



---

# Salesforce Knowledge Developer Guide

Version 39.0, Spring '17





# CONTENTS

<b>Chapter 1: Develop with Salesforce Knowledge</b> .....	<b>1</b>
Set Up a Knowledge Development Organization .....	<b>3</b>
Enable Salesforce Knowledge .....	<b>3</b>
Edit the Default Article Type .....	<b>4</b>
Create an Article .....	<b>6</b>
Publish an Article .....	<b>6</b>
<b>Chapter 2: Salesforce Knowledge SOAP API</b> .....	<b>8</b>
Knowledge Object Diagram .....	<b>9</b>
Data Categories and Knowledge with SOAP API Objects .....	<b>10</b>
Knowledge SOAP API Objects .....	<b>11</b>
Article_Type__DataCategorySelection .....	<b>12</b>
Article_Type__Feed .....	<b>14</b>
KnowledgeArticle .....	<b>20</b>
KnowledgeArticleVersion .....	<b>22</b>
KnowledgeArticleVersionHistory .....	<b>29</b>
KnowledgeArticleViewStat .....	<b>31</b>
KnowledgeArticleVoteStat .....	<b>33</b>
CaseArticle .....	<b>34</b>
LinkedArticle .....	<b>36</b>
LinkedArticleFeed .....	<b>38</b>
LinkedArticleHistory .....	<b>41</b>
NewsFeed .....	<b>42</b>
RecentlyViewed .....	<b>49</b>
ReportFeed .....	<b>52</b>
SearchPromotionRule .....	<b>60</b>
TopicAssignment .....	<b>60</b>
UserProfileFeed .....	<b>62</b>
Knowledge SOAP API Calls .....	<b>69</b>
describeKnowledge()Call Name .....	<b>69</b>
describeDataCategoryGroups() .....	<b>70</b>
describeDataCategoryGroupStructures() .....	<b>73</b>
search() .....	<b>78</b>
<b>Chapter 3: Salesforce Knowledge REST APIs</b> .....	<b>82</b>
Manage Knowledge with REST APIs .....	<b>83</b>
Archive the Master Version of an Article .....	<b>84</b>
Assign a Task Related to a Master Article .....	<b>85</b>
Assign a Task Related to a Translation .....	<b>86</b>

## Contents

Delete a Master Version of an Article	87
Delete a Translated Version of an Article	87
Edit an Online Version of a Master Article	88
Publish the Master Version of an Article	88
Restore an Archived Version of an Article	89
Retrieve Article Metadata	90
Retrieve Knowledge Language Settings	90
Retrieve a Version of an Article	92
Search for Metadata Elements of a Master Version	92
Search for Metadata Elements of a Translated Version	92
Set a Translated Article Version to Complete	93
Set a Translated Article Version to Incomplete	94
Submit an Article for Translation	94
Unpublish the Master Version of an Article	95
Unpublish the Online Version of a Translated Article	96
Parameterized Search	96
Search	104
Search Scope and Order	105
Search Result Layouts	105
Search Suggested Records	106
Search Suggested Article Title Matches	111
Search Suggested Queries	113
Support Knowledge with REST API	115
Data Category Groups	116
Data Category Detail	118
Articles List	120
Articles Details	124
<b>Chapter 4: Salesforce Knowledge Metadata API</b>	<b>129</b>
ArticleType	130
ArticleType Layout	133
ChannelLayout	135
ArticleType CustomField	136
KnowledgeSettings	139
DataCategoryGroup	144
SearchSettings	150
SearchLayouts	153
SynonymDictionary	155
ExternalDataSource	157
<b>Chapter 5: Salesforce Knowledge with SOQL and SOSL</b>	<b>162</b>
Update an Article's Keyword Tracking with SOQL	163
Update an Article Viewstat with SOQL	163
WITH filteringExpression	163

## Contents

WITH DATA CATEGORY filteringExpression .....	164
Using Relationship Queries with Data Category Selection Objects .....	168
About SOSL .....	168
Update an Article's Keyword Tracking with SOSL .....	169
Update an Article's Viewstat with SOSL .....	170
WITH DATA CATEGORY DataCategorySpec .....	170
WITH SNIPPET .....	172
<b>Chapter 6: Salesforce Knowledge, Apex, and Visualforce</b> .....	<b>176</b>
PublishingService Class .....	177
PublishingService Methods .....	177
<b>Index</b> .....	<b>189</b>



# CHAPTER 1 Develop with Salesforce Knowledge

## In this chapter ...

- [Set Up a Knowledge Development Organization](#)

Salesforce Knowledge gives your website visitors, clients, partners, and service agents the ultimate in support. You can create and manage your company information and securely share it when and where it is needed. Most features are declarative and let you point-and-click your way to a successful implementation. However, with the Force.com platform, you can also code your way to case deflection, happier customers, and more productive support agents.

*Articles* capture information about your company's products and services that you want to make available in your knowledge base. Articles in the knowledge base can be classified by using one or more data categories to make it easy for users to find the articles they need. Administrators can use data categories to control access to articles.

Articles are based on article types, which rely on:

- Article-type layouts to organize the content in sections.
- Article-type templates to render articles.

Every article is managed in a publishing cycle.

### Article Type

All articles in Salesforce Knowledge are assigned to an *article type*. An article's type determines the type of content it contains, its appearance, and which users can access it. For example, a simple FAQ article type might have two custom fields, `Question` and `Answer`, where article managers enter data when creating or updating FAQ articles. A more complex article type may require dozens of fields organized into several sections. Using layouts and templates, administrators can structure the article type in the most effective way for its particular content. User access to article types is controlled by permissions. For each article type, an administrator can grant "Create," "Read," "Edit," or "Delete" permissions to users. For example, the article manager might want to allow internal users to read, create, and edit FAQ article types, but let partner users only read FAQs.

### Article-Type Layout

An *article-type layout* enables administrators to create sections that organize the fields on an article, as well as choose which fields users can view and edit. One layout is available per article type. Administrators modify the layout from the article-type detail page.

### Article-Type Template

An *article-type template* specifies how the sections in the article-type layout are rendered. An article type can have a different template for each of its four channels. For example, if the Customer Portal channel on the FAQ article-type is assigned to the Tab template, the sections in the FAQ's layout appear as tabs when customers view an FAQ article. For the Table of Contents template, the sections defined in the layout appear on a single page (with hyperlinks) when the article is viewed. Salesforce provides two standard article-type templates, Tab and Table of Contents. Custom templates can be created with Visualforce.

### Channel

A channel refers to the medium by which an article is available. Salesforce Knowledge offers four channels where you can make articles available.

- Internal App: Salesforce users can access articles in the Articles tab depending on their role visibility.

