitleName>
        <Department>QUALITY</Department>
        <RoomNumber>C1.23</RoomNumber>
        <Building>WDF01</Building>
      </WorkplaceInformation>
    </BusinessUser>
    <BusinessUser actionCode="01"
personalInformationListCompleteTransmissionIndicator="false"
userListCompleteTransmissionIndicator="false"
userAssignmentListCompleteTransmissionIndicator="false"
workplaceInformationListCompleteTransmissionIndicator="false">
      <PersonExternalID>MINIMUSTER01</PersonExternalID>
      <BusinessPartnerRoleCode>BUP003</BusinessPartnerRoleCode>
      <PersonalInformation actionCode="01">
        <FormOfAddress>0001</FormOfAddress>
        <FirstName>Mini</FirstName>
        <LastName>Muster</LastName>
        <PersonFullName>Prof. Dr. Mini Muster</PersonFullName>
        <AcademicTitle>0002</AcademicTitle>
        <CorrespondenceLanguage>D</CorrespondenceLanguage>
        <AcademicSecondTitle>0001</AcademicSecondTitle>
        <LastNamePrefix>0001</LastNamePrefix>
      </PersonalInformation>
      <User actionCode="01"
roleListCompleteTransmissionIndicator="false">
        <!--Optional:-->
        <UserName>MINIMUSTER01</UserName>
        <LogonLanguageCode>DE</LogonLanguageCode>
        <LockedIndicator>>false</LockedIndicator>
        <Role actionCode="01">
          <RoleName>SAP_BR_MANAGER</RoleName>
        </Role>
        <Role actionCode="01">
          <RoleName>SAP_BR_BPC_EXPERT</RoleName>
        </Role>
      </User>
      <WorkplaceInformation actionCode="01"
phoneInformationListCompleteTransmissionIndicator="true">
        <EmailAddress>Mini.Muster01@Test.com</EmailAddress>
        <PhoneInformation actionCode="01">
          <PhoneType>C</PhoneType>
          <CountryDialingCode>+49</CountryDialingCode>
          <PhoneNumberSubscriberID>0160123456</
PhoneNumberSubscriberID>
        </PhoneInformation>
        <PhoneInformation actionCode="01">
          <PhoneType>B</PhoneType>
          <CountryDialingCode>+49</CountryDialingCode>
          <PhoneNumberAreaID>06227</PhoneNumberAreaID>
          <PhoneNumberSubscriberID>7</PhoneNumberSubscriberID>
          <PhoneNumberExtension>12345</PhoneNumberExtension>
        </PhoneInformation>
        <FunctionalTitleName>TESTER</FunctionalTitleName>
        <Department>QUALITY</Department>
        <RoomNumber>C1.23</RoomNumber>

```