- **Customer:** Customers can access articles if the Articles tab is available in a community or Customer Portal. Customer users inherit the role visibility of the manager on the account. In a community, the article is only available to users with Customer Community or Customer Community Plus licenses.
- **Partner:** Partners can access articles if the Articles tab is available in a community or partner portal. Partner users inherit the role visibility of the manager on the account. In a community, the article is only available to users with Partner Community licenses.
- **Public Knowledge Base:** Articles can be made available to anonymous users by creating a public knowledge base using the *Sample Public Knowledge Base for Salesforce Knowledge* app from the AppExchange. Creating a public knowledge base requires Sites and Visualforce.

### **Publishing Cycle**

Salesforce Knowledge Articles move through a publishing cycle from their creation to their deletion. The publishing cycle includes three different statuses: `Draft` is the stage when a new article is being created or an existing one is being updated. Articles with the `Online` status are draft articles that have been published and are now available to their different channels. Eventually, when a published article is at the end of its life, it can be moved to the `Archived` status or sent back to `Draft` to be updated in a subsequent version.



# Set Up a Knowledge Development Organization

To develop with Salesforce Knowledge, first enable Knowledge in your development organization.

To build a Knowledge base, you can follow the steps in this guide or get a badge while you're at it with the [Knowledge Basics Trailhead module](#).

## IN THIS SECTION:

### [Enable Salesforce Knowledge](#)

Get Knowledge turned on in your development organization.

### [Edit the Default Article Type](#)

Article types, such as FAQs and procedures, provide a format, or structure, for agents to compose quality articles. An article-type is created for you to get started, but it has no fields for content. Add a custom field to the default article type.

### [Create an Article](#)

Create articles from the Knowledge tab to build your knowledge base.

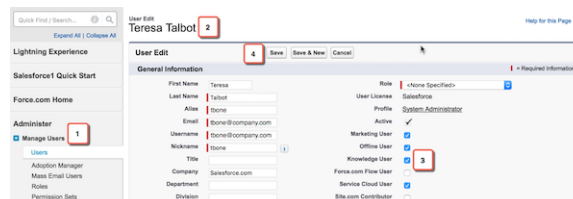
### [Publish an Article](#)

Create articles from the Knowledge tab to build your knowledge base.

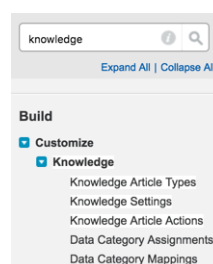
## Enable Salesforce Knowledge

Get Knowledge turned on in your development organization.

1. From Setup, click **Manage Users > Users**.
2. Click **Edit** next to your name.
3. Select the **Knowledge User** checkbox.

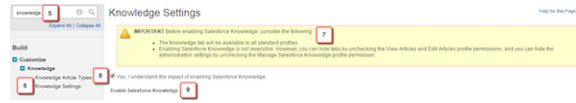


4. Click **Save**.  
Assigning a Knowledge User license makes the Knowledge pages show up in Setup.

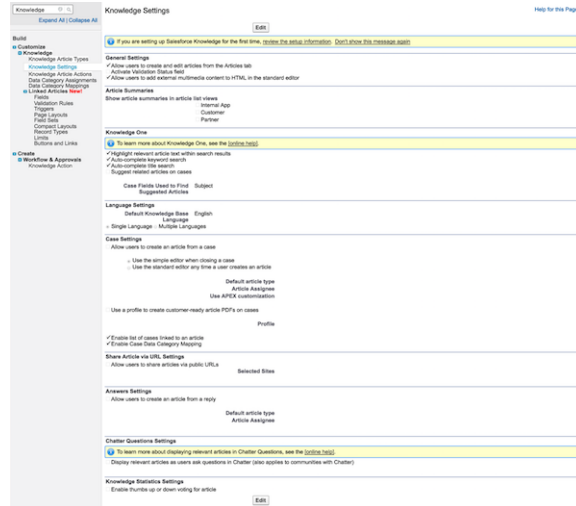


5. From Setup, type *Knowledge* into the **Quick Find** box.

6. Click **Knowledge Settings**.



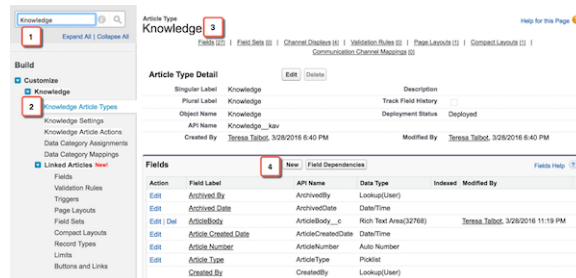
7. Read the information on enabling Salesforce Knowledge in your org.
8. Check **Yes, I understand the impact of enabling Salesforce Knowledge**.
9. Click **Enable Salesforce Knowledge**.



## Edit the Default Article Type

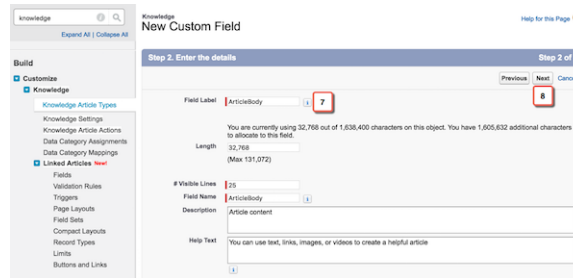
Article types, such as FAQs and procedures, provide a format, or structure, for agents to compose quality articles. An article-type is created for you to get started, but it has no fields for content. Add a custom field to the default article type.

1. From Setup, type *Knowledge* into the **Quick Find** box.
2. Click **Knowledge Article Types**.
3. Click **Knowledge** in the **Label** column.
4. Click **New** in the **Fields** section.

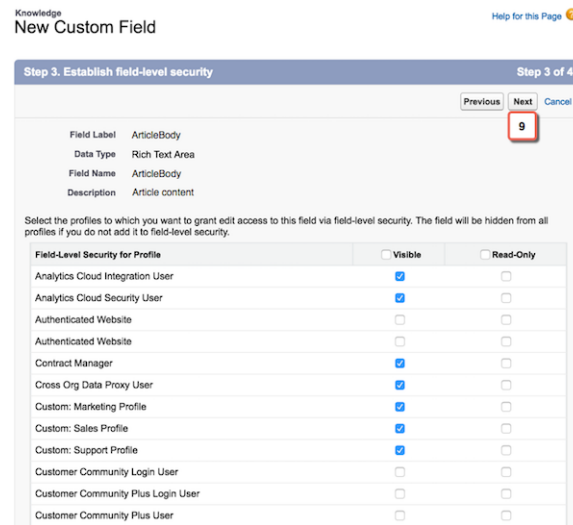


5. Select **Text Area (Rich)**.
6. Click **Next**.

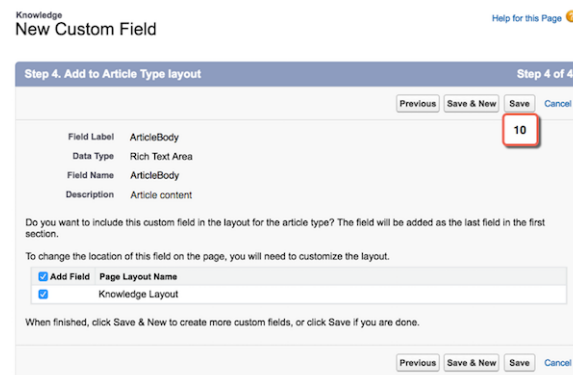
7. Name your new field *ArticleBody*.
8. Click **Next**.