```

        <Building>WDF01</Building>
    </WorkplaceInformation>
</BusinessUser>
</aba:BusinessUserBundleMaintainRequest_sync>
</soapenv:Body>
</soapenv:Envelope>

```

## Service Response

You receive a confirmation message response for each bundle of business users you send. If the service request is processed, a confirmation message is sent. This contains crucial information provided by the fields `PersonExternalID`, `PersonID`, and `PersonUUID` for each business user of the bundle.

The following table provides an overview of the response structure for the `BusinessUser` service node.

Field or Node	Description	Maximum Field Length	Cardinality	
<code>PersonExternalID</code>	Person External ID	60	0..1	
<code>PersonID</code>	Person ID	10	0..1	
<code>PersonUUID</code>	Person UUID	36	0..1	
Log Cardinality: 1	<code>BusinessDocumentProcessingResultCode</code>	Not in use	2	0..1
	<code>MaximumLogItemSeverityCode</code>	If several messages are stored for a business user, the maximum of all received severity codes the most severe level will be shown.	1	0..1
Item Cardinality: 0..unbounded	<code>TypeID</code>	Message number	40	0..1
	<code>CategoryCode</code>	Not in use	15	0..1
	<code>SeverityCode</code>	Severity code definition: <ul style="list-style-type: none"> <li>1 - Information</li> <li>2 - Warning</li> <li>3 - Error</li> </ul>	1	0..1

Field or Node		Description	Maximum Field Length	Cardinality
	Note	Contains the message texts.	200	1
	WebURI	Not in use		0..1

## Error Codes

Error Code	Description
104	<p>Combination of Ext. ID &amp;1 and ID &amp;2 inconsistent. Processing cancelled.</p> <p>PersonExternalID and PersonID have a 1:1 relationship. Enter thePersonID that corresponds with the PersonExternalID.</p>
105	<p>Combination of Ext. ID &amp;1 and UUID &amp;2 inconsistent.</p> <p>Person ExternalID and PersonUUID have a 1:1 relationship.</p> <p>Enter thePersonUUID that corresponds with the PersonExternalID</p>

## Constraints

This service does not support:

- Service Performer (BBP005) business users
- Freelancer (BBP010) business users

## Additional Information

Business User 

### i Note

For more information about the API, choose the [Details](#) tab on the SAP API Business Hub.

## 5.10.2 Business User - Read

Technical name: QUERYBUSINESSUSERIN

This synchronous inbound SOAP service enables you to provision users from your external data source such as an identity management system.

## Service Request

The service is structured into the following two top-level nodes:

### Business User (`BusinessUser`)

The service node contains the search parameters.

Nodes and Fields for the BusinessUser Node

Field or Node	Description	Maximum Field Length	Cardinality
<code>PersonExternalIdInterval</code> Cardinality: 0..unbounded	You can use the following values: <ul style="list-style-type: none"> <li>1- Equal No upper boundary value must be set.</li> <li>3 - Between Upper boundary value is mandatory.</li> <li>6 - Lower than Upper boundary value is optional.</li> <li>7 - Lower equal Upper boundary value is optional.</li> <li>8 - Greater than Upper boundary value is optional.</li> <li>9 - Greater equal Upper boundary value is optional.</li> </ul> This field is mandatory if <code>LowerBoundaryPersonExtId</code> is set.	1	1
<code>LowerBoundaryPersonExtId</code>	Employee name	60	0..1

Field or Node		Description	Maximum Field Length	Cardinality
	UpperBoundaryPersonExtId		60	0..1
PersonIDInterval Cardinality: 0..unbounded	IntervalBoundaryTypeCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• 1- Equal No upper boundary value must be set.</li> <li>• 3 - Between Upper boundary value is mandatory.</li> <li>• 6 - Lower than Upper boundary value is optional.</li> <li>• 7 - Lower equal Upper boundary value is optional.</li> <li>• 8 - Greater than Upper boundary value is optional.</li> <li>• 9 - Greater equal Upper boundary value is optional.</li> </ul> <p>This field is mandatory if LowerBoundaryPersonId is set.</p>	1	1
	LowerBoundaryPersonId		10	0..1
	UpperBoundaryPersonId		10	0..1

Field or Node		Description	Maximum Field Length	Cardinality
BusinessPartnerRoleCodeInterval	IntervalBoundaryTypeCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>1- Equal No upper boundary value must be set.</li> <li>3 - Between Upper boundary value is mandatory.</li> <li>6 - Lower than Upper boundary value is optional.</li> <li>7 - Lower equal Upper boundary value is optional.</li> <li>8 - Greater than Upper boundary value is optional.</li> <li>9 - Greater equal Upper boundary value is optional.</li> </ul> <p>This field is mandatory if LowerBoundaryBusinessPartnerRoleCode is set.</p>	1	1
	LowerBoundaryBusinessPartnerRoleCode	Only business partner role code BUP003 (Employee) is supported.	6	0..1
MarketForArchivingIndicator	IntervalBoundaryTypeCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>True</li> <li>False</li> </ul>		1
	LowerBoundaryMarkedForArchivingIndicator		1	0..1

Field or Node		Description	Maximum Field Length	Cardinality
UserIdInterval	IntervalBoundaryTypeCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• 1- Equal No upper boundary value must be set.</li> <li>• 3 - Between Upper boundary value is mandatory.</li> <li>• 6 - Lower than Upper boundary value is optional.</li> <li>• 7 - Lower equal Upper boundary value is optional.</li> <li>• 8 - Greater than Upper boundary value is optional.</li> <li>• 9 - Greater equal Upper boundary value is optional.</li> </ul> <p>This field is mandatory if LowerBoundaryUserId is set.</p>	1	1
	LowerBoundaryUserId		12	0..1
	UpperBoundaryUserId		12	0..1

Field or Node		Description	Maximum Field Length	Cardinality
UserNameInterval	IntervalBoundaryTypeCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• 1- Equal No upper boundary value must be set.</li> <li>• 3 - Between Upper boundary value is mandatory.</li> <li>• 6 - Lower than Upper boundary value is optional.</li> <li>• 7 - Lower equal Upper boundary value is optional.</li> <li>• 8 - Greater than Upper boundary value is optional.</li> <li>• 9 - Greater equal Upper boundary value is optional.</li> </ul> <p>This field is mandatory if LowerBoundaryUserName is set.</p>	1	1
Cardinality: 0..unbounded				
	LowerBoundaryUserName		40	0..1
	UpperBoundaryUserName		40	0..1

Field or Node		Description	Maximum Field Length	Cardinality
FirstNameInterval	IntervalBoundaryTypeCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• 1- Equal No upper boundary value must be set.</li> <li>• 3 - Between Upper boundary value is mandatory.</li> <li>• 6 - Lower than Upper boundary value is optional.</li> <li>• 7 - Lower equal Upper boundary value is optional.</li> <li>• 8 - Greater than Upper boundary value is optional.</li> <li>• 9 - Greater equal Upper boundary value is optional.</li> </ul> <p>This field is mandatory if LowerBoundaryFirstName is set.</p>	1	1
	LowerBoundaryFirstName		35	0..1
	UpperBoundaryFirstName		35	0..1



Field or Node	Description	Maximum Field Length	Cardinality
LastNameInterval Cardinality: 0..unbounded	IntervalBoundaryTypeCode  <ul style="list-style-type: none"> <li>1- Equal No upper boundary value must be set.</li> <li>3 - Between Upper boundary value is mandatory.</li> <li>6 - Lower than Upper boundary value is optional.</li> <li>7 - Lower equal Upper boundary value is optional.</li> <li>8 - Greater than Upper boundary value is optional.</li> <li>9 - Greater equal Upper boundary value is optional.</li> </ul> <p>This field is mandatory if LowerBoundaryLastName is set.</p>	1	1
	LowerBoundaryLastName	40	0..1
	UpperBoundaryLastName	40	0..1
EmailAddressInterval Cardinality: 0..unbounded	IntervalBoundaryTypeCode  LowerBoundaryEmailAddress  UpperBoundaryEmailAddress	1  241  241	1  0..1  0..1

### Query Processing Conditions (QueryProcessingConditions)

The service nodes contain the service's business data.

## Fields for the QueryProcessingConditions Node

Field	Description	Maximum Field Length	Cardinality
QueryHitsTotalNumberIndicator	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• True</li> <li>• False (default)</li> </ul>		1
QueryHitsMaximumNumberValue	<p>Enter the maximum number of hits. If no value is entered, the default is automatically set to 1000.</p>	999999999	0..1
QueryHitsUnlimitedIndicator	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• True</li> <li>• False (default)</li> </ul> <p>Set <b>True</b> to get all data based on selection criteria.</p>		1
QueryLastReturnedObjectID	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• True</li> <li>• False (default)</li> </ul> <p>If <code>QueryHitsMaximumNumberValue</code> is set and more data is available, you can set this value to <b>True</b>.</p>		0..1

## Sample Payload

### Sample Code

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aba="http://sap.com/xi/ABA">
  <soapenv:Header/>
  <soapenv:Body>
    <aba:BusinessUserSimpleByElementsQuery_sync>
      <BusinessUser>
        <PersonIDInterval>
          <IntervalBoundaryTypeCode>1</IntervalBoundaryTypeCode>
          <!--Optional:-->
          <LowerBoundaryPersonID>9980035943</LowerBoundaryPersonID>
          <!--Optional:-->
        </PersonIDInterval>
        <BusinessPartnerRoleCodeInterval>
          <IntervalBoundaryTypeCode>1</IntervalBoundaryTypeCode>
          <!--Optional:-->
          <LowerBoundaryBusinessPartnerRoleCode>bup003</
LowerBoundaryBusinessPartnerRoleCode>
        </BusinessPartnerRoleCodeInterval>
      </BusinessUser>
    <QueryProcessingConditions>
      <!--Optional:-->
    </QueryProcessingConditions>
  </soapenv:Body>
</soapenv:Envelope>
```

```

    <QueryHitsMaximumNumberValue>1</QueryHitsMaximumNumberValue>
    <QueryHitsUnlimitedIndicator>>false</QueryHitsUnlimitedIndicator>
  </QueryProcessingConditions>
</aba:BusinessUserSimpleByElementsQuery_sync>
</soapenv:Body>
</soapenv:Envelope>