9. Here's where you can hide this field from user profiles to add more security. Since your profile is System Administrator, the field is visible to you by default. For now, let's leave the default settings and click **Next**.



10. You must connect the article-type to a page layout. By default, the field is added as the last field in the first section, the information section. Let's again go with the default settings and click **Save**.




The new field, ArticleBody, is added alphabetically to the article type detail page in the Fields section.

# Create an Article

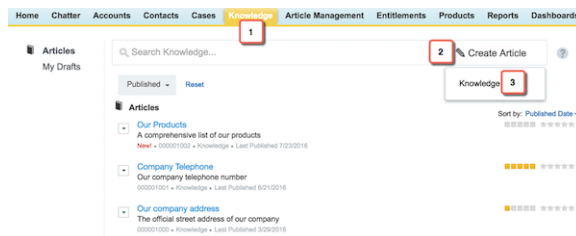
Create articles from the Knowledge tab to build your knowledge base.

The more articles you have the more realistic your development knowledge base. You can import multiple articles with .csv files; one for each article type. For information on importing articles, see [https://help.salesforce.com/HTViewHelpDoc?id=knowledge\\_article\\_importer.htm](https://help.salesforce.com/HTViewHelpDoc?id=knowledge_article_importer.htm).

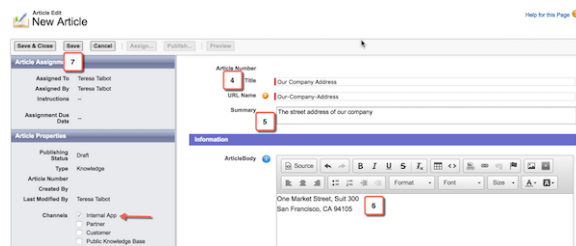
1. Click the **Knowledge** tab.

 **Note:** You may need to click the plus sign at the end of your tabs and find the Knowledge tab in the list of all tabs available. You can add the Knowledge and Article Management tabs to an app.

2. Click **Create Article**.
3. Select the **Knowledge** article type.



4. Create a title for your article, such as “Our company address.”
5. Write a description, this goes under the title in the search results.
6. Put your company address in the rich text area field you created in the last unit. Notice that the article is available by default to your internal users.
7. Click **Save**.



# Publish an Article

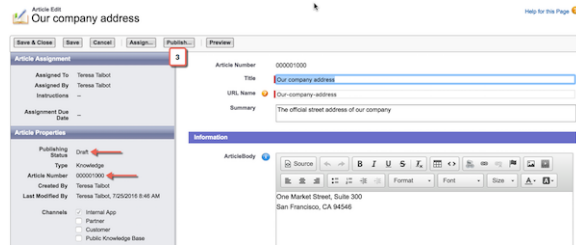
Create articles from the Knowledge tab to build your knowledge base.

You must publish the articles after creation or import. Non-published articles do not display on your public channels: customer and partner communities or public websites.

1. If you’ve navigated away from the article you created, you can find it in either tab, Knowledge or Article Management, under My Drafts.



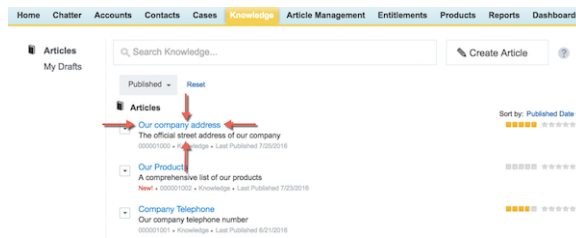
2. Click the **Our company address** draft. Notice that the article now has an article number and a publishing status.
3. Click **Publish**.



4. Determine if you want to publish now or at a scheduled time in the future. Let's stick with the default once again and publish now.
5. Click **OK**.



Salesforce Knowledge takes you back to the Knowledge tab and your article is at the top of the list!



# CHAPTER 2 Salesforce Knowledge SOAP API

## In this chapter ...

- Knowledge Object Diagram
- Data Categories and Knowledge with SOAP API Objects
- Knowledge SOAP API Objects
- Knowledge SOAP API Calls

Articles are available through the KnowledgeArticleVersion and KnowledgeArticle objects in SOAP API. They both represent an article but provide different capabilities.

### KnowledgeArticleVersion

Every new draft article in Salesforce Knowledge has a version number. When an article is published and you want to update it, you can create a new `Draft` with a distinct version number. Each version has its own ID. Once the updated version is ready to be published, it replaces the former one and updates the version number. You can access the content of an article version using the KnowledgeArticleVersion object and filter on its Draft or Online status. For example, the following query returns the title of the Draft version of all the articles across all article types in United States English:

```
SELECT Title
FROM KnowledgeArticleVersion
WHERE PublishStatus='Draft'
AND language = 'en_US'
```

You can change the language to return any other language your knowledge base supports, see “Support a Multilingual Knowledge Base” in the Salesforce online help.

Articles are also auto-assigned an Article Number, which is not a unique identifier to an individual article, but an identifier to a master article and all of its available translations.



**Note:** Both the master version (the Knowledge article with `IsMasterLanguage = 1`) and the translations are KnowledgeArticleVersion objects.