```

## Service Response

### Business User (BusinessUser)

#### Note

The fields below the node `User` will be filled.

Node or Field	Description	Maximum Field Length	Cardinality	
PersonExternalID	Person External ID	60	0..1	
PersonID	Person ID	10	1	
PersonUUID	Person UUID	36	1	
BusinessPartnerRoleCode	Business Partner Role Code	6	1	
MarkedForArchivingIndicator	<ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>		1	
ValidityPeriod	StartDate Format: YYYY-MM-DD		1	
Cardinality: 0..1	EndDate Format: YYYY-MM-DD		1	
PersonalInformation	FormOfAddress	Form of address	4	0..1
Cardinality: 0..1	FirstName	First name	40	0..1
	LastName	Last name	40	0..1
	PersonFullName	Person full name	80	0..1
	AcademicTitle	Academic title	4	0..1
	CorrespondenceLanguage	Correspondence language	9	0..1

Node or Field	Description	Maximum Field Length	Cardinality	
MiddleName	Middle name	40	0..1	
AdditionalLastName	Additional last name	40	0..1	
BirthName	Birth name	40	0..1	
NickName	Nick name	40	0..1	
Initials	Initials	10	0..1	
AcademicSecondTitle	Academic second title	4	0..1	
LastNamePrefix	Last name prefix	4	0..1	
LastNameSecondPrefix	Last name second prefix	4	0..1	
NameSupplement	Name supplement	4	0..1	
User	User ID	12	1	
Cardinality: 0..1	UserName	User name/Alias	40	1
	LogonLanguageCode	Logon language	9	0..1

Node or Field	Description	Maximum Field Length	Cardinality
DateFormatCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>• 1 - DD.MM.YYYY (Gregorian Date)</li> <li>• 2 - MM/DD/YYYY (Gregorian Date)</li> <li>• 3 - MM-DD-YYYY (Gregorian Date)</li> <li>• 4 - YYYY.MM.DD (Gregorian Date)</li> <li>• 5 - YYYY/MM/D D (Gregorian Date)</li> <li>• 6 - YYYY-MM-DD (Gregorian Date, ISO 8601)</li> <li>• 7 - GYY.MM.DD (Japanese Date)</li> <li>• 8 - GYY/MM/DD (Japanese Date)</li> <li>• 9 - GYY-MM-DD (Japanese Date)</li> <li>• A - YYYY/MM/D D (Islamic Date 1)</li> <li>• B - YYYY/MM/D D (Islamic Date 2)</li> </ul>	2	0..1

Node or Field	Description	Maximum Field Length	Cardinality
	<ul style="list-style-type: none"> <li>C - YYYY/MM/D (Iranian Date)</li> </ul>		
DecimalFormatCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>1.234.567,89</li> <li>X - 1,234,567.89</li> <li>Y - 1 234 567,89</li> </ul>	2	0..1
TimeZoneCode	Time zone	10	0..1
TimeFormatCode	<p>You can use the following values:</p> <ul style="list-style-type: none"> <li>0 - 24 Hour Format (Example: 12:05:10)</li> <li>1 - 12 Hour Format (Example: 12:05:10 PM)</li> <li>2 - 12 Hour Format (Example: 12:05:10 pm)</li> <li>3 - Hours from 0 to 11 (Example: 00:05:10 PM)</li> <li>4 - Hours from 0 to 11 (Example: 00:05:10 pm)</li> </ul>	2	0..1
LockedIndicator	Locked indicator	5	0..1

Node or Field	Description	Maximum Field Length	Cardinality
ValidityPeriod	StartDate	Format: YYYY-MM-DD  If no start date is maintained for the User, the StartDate for the BusinessUser is entered.	1
	EndDate	Format: YYYY-MM-DD  If no EndDate is maintained, it is set to 9999-12-31.	1
Role	RoleName	Role name	40
			1
			Cardinality: 0..unbounded
UserAssignment	UserID	User ID	12
			1
			Cardinality: 0..1
	UserName		40
			0..1
WorkplaceInformation	EmailAddress	Email address	241
			0..1
Cardinality: 0..1	PhoneInformation	PhoneType	Phone type
			1
		CountryDialingCode	Country dialing code
			10
			0..1
	PhoneNumberAreaID	Phone number area code	
			10
			0..1
	PhoneNumberSubscriberID	Phone number subscriber ID	
			30
			0..1
	PhoneNumberExtension	Phone number extension	
			10
			0..1
	FunctionalTitleName	Functional title name	
			40
			0..1
	Department	Department name	
			40
			0..1
	RoomNumber	Room number	
			10
			0..1

Node or Field	Description	Maximum Field Length	Cardinality
Building	Building name	10	0..1