### KnowledgeArticle

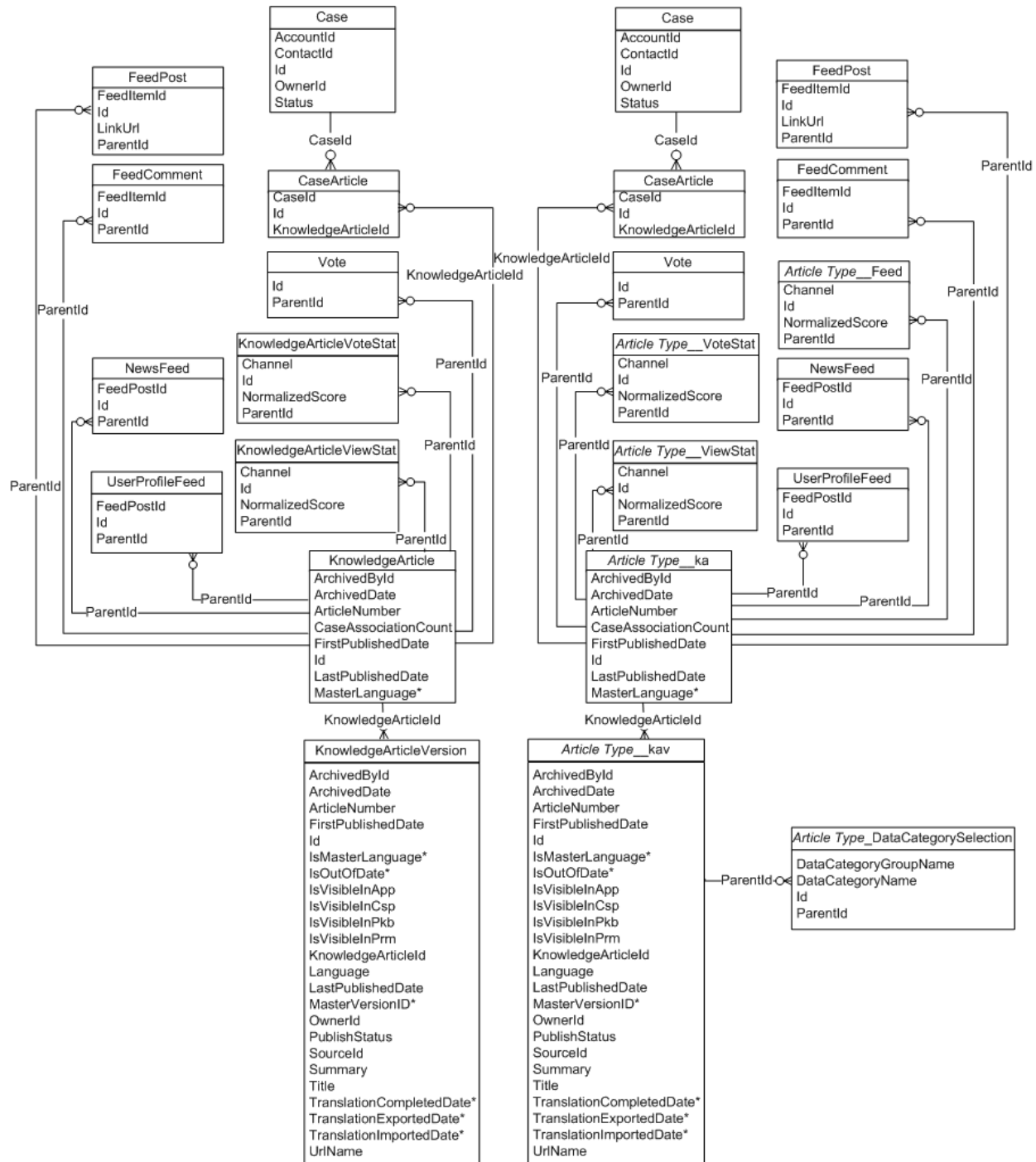
Unlike KnowledgeArticleVersion, the ID of a KnowledgeArticle record is identical irrespective of the article's version (status). Where the KnowledgeArticleVersion object provides API access to an article's custom field values, the KnowledgeArticle object provides API access to an article's metadata fields.

The article record is the parent container of all versions of an article, whatever the publishing status (draft, published, archived) and the language. While KnowledgeArticle and KnowledgeArticleVersion represent any article in the Knowledge Base, `<Article Type>__ka` and `<Article Type>__kav` are the concrete representation of respectively KnowledgeArticle (“\_\_ka” suffix) and KnowledgeArticleVersion (“\_\_kav” suffix) for a specific article type. For example, `Offer__kav` gives access to every Offer articles. KnowledgeArticle and `<Article Type>__ka` give access to an article independent of its version. KnowledgeArticleVersion and `<Article Type>__kav` enables you to retrieve a specific article version and all its standard (KnowledgeArticleVersion) and custom (`<Article Type>__kav`) fields. The following query returns the title for all the published offers in United States English:

```
SELECT Title
FROM Offer__kav
WHERE PublishStatus='online'
AND language = 'en_US'
```

# Knowledge Object Diagram

This entity relationship diagram (ERD) illustrates relationships between the Salesforce Knowledge objects.



## Data Categories and Knowledge with SOAP API Objects

---

Salesforce Knowledge uses data categories to support successful article search.

Data categories are organized by category group and let:

- Users classify and find records.
- Administrators control access to records.

Data categories can be used by Salesforce Knowledge (articles) and answers communities (questions).

### Data Categories and Articles

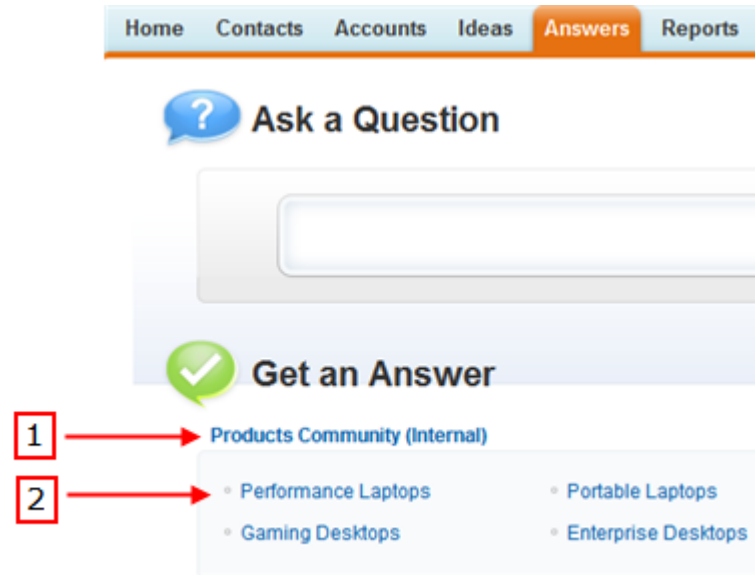
Salesforce Knowledge uses data categories to classify articles and make them easier to find. For example, to classify articles by sales regions and products, create two category groups: Sales Regions and Products. The Sales Regions category group could consist of a geographical hierarchy, such as All Sales Regions as the top level and North America, Europe, and Asia at the second level. The Products group could have All Products as the top level and Phones, Computers, and Printers at the second.

Data Category Limits	Details	
Maximum number of data category groups and active data category groups	5 category groups, with 3 groups active at a time	5 category groups, with 3 groups active at a time
Maximum number of categories per data category group	100 categories in a data category group	100 categories in a data category group
Maximum number of levels in data category group hierarchy	5 levels in a data category group hierarchy	5 levels in a data category group hierarchy
Maximum number of data categories from a data category group assigned to an article	8 data categories from a data category group assigned to an article	8 data categories from a data category group assigned to an article

### Data Categories and Answers

In an answers zone, data categories help organize questions for easy browsing. Each answers zone supports one category group. For example, if you're a computer manufacturer you might create a Products category group that has four sibling categories: Performance Laptops, Portable Laptops, Gaming Desktops, and Enterprise Desktops. On the Answers tab, zone members can assign one of the four categories to each question and then browse these categories for answers to specific questions.





## Knowledge SOAP API Objects

---

SOAP API object resources for working with Salesforce Knowledge.

### IN THIS SECTION:

#### [Article Type\\_\\_DataCategorySelection](#)

A data category selection represents a data category that classifies an article. This object is available in API version 19.0 and later.

#### [Article Type\\_\\_Feed](#)

Represents a single feed item in the feed displayed on the detail page for an article. This object is available in API version 20.0 and later.

#### [KnowledgeArticle](#)

Provides read-only access to an article and the ability to delete the master article. This object is available in API version 19.0 and later.

#### [KnowledgeArticleVersion](#)

Provides a global view of standard article fields across all article types depending on their version. This object is available in API version 18.0 and later.

#### [KnowledgeArticleVersionHistory](#)

Enables read-only access to the full history of an article. This object is available in API version 25.0 and later.

#### [KnowledgeArticleViewStat](#)

Provides statistics on the number of views for the specified article across all article types. This object is read-only and available in API version 20 and later.

#### [KnowledgeArticleVoteStat](#)

Provides the weighted rating for the specified article on a scale of 1 to 5 across all article types. This object is read-only and available in API version 20 and later.

#### [CaseArticle](#)

Represents the association between a Case and a KnowledgeArticle. This object is available in API version 20.0 and later.

[LinkedArticle](#)

Represents a Knowledge article that is attached to a record. This object is available in API version 37.0 and later.

[LinkedArticleFeed](#)

Represents a single feed item on a linked Knowledge article attached to a record. This object is available in API version 37.0 and later.

[LinkedArticleHistory](#)

Represents the history of changes made to tracked fields on a linked article. This object is available in API version 37.0 and later.

[NewsFeed](#)

Represents a single feed item on a user's home tab. A Chatter feed shows recent changes to records that the user is following.

[RecentlyViewed](#)

Represents records that the current user has recently viewed or referenced (by viewing a related record).

[ReportFeed](#)

Represents a single feed item in the feed displayed on a report. This object is available in API version 20.0 and later.

[SearchPromotionRule](#)

Represents a promoted search term, which is one or more keywords that you associate with a Salesforce Knowledge article. When a user's search query includes these keywords, the associated article is returned first in search results. This object is available in API version 31.0 and later.

[TopicAssignment](#)

Represents the assignment of a topic to a specific feed item, record, or file. This object is available in API version 28.0 and later.

[UserProfileFeed](#)

Represents a user profile feed, which tracks all actions by a user on records that can be tracked in a feed. This feed is displayed on the user profile page.

## *Article Type*\_\_DataCategorySelection

A data category selection represents a data category that classifies an article. This object is available in API version 19.0 and later.

This object can be used to associate an article with data categories from a data category group or to query the category selections for an article.

The object name is variable and has a syntax of *Article Type*\_\_DataCategorySelection, where *Article Type* is the Object Name for the article type associated with the article. For example, *Offer*\_\_DataCategorySelection represents the association between the *Offer* article type and its data categories. Every article is associated with an article type.

### Supported Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `getDeleted()`, `retrieve()`

### Special Access Rules


Users can only access, create or delete data category selection visible to their role, permission set, or profile. If a user has partial visibility on an article's categorization, only the visible categories are returned.

## Fields

Field Name	Details
DataCategoryGroupName	<p><b>Type</b> DataCategoryGroupReference</p> <p><b>Properties</b> Create</p> <p><b>Description</b> Unique name of the data category group which has categories associated with the article.</p>
DataCategoryName	<p><b>Type</b> DataCategoryGroupReference</p> <p><b>Properties</b> Create</p> <p><b>Description</b> Unique name of the data category associated with the article.</p>
ParentId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter</p> <p><b>Description</b> ID of the article associated with the data category selection.</p>

## Usage

Every article in Salesforce Knowledge can be categorized. A data category selection represents a category that has been selected to classify an article. You can use the *Article Type\_\_DataCategorySelection* object to query and manage article categorization in your organization. Client applications can create a categorization for an article with a Draft status. They can also delete and query article categorizations.

 **Note:** When using *Article Type\_\_DataCategorySelection* to classify an article, you can't select both a category (for example USA) and one of its descendants (California) or ascendant categories (North America). In this case, only the first category is selected.

Answer communities use *QuestionDataCategorySelection* to classify questions.

## SOQL Sample

The following SOQL query returns the data category selections used to classify the article whose ID is `ka0D000000005ApIAI`.

```
SELECT Id,DataCategoryName, ParentId
FROM Offer__DataCategorySelection WHERE ParentId='ka0D000000005ApIAI'
```

This clause only returns category unique names. To retrieve category labels use the following clause:

```
SELECT Id,toLabel(DataCategoryName), ParentId
FROM Offer__DataCategorySelection WHERE ParentId='ka0D000000005ApIAI'
```

 **Tip:** You can also use relationship queries to retrieve categorizations from an article type.

## Article Type\_\_Feed


Represents a single feed item in the feed displayed on the detail page for an article. This object is available in API version 20.0 and later.

An article feed shows recent changes to an article record for any fields that are tracked in feeds, and posts and comments about the article. It is a useful way to stay up-to-date with changes made to articles in Salesforce Knowledge. This object is available in API version 20.0 and later. The object name is variable and uses a *Article Type\_\_Feed* syntax, where *Article Type* is the Object Name for the article type associated with the article. For example, *Offer\_\_Feed* represents a feed item on an article based on the *Offer* article type.