#### Response Processing Conditions (ResponseProcessingConditions)

Field	Description	Maximum Field Length	Cardinality
HitsTotalNumberValue	Contains the number of users based on given criteria.	999999999	1
ReturnedQueryHitsNumberValue	Contains the number of found data sets for business users.	999999999	1
MoreHitsAvailableIndicator	The indicator is set if the query was limited to a number of hits, but more business user data sets are available based on the query.		1
LastReturnedObjectID	Displays the last row of the found results list, limited by the found hits or by the value given for QueryHitsMaximumNumberValue.	999999999	0..1

#### Log (Log)

If errors occur, the log contains the information shown in the table below:

Field or Node	Description	Maximum Field Length	Cardinality
BusinessDocumentProcessingResultCode		2	0..1
MaximumLogItemSeverityCode	If several messages are stored for a business user, the maximum of all dropped severity codes worst level will be shown.	1	0..1
Item	TypeID	Message number	40
Cardinality: 0..unbounded	CateoryCode	Not in use	15



Field or Node	Description	Maximum Field Length	Cardinality
SeverityCode	Severity code definition: <ul style="list-style-type: none"> <li>• 1 - Information</li> <li>• 2 - Warning</li> <li>• 3 - Error</li> </ul>	1	0..1
Note	Contains the message texts.	200	1
WebURI	Not in use		0..1

## Constraints

This service does not support:

- Service Performer (BBP005) business users
- Freelancer (BBP010) business users

## Additional Information

[Business User - Read](#) 

### i Note

For more information about the API, choose the [Details](#) tab on the SAP API Business Hub.

## 5.10.3 Business User - Read Metadata

Technical name: `QueryBusinessUserMetadataIn`

This service enables you to read metadata information from your external data source such as an identity management system for service Business User with this synchronous inbound service.

This service provides the search parameter `RoleCategory`, which restricts the result set of the metadata information of the service. The response includes the metadata information based on the given criteria. If errors occur, the log contains information about the severity code, the message number and message texts.

This service is available on the SAP API Business Hub, for more information see [APIs on SAP API Business Hub](#).

## Service Request

### Service Nodes

The service nodes contain the service's business data.

Node or Field	Description	Cardinality
BusinessPartnerRoleCategoryInterval Cardinality: 0..unbounded	IntervalBoundaryTypeCode  You can only use the following value: <ul style="list-style-type: none"><li>1 - Equal</li></ul>	1..1
	LowerBoundaryBusinessPartnerRoleCategoryCode  For example: BUP003	0..1