## Supported Calls



`delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

## Fields

Field	Details
Body	<p><b>Type</b> textarea</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The content of <i>Article Type__Feed</i>. Required when <i>Type</i> is <i>TextPost</i>. Optional when <i>Type</i> is <i>ContentPost</i> or <i>LinkPost</i>. This field is the message that appears in the feed.</p>
CommentCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of <i>FeedComments</i> associated with this feed item.</p> <p> <b>Tip:</b> In a feed that supports pre-moderation, <i>CommentCount</i> isn't updated until a comment is published. For example, say you comment on a post that already has one published comment and your comment triggers moderation. Now there are two comments on the post, but the count says there's only one. In a moderated feed, your comment isn't counted until it's approved by an admin or a person with "CanApproveFeedPostAndComment" or "ModifyAllData" permission.</p> <p>This has implications for how you retrieve feed comments. In a moderated feed, rather than retrieving comments by looping through <i>CommentCount</i>, go through pagination until end of comments is returned.</p>


Field	Details
ContentData	<p><b>Type</b> base64</p> <p><b>Properties</b> Nillable</p> <p><b>Description</b> Required if <code>Type</code> is <code>ContentPost</code>. Encoded file data in any format, and can't be 0 bytes. Setting this field automatically sets <code>Type</code> to <code>ContentPost</code>.</p>
ContentDescription	<p><b>Type</b> textarea</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The description of the file specified in <code>ContentData</code>.</p>
ContentFileName	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> This field is required if <code>Type</code> is <code>ContentPost</code>. The name of the file uploaded to the feed. Setting <code>ContentFileName</code> automatically sets <code>Type</code> to <code>ContentPost</code>.</p>
ContentSize	<p><b>Type</b> int</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The size of the file (in bytes) uploaded to the feed. This field is read-only and is automatically determined during insert.</p>
ContentType	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The MIME type of the file uploaded to the feed. This field is read-only and is automatically determined during insert.</p>
CreatedDate	<p><b>Type</b> dateTime</p>

Field	Details
	<p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Date and time when this record was created. This field is a standard system field. Ordering by <code>CreatedDate DESC</code> sorts the feed by the most recent feed item.</p>
FeedPostId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> This field was removed in API version 22.0, and is available in earlier versions for backward compatibility only.  ID of the associated FeedPost. A FeedPost represents the following types of changes in an FeedItem: changes to tracked fields, text posts, link posts, and content posts.</p>
InsertedById	<p><b>Type</b> reference</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> ID of the user who added this object to the feed. For example, if an application migrates posts and comments from another application into a feed, the <code>InsertedBy</code> value is set to the ID of the logged-in user.</p>
IsDeleted	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Standard system field. Indicates whether the record has been moved to the Recycle Bin (<code>true</code>) or not (<code>false</code>).</p>
IsRichText	<p><b>Type</b> boolean</p> <p><b>Properties</b> Create, Defaulted on create, Filter, Group, Sort, Update</p> <p><b>Description</b> Indicates whether the feed item <code>Body</code> contains rich text. Set <code>IsRichText</code> to <code>true</code> if you post a rich text feed item via SOAP API. Otherwise, the post is rendered as plain text.  Rich text supports the following HTML tags:</p>

Field	Details
	<ul style="list-style-type: none"><li>• <code>&lt;p&gt;</code></li></ul> <p> <b>Tip:</b> Though the <code>&lt;br&gt;</code> tag isn't supported, you can use <code>&lt;p&gt;&amp;nbsp;&lt;/p&gt;</code> to create lines.</p> <ul style="list-style-type: none"><li>• <code>&lt;b&gt;</code></li><li>• <code>&lt;code&gt;</code></li><li>• <code>&lt;i&gt;</code></li><li>• <code>&lt;u&gt;</code></li><li>• <code>&lt;s&gt;</code></li><li>• <code>&lt;ul&gt;</code></li><li>• <code>&lt;ol&gt;</code></li><li>• <code>&lt;li&gt;</code></li><li>• <code>&lt;img&gt;</code></li></ul> <p>The <code>&lt;img&gt;</code> tag is accessible only via the API and must reference files in Salesforce similar to this example: <code>&lt;img src="sfdc://069B0000000omjh"&gt;&lt;/img&gt;</code></p> <p> <b>Note:</b> In API version 35.0 and later, the system replaces special characters in rich text with escaped HTML. In API version 34.0 and prior, all rich text appears as a plain-text representation.</p>
LastModifiedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Date and time when a user last modified this record. This field is a standard system field. When a feed item is created, <code>LastModifiedDate</code> is the same as <code>CreatedDate</code>. If a <code>FeedComment</code> is inserted on that feed item, then <code>LastModifiedDate</code> becomes the <code>CreatedDate</code> for that <code>FeedComment</code>. Deleting the <code>FeedComment</code> does not change the <code>LastModifiedDate</code>. Ordering by <code>LastModifiedDate DESC</code> sorts the feed by both the most recent feed item or comment.</p>
LikeCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of <code>FeedLikes</code> associated with this feed item.</p>
LinkUrl	<p><b>Type</b> url</p>

Field	Details
	<p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The URL of a <code>LinkPost</code>.</p>
ParentId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> ID of the article that is tracked in the feed. The feed is displayed on the detail page for this record.</p>
RelatedRecordId	<p><b>Type</b> reference</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> ID of the <code>ContentVersion</code> object associated with a <code>ContentPost</code>. This field is null for all posts except <code>ContentPost</code>.</p>
Title	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> The title of the feed item. When the <code>Type</code> is <code>LinkPost</code>, the <code>LinkUrl</code> is the URL and this field is the link name.</p>
Type	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> The type of feed item:</p> <ul style="list-style-type: none"> <li><code>ActivityEvent</code>—indirectly generated event when a user or the API adds a <code>Task</code> associated with a feed-enabled parent record (excluding email tasks on cases). Also occurs when a user or the API adds or updates a <code>Task</code> or <code>Event</code> associated with a case record (excluding email and call logging).</li> </ul> <p>For a recurring <code>Task</code> with <code>CaseFeed</code> disabled, one event is generated for the series only. For a recurring <code>Task</code> with <code>CaseFeed</code> enabled, events are generated for the series and each occurrence.</p>