### Sample Payload

#### Sample Code

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aba="http://sap.com/xi/ABA">
  <soapenv:Header/>
  <soapenv:Body>
    <aba:BusinessUserMetaDataQuery_sync>
      <!--Zero or more repetitions:-->
      <BusinessPartnerRoleCategoryInterval>
        <IntervalBoundaryTypeCode>1</IntervalBoundaryTypeCode>
        <LowerBoundaryBusinessPartnerRoleCategoryCode>BUP003</
LowerBoundaryBusinessPartnerRoleCategoryCode>
      </BusinessPartnerRoleCategoryInterval>
    </aba:BusinessUserMetaDataQuery_sync>
  </soapenv:Body>
</soapenv:Envelope>
```

## Service Response

Service Node	Description	Link to Details
BusinessUserMetaData RoleCategoryDependentMetadata	This node contains all meta-data, which depends on the role category of a business user, such as the role category with its role, external ID category with its external ID type or relationship category.	<a href="#">RoleCategoryDependentMetadata [page 1088]</a>

Service Node	Description	Link to Details
CodeList	This node provides the available code lists for SAP specific codes. For example country/region code, academic title.	<a href="#">CodeList [page 1090]</a>
Log	This nodes displays occurred messages.	<a href="#">Log [page 1091]</a>

## Error Codes

Error Code	Message	Description
112	Interval Boundary Type Code &1 is not supported for BusinessPartnerRoleCategoryInterval.	You can only use the following value: <ul style="list-style-type: none"> <li>1 - Equal</li> </ul>
118	No data found by given search criteria.	To display all data, don't enter any value.

## Authentication Method

You can use the following authentication methods: User ID/password (Username Token), X.509 certificate (X509 Token) or Single Sign On using SAML (SAML Token).

## Constraints

Currently this SOAP service is only enabled for English.

## Additional Information

If you have any issues, report an incident for component CA-GTF-BUM.

### i Note

For more information about the API, choose the [Details](#) tab on the SAP API Business Hub.

## 5.10.3.1 RoleCategoryDependentMetaData

### Nodes and Fields

#### i Note

In the following table, field attributes are marked in blue.

RoleCategoryDependentMetaData

Node or Field	Description	Cardinality
BusinessPartnerRoleCategory	BusinessPartnerRoleCategoryCode For example: BUP003	1..1
Cardinality: 1..1	Description languageCode For example: Employee	0..unbounded
BusinessPartnerRole	BusinessPartnerRoleCode For example: BBP005	1..1
Cardinality: 1..1	Description languageCode DefaultIndicator Can be <i>true</i> or <i>false</i>	0..unbounded
BusinessPartnerExternalIDCategory	BusinessPartnerExternalIDCategoryCode For example: HCM030	1..1
Cardinality: 0..1	Description languageCode	0..unbounded
BusinessPartnerExternalID	BusinessPartnerExternalIDCode For example: HCM030	1..1
Cardinality: 0..unbounded	Description languageCode DefaultIndicator The value can be either <i>true</i> or <i>false</i>	0..unbounded
BusinessPartnerRelationshipCategory	BusinessPartnerRelationshipCategoryCode For example: BUR025	1..1
Cardinality: 0..1	Description languageCode	0..unbounded
Partner1_BusinessPartnerCategory	BusinessPartnerCategoryCode Can be: <ul style="list-style-type: none"> <li>2 - Organization</li> <li>3 - Group</li> </ul>	1..1
Cardinality: 0..unbounded	Description languageCode	0..unbounded
Partner2_BusinessPartnerCategory	BusinessPartnerCategoryCode Can be: <ul style="list-style-type: none"> <li>1 - Person</li> </ul>	1..1

Node or Field	Description		Cardinality	
	Cardinality: 0..unbounded	Description	languageCode	0..unbounded
NodeProperties	NodePath			For example: BUSINESS_USER
Cardinality: 0..unbounded	NodeProperty	NodePropertyCode		1.1
	Cardinality: 0..unbounded	Description	languageCode	Node property codes are: <ul style="list-style-type: none"> <li>• 01 - enabled</li> <li>• 02 - disabled</li> <li>• 03 - read only</li> </ul>
FieldProperties	NodePath			For example: <ul style="list-style-type: none"> <li>• BUSINESS_USER-USER-ROLE</li> <li>• BUSINESS_USER-USER-VALIDITY_PERIOD</li> <li>• BUSINESS_USER-USER_ASSIGNMENT</li> </ul>
0..unbounded				1.1
	Fieldname			Name of a field. 1.1

Node or Field	Description	Cardinality
FieldProperty Cardinality: 0..unbounded	FieldPropertyCode Node property codes are: <ul style="list-style-type: none"> <li>• 01 - enabled</li> <li>• 02 - disabled</li> <li>• 03 - read only</li> <li>• 04 - mandatory</li> <li>• 05 - enabled, read only for update</li> <li>• 06 - mandatory, read only for update</li> </ul>	1..1
	Description languageCode	0..unbounded

### i Note

To display more information about the service node on the SAP API Business Hub, choose [API References](#) and select the operation.

## 5.10.3.2 CodeList

### Nodes and Fields

### i Note

In the following table, field attributes are marked in blue.

CodeList

Node or Field	Description	Cardinality
FieldName	Name of the field.	1..1
FieldCodeList Cardinality: 0..unbounded	FieldValue This node provides the available code lists for SAP specific codes. For example country/	1..1

Node or Field	Description	Cardinality
FieldDescription	languageCode	region code, academic title.
		0..unbounded

### i Note

To display more information about the service node on the SAP API Business Hub, select the operation.

## 5.10.3.3 Log

### Nodes and Fields

### i Note

In the following table, field attributes are marked in blue.

Log

Node or Field	Description	Cardinality
BusinessDocumentProcessingResultCode		0..1
MaximumLogItemSeverityCode	If several messages are stored for a business user, the maximum of all dropped severity codes worst level will be shown.	0..1
Item	TypeID	Message number.
Cardinality: 0..unbounded	CategoryCode	Not in use.
	SeverityCode	Severity code definition: <ul style="list-style-type: none"> <li>1 - Information</li> <li>2 - Warning</li> <li>3 - Error</li> </ul>
	Note	Contains the message texts.
	WebURI	Not in use.
		0..1

### i Note

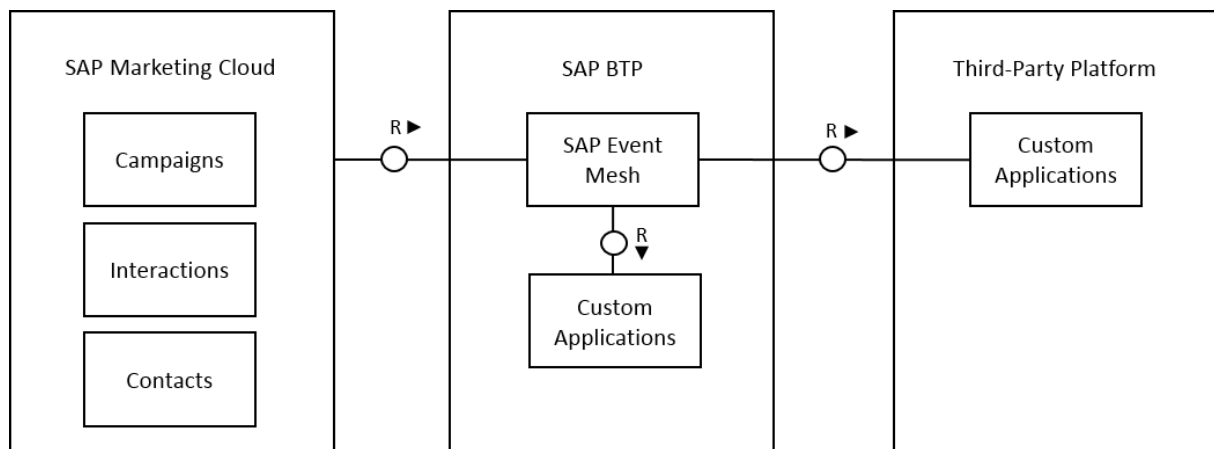
To display more information about the service node on the SAP API Business Hub, select the operation.

## 6 Business Event Handling

Subscribe to business events in SAP Marketing Cloud.

You can subscribe to business events triggered in SAP Marketing Cloud to receive notifications in extensions on SAP BTP. Then you can take follow-up actions on the events, if desired. When an event is triggered in SAP Marketing Cloud, there is at least one API available to fetch the necessary information.

The following diagram illustrates the systems required to use business event handling in SAP Marketing Cloud.



You can use the SAP-managed deployment service for the Kubernetes cluster to set up the SAP BTP, Kyma runtime. For more information, see [Kyma Environment](#).

### Prerequisites

- You have connected SAP Marketing Cloud to SAP Event Mesh and have enabled the events in SAP Marketing Cloud. For more information, see [Integrating Enterprise Event Enablement](#).
- You have configured SAP Event Mesh and set up a message queue to consume the events being triggered in SAP Marketing Cloud. For more information, see [Using Event Mesh](#).

For details about configuring the business events and APIs with SAP Marketing Cloud and extensions on SAP BTP, see the following blog: [Use SAP BTP, Kyma runtime to extend SAP Marketing Cloud](#)

#### i Note

The SAP blog isn't part of the official documentation of SAP Marketing Cloud and some of the information may be outdated.

### Example

A customer in B2B Marketing, running email campaigns in SAP Marketing Cloud, wants to use business events on interactions. They want to notify the responsible sales person that a contact has opened or clicked through



a recent email, which was supporting their sales activities in a given opportunity. The sales person is also notified if the email wasn't delivered or opened because it was hard bounce, for example, if the email address was invalid. The sales person is also notified if the email wasn't delivered or opened due to a soft bounce, for example, the email was identified as JUNK or SPAM and didn't reach the inbox of the contact.

For more examples of business scenarios where you can use business events from SAP Marketing Cloud, see the following blog: [Extended Capabilities of SAP Marketing Cloud using Business Event Handling](#). 📄

### i Note

The SAP blog isn't part of the official documentation of SAP Marketing Cloud and some of the information may be outdated.

## More Information

[SAP Event Mesh](#)

[Business Event Handling](#)

[SAP Marketing Cloud Business Events](#) 📄

## 6.1 Campaign File Export

Business event for campaign file export (`MarketingObjectAttachment`).

During campaign execution, you can export target group members with the fields of the export definition as a file. The file is stored in relation to the campaign in SAP Marketing Cloud.

If another system wants to use this file to access the contact information and send out white papers, for example, then it alerts SAP Marketing Cloud by a business event. The event is raised after a successful export. The external system can retrieve the contact information using the `Read Content of Export Files in Campaigns` API (`API_MKT_EXPORT_DEFINITION`). For more information, see [Read Content of Export Files in Campaigns \[page 901\]](#).

## Business Events

The following business event is available for the `MarketingObjectAttachment` object:

## Events

Event	Description	Payload
Created	This event is raised when a marketing campaign export file is created.	<code>MarketingObjectUUID</code> : The unique identifier of the marketing object. <code>MarketingObject</code> : <code>MarketingObject</code> <code>MarketingObjectType</code> : The type of the marketing object. <code>MktgObjectAttachmentFilename</code> : The filename of the marketing object attachment.

## Additional Information

For more information about how business events are handled, see [Business Event Handling](#).

For more technical information about this event, see the [Marketing Campaign Export File Events](#) page on the SAP API Business hub. One-time registration is required for first-time users.

## 6.2 Campaigns

Business events for campaigns (`MarketingCampaign`).

A business event can be raised for externally executed campaigns when the campaign has ended and the status has been changed to Stopped. The event allows the external system on which the campaign was executed, to react and retrieve further information using the `Campaigns API (API_MKT_CAMPAIGN)`. For more information, see [Campaigns \[page 767\]](#).

## Business Events

The following business event is available for the `MarketingCampaign` object:

### Events

Event	Description	Payload
Completed	This event is raised when a marketing campaign is completed.	<code>CampaignUUID</code> : The unique identifier of the campaign.

## Additional Information

For more information about how business events are handled, see [Business Event Handling](#).

For more technical information about this event, see the [Marketing Campaign Events](#) page on the SAP API Business hub. One-time registration is required for first-time users.

## 6.3 Coupon Code Usages

Business event for coupon code usage (`CouponCodeUsages`).

A business event can be raised to collect coupon code usage changes for coupons. Follow-up activities can then be triggered. You can retrieve detailed information about the coupon code usage using the `Coupons` API (`API_MKT_COUPON_SRV`). For more information, see [Coupons \[page 1026\]](#).

## Business Events

The following business event is available for the `CouponCodeUsages` business object:

Events

Event	Description	Payload
Changed	This event is raised when a marketing coupon is used during campaign execution.	<code>CouponCodeUUID</code> : The unique identifier of a coupon code.

## Example for a Received Message

Sample for Enterprising Messaging Java API (emjapi):

### Sample Code