Field	Details
	<ul style="list-style-type: none"> <li>• <code>AdvancedTextPost</code>—created when a user posts a group announcement.</li> <li>• <code>AnnouncementPost</code>—Not used.</li> <li>• <code>ApprovalPost</code>—generated when a user submits an approval.</li> <li>• <code>BasicTemplateFeedItem</code>—Not used.</li> <li>• <code>CanvasPost</code>—a post made by a canvas app posted on a feed.</li> <li>• <code>CollaborationGroupCreated</code>—generated when a user creates a public group.</li> <li>• <code>CollaborationGroupUnarchived</code>—Not used.</li> <li>• <code>ContentPost</code>—a post with an attached file.</li> <li>• <code>CreatedRecordEvent</code>—generated when a user creates a record from the publisher.</li> <li>• <code>DashboardComponentAlert</code>—generated when a dashboard metric or gauge exceeds a user-defined threshold.</li> <li>• <code>DashboardComponentSnapshot</code>—created when a user posts a dashboard snapshot on a feed.</li> <li>• <code>LinkPost</code>—a post with an attached URL.</li> <li>• <code>PollPost</code>—a poll posted on a feed.</li> <li>• <code>ProfileSkillPost</code>—generated when a skill is added to a user’s Chatter profile.</li> <li>• <code>QuestionPost</code>—generated when a user posts a question.</li> <li>• <code>ReplyPost</code>—generated when Chatter Answers posts a reply.</li> <li>• <code>RypplePost</code>—generated when a user creates a Thanks badge in Work.com.</li> <li>• <code>TextPost</code>—a direct text entry on a feed.</li> <li>• <code>TrackedChange</code>—a change or group of changes to a tracked field.</li> <li>• <code>UserStatus</code>—automatically generated when a user adds a post. Deprecated.</li> </ul> <p>The following values appear in the <code>Type</code> picklist for all feed objects but apply only to <code>CaseFeed</code>:</p> <ul style="list-style-type: none"> <li>• <code>CaseCommentPost</code>—generated event when a user adds a case comment for a case object</li> <li>• <code>EmailMessageEvent</code>—generated event when an email related to a case object is sent or received</li> <li>• <code>CallLogPost</code>—generated event when a user logs a call for a case through the user interface. CTI calls also generate this event.</li> <li>• <code>ChangeStatusPost</code>—generated event when a user changes the status of a case</li> <li>• <code>AttachArticleEvent</code>—generated event when a user attaches an article to a case</li> </ul> <p> <b>Note:</b> If you set <code>Type</code> to <code>ContentPost</code>, also specify <code>ContentData</code> and <code>ContentFileName</code>.</p>

## Usage

Use this object to track changes for an article. You can only delete a feed that you created, or if you have the “Modify All Data” permission or “Modify All” permission on the `KnowledgeArticle` object.

## KnowledgeArticle

Provides read-only access to an article and the ability to delete the master article. This object is available in API version 19.0 and later.

Unlike KnowledgeArticleVersion, the ID of a KnowledgeArticle record is identical irrespective of the article's version (status). For more information on articles and article types, see “Work with Articles and Translations” and “Knowledge Article Types” in the Salesforce online help.

## Supported Calls

`delete()`, `describeSObjects()`, `query()`, `retrieve()`

## Special Access Rules

A user must have the “View Articles” permission enabled. Salesforce Knowledge users, unlike Customer Portal and partner portal users, must also be granted the Knowledge User feature license.

## Fields

Field Name	Details
ArchivedById	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the user who archived the article.</p>
ArchivedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> The date when the article was archived.</p>
ArticleNumber	<p><b>Type</b> string</p> <p><b>Properties</b> Autonumber, Defaulted on create, Filter, Sort</p> <p><b>Description</b> The unique number automatically assigned to the article when it's created. You can't change the format or value for this field.</p>
CaseAssociationCount	<p><b>Type</b> int</p>

Field Name	Details
	<p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of cases attached to the article.</p>
FirstPublishedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> The date when the article was first published.</p>
LastPublishedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> The date when the article was last published.</p>
LastReferencedDate	<p><b>Type</b> date</p> <p><b>Properties</b> Filter, Nillable, Sort, Update</p> <p><b>Description</b> The timestamp for when the current user last viewed a record related to this record.</p>
LastViewedDate	<p><b>Type</b> date</p> <p><b>Properties</b> Filter, Nillable, Sort, Update</p> <p><b>Description</b> The timestamp for when the current user last viewed this record. If this value is null, this record might only have been referenced (<code>LastReferencedDate</code>) and not viewed.</p>
MasterLanguage	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Restricted picklist</p> <p><b>Description</b> The article's original language. Only accessible if your knowledge base supports multiple languages.</p>

## Usage

Use this object to query or retrieve articles. KnowledgeArticle can be used in a SOQL clause, but doesn't provide access to the fields from the article. Provides read-only access to an article and the ability to delete the master article.

## KnowledgeArticleVersion

Provides a global view of standard article fields across all article types depending on their version. This object is available in API version 18.0 and later.

Use this object to:

- Query or search generically across multiple article types.
- Filter on a specific version.
- Update standard fields, those fields that are updateable, in draft versions.

When you query on the archived article, the results include both the article and the article's archived versions.



**Note:** You can't update draft translations with the API.

## Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `query()`, `retrieve()`, `search()`, and `update()`.



**Note:**

- You can only update draft versions.
- To create, update, or delete a Knowledge article version, use the call on `ArticleType_kav`, where `ArticleType` is the name of the article's type. For example, to delete, use `ArticleType_kav.delete()`.

## Special Access Rules

A user must have the "View Articles" permission enabled. Salesforce Knowledge users, unlike Customer Portal and partner portal users, must also be granted the `Knowledge User` feature license to view and edit article versions.

## Fields

Field Name	Details
ArchivedById	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> The ID of the user who archived the article.</p>
ArchivedDate	<p><b>Type</b> dateTime</p>

Field Name	Details
	<p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> The date when the article was archived.</p>
ArticleNumber	<p><b>Type</b> string</p> <p><b>Properties</b> Autonumber, Defaulted on create, Filter</p> <p><b>Description</b> The unique number automatically assigned to the article when it's created. You can't change the format or value for this field.</p>
ArticleType	<p><b>Type</b> string</p> <p><b>Properties</b> Defaulted on create, Filter</p> <p><b>Description</b> Indicates the API Name of the article type. The <code>ArticleType</code> is assigned to the article when it's created. You can't change the value of this field. This field is available in API version 26.0 and later.</p>
FirstPublishedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> The date when the article was first published.</p>
IsLatestVersion	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Group, Filter, Sort</p> <p><b>Description</b> Indicates whether the article is the most current version. (<code>true</code>) or not (<code>false</code>). This field can be <code>true</code> on the online or published version, a draft version in the master language, a draft version in a translation, and the latest archived version. However, you can't filter by (<code>PublishState='Online'</code>) and (<code>IsLatestVersion=false</code>) because the online version is also the latest version. This field is available in API version 24.0 and later.</p>

Field Name	Details
IsMasterLanguage	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter</p> <p><b>Description</b> Indicates whether the article has one or more translations associated with it (<code>true</code>) or not (<code>false</code>). Only accessible if your knowledge base supports multiple languages.</p>
IsOutOfDate	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter</p> <p><b>Description</b> Indicates whether the source article has been updated since this translated version was created (<code>true</code>) or not (<code>false</code>). Only accessible if your knowledge base supports multiple languages.</p>
IsVisibleInApp	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter</p> <p><b>Description</b> Required. Indicates whether the article is visible in the Articles tab (<code>true</code>) or not (<code>false</code>).</p>
IsVisibleInCsp	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter</p> <p><b>Description</b> Required. Indicates whether the article is visible in the Customer Portal (<code>true</code>) or not (<code>false</code>).</p>
IsVisibleInPkb	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter</p> <p><b>Description</b> Required. Indicates whether the article is visible in the public knowledge base (<code>true</code>) or not (<code>false</code>).</p>