```
[
  {
    "message": "    {\"eventType\": \"BO.MarketingCoupon.Changed\",
    \"cloudEventsVersion\": \"0.1\", \"source\": \"https://S4HANAOD.sap.com\",
    \"eventID\": \"+hY+UEf5HtmYoI4D1/x9uw==\", \"eventTime\": \"2019-04-17T11:32:10Z
    \", \"schemaURL\": \"https://S4HANAOD.sap.com/sap/opu/odata/IWXBE/
    BROWSER_SRV/\", \"contentType\": \"application/json\", \"data\": {\"KEY\":
    [{\"COUPONUUID\": \"FA163E5047F91ED9909BF4279961A4C7\"}], \"COUPONCODEUUID\":
    [{\"COUPONCODEUUID\": \"364CD2B40C8B189316006024A4657417\"}]}}\",
    \"timestamp\": \"4/17/19 11:32 AM\",
    \"id\": \"47a2003c-95d2-475d-bd59-cfc81a053f2d\"
  }
]
```

For further processes, such as reading further coupon information requested by the OData services (such as `API_MKT_COUPON`, `CUAN_COUPON_MAINTAIN`, and so on), `CouponUUID` or `CouponCodeUUID` must be converted and entered as `CouponUUID equals fa163e50-47f9-1ed9-909b-f4279961a4c7` and `CouponCodeUUID equals 364C-d2b4-0c8b-18931600-6024a4657417`.

### Examples

Coupon information with GUID: `/sap/opu/odata/SAP/API_MKT_COUPON_SRV/Coupons(guid'fa163e50-47f9-1ed9-909b-f4279961a4c7')`

Coupon code information with GUID: `/sap/opu/odata/SAP/API_MKT_COUPON_SRV/CouponCodes(guid'364C-d2b4-0c8b-18931600-6024a4657417')`

## Additional Information

For more information about how business events are handled, see [Business Event Handling](#).

For more technical information about this event, see the [Marketing Coupon Events](#) page on the SAP API Business hub. One-time registration is required for first-time users.

## 6.4 Interactions

Business events for interactions (`Interaction`).

A business event can be raised for any interaction type. Follow-up activities can then be triggered. You can retrieve detailed information using the `Interactions` API (`API_MKT_INTERACTION`). For more information, see [Interactions \[page 615\]](#).

## Business Events

The following business event is available for the `Interaction` object:

Events

Event	Description	Payload
Created	This event is raised when an interaction is created.	<code>InteractionUUID</code> : The unique identifier of the Interaction.  <code>InteractionType</code> : The type of the Interaction.

## Marking an Interaction Type for Business Events

To indicate that you want to raise a business event for an interaction type, do the following:

1. Open the *Manage Your Solution* app.
2. Go to the *Managing Interaction Content* configuration step under *Contacts and Profiles*.
3. Find the interaction type you want to raise a business event for.
4. Select the *Business Event* checkbox in the *Key Information* screen.

The following screen capture shows an example of an interaction type that has been identified as a business event.

The screenshot shows the SAP configuration interface for an interaction type. The title bar reads 'Edit Interaction Type: Click Through'. Below the title bar, there are tabs for 'Key Information', 'Interaction Reasons', and 'Assignments'. The 'Key Information' tab is active. It contains the following fields and options:

- Name: \* Click Through
- ID: CLICK\_THROUGH
- Direction: Inbound (dropdown menu)
- Empty ID:
- Technical:
- Text Analysis:
- Business Event:  (highlighted with a red box)

## Additional Information

For more information about how business events are handled, see [Business Event Handling](#).

For more technical information about this event, see the [Interaction Events](#) page on the SAP API Business hub. One-time registration is required for first-time users.

## 6.5 Interaction Contacts

Business events for interaction contacts (*InteractionContact*).

If you implement custom or partner scenarios that are dependent on the lifecycle of an interaction contact, you can use business events for interaction contacts. These events are used to trigger a reaction when an interaction contact is created, updated, merged, or deleted. You can then react on these changes. You can retrieve detailed information about the interaction contacts using the following APIs:

- *Contacts* API (*API\_MKT\_CONTACTS*). For more information, see [Contacts \[page 412\]](#).
- *Interaction Contacts* API (*API\_MKT\_INTERACTION\_CONTACT*). For more information, see [Interaction Contacts \[page 469\]](#).
- *Corporate Accounts* API (*API\_MKT\_CORPORATE\_ACCOUNT*). For more information, see [Corporate Accounts \[page 512\]](#).

## Business Events

The following business events are available for the `InteractionContact` object:

Events

Event	Description	Payload
Created	This event is raised when an interaction contact is created.	<code>InteractionContactUUID</code> : The unique identifier of the Interaction Contact.
Changed	This event is raised when an interaction contact is changed.	<code>InteractionContactType</code> : The type of the Interaction Contact.
Deleted	This event is raised when an interaction contact is deleted.	
Merged	This event is raised when an interaction contact is merged.	<code>InteractionContactUUID</code> : The unique identifier of the Interaction Contact.  <code>AssgdToInteractionContactUUID</code> : The unique identifier of the interaction contact, where the merged contact is assigned to.  <code>AssgdToInteractionContactType</code> : The type of the Interaction Contact, where the merged contact is assigned to.

## Additional Information

For more information about how business events are handled, see [Business Event Handling](#).

For more technical information about these events, see the [Interaction Contact Events](#) page on the SAP API Business Hub. One-time registration is required for first-time users.

## 6.6 Marketing Permissions

Business events for marketing permissions (`MarketingPermission`).

Marketing permissions that are gathered in SAP Marketing Cloud are often distributed to other marketing-related systems, for example, call center solutions. When a permission is created or changed in SAP Marketing Cloud, a business event can be raised to trigger the distribution of that permission. You can retrieve detailed information about the permission using the `Interaction Contacts API` (`API_MKT_INTERACTION_CONTACT`). For more information, see [Interaction Contacts \[page 469\]](#).

## Business Events

The following business events are available for the `MarketingPermission` object:

Events

Event	Description	Payload
Created	This event is raised when a marketing permission is created.	<code>MarketingPermissionUUID</code> : The unique identifier of a marketing permission.
Changed	This event is raised when a marketing permission is changed.	<code>ContactPermission</code> : Contact permission (contact allowed / not allowed or permission to be checked).

## Additional Information

For more information about how business events are handled, see [Business Event Handling](#).

For more technical information about these events, see the [Marketing Permission Events](#) page on the SAP API Business hub. One-time registration is required for first-time users.

## 6.7 Marketing Subscriptions

Business events for marketing subscriptions (`MarketingSubscription`).

Marketing subscriptions that are gathered in SAP Marketing Cloud are often distributed to other marketing-related systems, for example, call center solutions. When a contact subscribes or changes their subscription in SAP Marketing Cloud, a business event can be raised to trigger the distribution of that subscription. You can retrieve detailed information about the subscription using the `Interaction Contacts` API (`API_MKT_INTERACTION_CONTACT`). For more information, see [Interaction Contacts \[page 469\]](#).

## Business Events

The following business events are available for the `MarketingSubscription` object:

## Events

Event	Description	Payload
Created	This event is raised when a marketing subscription is created.	<code>MarketingSubscriptionUUID</code> : The unique identifier of a marketing subscription.
Changed	This event is raised when a marketing subscription is changed.	<code>ContactSubscription</code> : Contact subscription (contact allowed / not allowed or permission to be checked).

## Additional Information

For more information about how business events are handled, see [Business Event Handling](#).

For more technical information about these events, see the [https://api.sap.com/event/SAPMarketingCloudBusinessEvents\\_MarketingSubscriptionEvents/overview](https://api.sap.com/event/SAPMarketingCloudBusinessEvents_MarketingSubscriptionEvents/overview) page on the SAP API Business hub. One-time registration is required for first-time users.



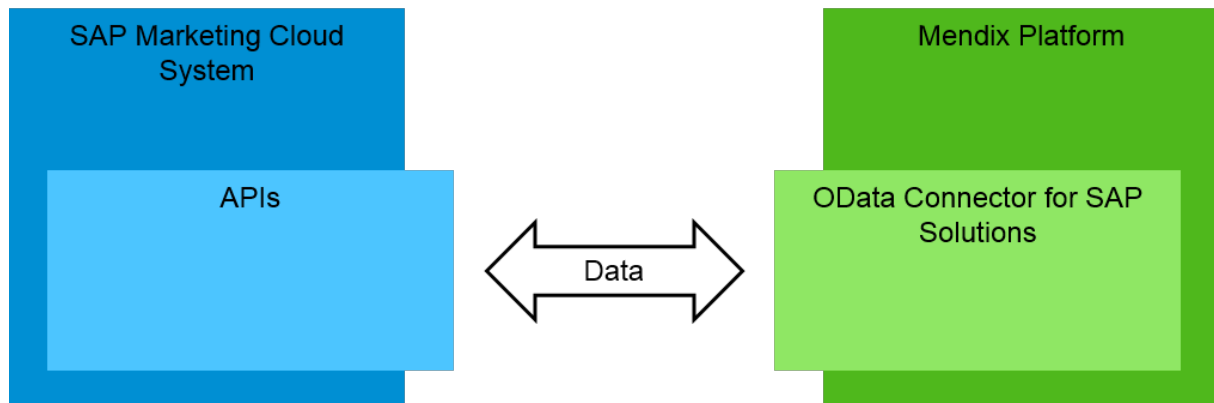
# 7 Integration Technologies

Here you can find an overview about the integration technologies used in your solution.

Technology	Description
SOAP	SOAP is a protocol specification for exchanging structured information in the implementation of web services in computer networks. The message format is based on XML. Message transfer is based on other web protocols, usually HTTP(S).
REST	Representational State Transfer (REST) is a architecture style for creating scalable web services. REST services are usually based on HTTP(S). They use HTTP URIs for resource identification and HTTP methods for service operations. It is used widely as an alternative to SOAP, as REST services usually provide better performance, scalability and simpler interfaces.
OData	OData provides a protocol for queryable and interoperable RESTful APIs. It provides an entity-based data model and a query language. Create, read, update and delete methods expressed using HTTP methods. All OData services use HTTPS protocol to ensure data security. The standard port for HTTPS is 443.
CSV	A comma-separated values (CSV) (also sometimes called character-separated values) file stores tabular data (numbers and text) in plain-text form. CSV files are widely used as import or export format and can be down- and uploaded to many systems.

## 8 Create Your Own Apps: SAP Rapid Application Development by Mendix

With this integration, you can supplement marketing capabilities with apps that you've created on the low code platform Mendix.



### Business Benefits

- Extend SAP Marketing Cloud with customer-specific applications for dedicated business use cases, for example, a trade fair app for registration of contacts for demo sessions.
- Enable business users to easily create apps for marketing using SAP Rapid Application Development by Mendix.

### Key Capabilities

- Extend SAP Marketing Cloud with a side-by-side approach, public APIs, and extensions.
- Provide user authentication and ensure security standards.
- Provide a means of extending apps to support a two-tier landscape for quality and productive systems.
- Fulfill requirements when the connected SAP Marketing Cloud system is upgraded.

### Example Scenarios

- In B2B scenarios, apps can enable sales people to enter leads and maintain contact data in marketing.
- Other companies' access to customer contacts for registration at session events or to request a product demo.
- For B2C customers, it's beneficial for their call-center agents to put together a well-tailored factsheet of the most important profile data of their consumers by means of a low code app.
- You're already running SAP Marketing Cloud and want to take the next step in your digitalization by offering individual product demo sessions at a trade fair. The registration can be realized by sending out emails with registration codes, for example, using offer coupon codes.

## Prerequisites

- You've created a user on Mendix to access their platform.
- We recommend to install the Mendix Studio Pro (development platform) **locally**.

## Steps to Get Your App

### Prepare

1. Make sure that you have the [Administrator](#) role assigned.
2. Define the business scenarios you want to realize with the app and figure out, which application programming interfaces (APIs) from SAP Marketing Cloud you need.  
Get familiar with the data model, the entity relationship model and the available nodes and components, and how they interact with each other on the side of SAP Marketing Cloud.
3. Map the required APIs to your business scenario and check if they're available in SAP Marketing Cloud.

### Steps in SAP Marketing Cloud

1. Create communication arrangements and communication systems for the required communication scenarios in the SAP system for the APIs you selected.
2. Open the metadata request URL in your browser and save the text file to your machine.  
The URL has the pattern `https://<host>/sap/opu/odata/sap/API_MKT_<object>_SRV;v=<latest version>/$metadata`.



#### Example

```
https://<host>/sap/opu/odata/sap/API_MKT_CONTACT_SRV;v=0003/$metadata
```

### Steps in Mendix

1. Use the OData Model Creator for SAP Solutions to connect the SAP system with the Mendix solution.  
You can find the connectors in the Mendix App Store under [Connectors > SAP](#).  
In our example, we used the [OData Connector for SAP Solutions](#) and the [OData Model Creator for SAP Solutions](#).
2. There you can manually upload the data model:
  1. Choose [Manual](#) and upload the locally stored metadata XML file.
  2. Select the locally stored file.
  3. Choose [Continue](#) twice, and then [Generate .mpk](#). The Mendix system generates a data model out of the SAP file.  
This generated data model can be imported into your Mendix project. You can see the data models next to your module folder.
3. Connect the required elements of data models in a combined [Domain Model](#) of your module.
4. We recommend using the offered app templates by Mendix for SAP, such as the [SAP App Template for Fiori Apps](#). With the template, the correct OData connector is also installed.
5. For deployment ensure that the app user has the business catalog roles assigned that are required for the APIs you are using. You can generally find this information in the API documentation.  
Alternatively, you can use a technical user with basic authentications.

## Related Information

- [Business Scenarios](#)
- [Implementing Integrations for Business Scenarios \[page 9\]](#)
- [Consuming the Integration APIs \[page 395\] > Getting Started \[page 387\] > Consuming the Integration APIs \[page 395\]](#)
- Documentation provided by SAP: [SAP Rapid Application Development by Mendix](#)
- Tutorial from SAP's Developer Community: [Get Started with SAP Rapid Application Development by Mendix](#) 
- Documentation provided by Mendix: [Low code development for SAP® by Mendix](#) 

## Component for Incidents



Please use the following component for incidents: XX-PART-MDX-RAD

# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2021 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.