Field Name	Details
IsVisibleInPrm	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter</p> <p><b>Description</b> Required. Indicates whether the article is visible in the partner portal (<code>true</code>) or not (<code>false</code>).</p>
KnowledgeArticleId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter</p> <p><b>Description</b> The ID of the article independent from its version. The value for this field is retrieved from the <code>Id</code> field of the KnowledgeArticle object.</p>
Language	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Restricted picklist</p> <p><b>Description</b> The language that the article is written in, such as <code>French</code> or <code>Chinese (Traditional)</code>.  Querying or searching articles in SOQL or SOSL requires that you specify the <code>Language</code> field in the WHERE clause. The language must be the same for all article types.</p>
LastPublishedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> The date when the article was last published.</p>
MasterVersionId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> ID of the source article, if the article is the translation of a source article. Only accessible if your knowledge base supports multiple languages.</p>

Field Name	Details
OwnerId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter</p> <p><b>Description</b> The ID of the article's owner.</p>
PublishStatus	<p><b>Type</b> picklist</p> <p><b>Properties</b> Defaulted on create, Filter, Restricted picklist</p> <p><b>Description</b> The publication status for the article:</p> <ul style="list-style-type: none"> <li>• <b>Draft</b>: any draft articles.</li> <li>• <b>Online</b>: articles published in Salesforce Knowledge.</li> <li>• <b>Archived</b>: archived articles.</li> </ul> <p>A user must have the “Manage Articles” permission enabled to use <b>Online</b>.</p> <p>Querying or searching articles in SOQL or SOSL requires that you specify either the <b>PublishStatus</b> or the <b>Id</b> field in the WHERE clause. You can search for only one publication status per article type in a single SOSL query. When searching for articles with a <b>PublishStatus</b> of <b>Archived</b>, also check that <b>IsLatestVersion</b> equals <b>false</b> in your WHERE clause.</p>
SourceId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> ID of the source from which the article was created (Case or Reply).</p>
Summary	<p><b>Type</b> textarea</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> Summary of the article. Maximum size is 1000 characters.</p>
Title	<p><b>Type</b> string</p> <p><b>Properties</b> Defaulted on create, Filter, idLookup</p>



Field Name	Details
	<p><b>Description</b> Required. Article's title. Maximum size is 255 characters.</p>
TranslationCompletedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> Date and time when the article was last translated. Only accessible if your knowledge base supports multiple languages.</p>
TranslationExportedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> Date and time when the article was last exported for translation. Only accessible if your knowledge base supports multiple languages.</p>
TranslationImportedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> Date and time when the article was last imported for translation. Only accessible if your knowledge base supports multiple languages.</p>
UrlName	<p><b>Type</b> string</p> <p><b>Properties</b> Filter</p> <p><b>Description</b> Required. Represents the article's URL. Can contain alphanumeric characters and hyphens but can't begin or end with a hyphen. <code>UrlName</code> is case-sensitive and its maximum size is 255 characters.</p>
ValidationStatus	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group</p>

Field Name	Details
	<p><b>Description</b></p> <p>Shows whether the content of the article has been validated. Default values are <code>Validated</code> and <code>Not Validated</code>. This field is available in API version 24.0 or later.</p>
<code>VersionNumber</code>	<p><b>Type</b></p> <p>int</p> <p><b>Properties</b></p> <p>None</p> <p><b>Description</b></p> <p>The number assigned to a version of an article. This field is available in API version 24.0 and later.</p>

## Usage

Use this object to query, retrieve, or search for articles across all article types depending on their version. You can update draft master articles. Additionally, you can delete articles that are not drafts. Client applications can use `KnowledgeArticleVersion` with `describeDataCategoryGroups()` and `describeDataCategoryGroupStructures()` to return the category groups and the category structure associated with Salesforce Knowledge.

Alternatively, client applications can use the article type `API Name` followed by `__kav` to query, retrieve, or search for articles for a specific article type.

To access an article independent of its version, use the `KnowledgeArticle` object.

## SOQL Samples

The following SOQL clause uses `KnowledgeArticleVersion` to query all published articles from **all article types** complying with the classification specified in the `WITH DATA CATEGORY` clause:

```
SELECT Title, Summary
FROM KnowledgeArticleVersion
WHERE PublishStatus='Online'
AND Language = 'en_US'
WITH DATA CATEGORY Geography__c ABOVE_OR_BELOW europe__c AND Product__c BELOW All__c
```

The following SOQL clause uses the article type `API Name` to limit the query to all draft articles from the `Offer__kav` article type:

```
SELECT Id, Title
FROM Offer__kav
WHERE PublishStatus='Draft'
AND Language = 'en_US'
WITH DATA CATEGORY Geography__c AT (france__c,usa__c) AND Product__c ABOVE dsl__c
```

The following SOQL clause uses `KnowledgeArticleVersion` to query the Ids of all archived versions of a particular article:

```
SELECT Id
FROM KnowledgeArticleVersion
WHERE PublishStatus='Archived'
```

```
AND IsLatestVersion=false
AND KnowledgeArticleId='kA1D00000001PQ6KAM'
```

## SOQL and SOSL with KnowledgeArticleVersion

- Always filter on a single value of `PublishStatus` unless the query filters on one or more primary key IDs. To support security, only users with the “Manage Articles” permission see articles whose `PublishStatus` value is `Draft`.
- Archived article versions are stored in the `articletype_kav` object. To query archived article versions, specify the article `Id` and set `IsLatestVersion='0'`.
- Always filter on a single value of `Language`. However, in SOQL, you can filter on more than one `Language` if there is a filter on `Id` or `KnowledgeArticleId`.

## KnowledgeArticleVersionHistory

Enables read-only access to the full history of an article. This object is available in API version 25.0 and later.

### Supported Calls

`describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

### Special Access Rules

This object respects field, entity, and record-level security. You must have at least “Read” permission on the article type or the field to access its history. For data category security, Salesforce determines access based on the categorization of the online version of an article. If there is no online version, then security is applied based on the archived version, followed by the security of the draft version.

### Fields

Field Name	Details
<code>EventType</code>	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Restricted picklist, Sort</p> <p><b>Description</b> The type of event that is tracked in the history table.</p>
<code>FieldName</code>	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> Name of the tracked field.</p>

Field Name	Details
Language	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Restricted picklist, Sort</p> <p><b>Description</b> The language that the article is written in, such as <code>French</code> or <code>Chinese (Traditional)</code>. Querying or searching articles in SOQL or SOSL requires that you specify the <code>Language</code> field in the WHERE clause. The language must be the same for all article types.</p>
NewValue	<p><b>Type</b> anyType</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The new value of the field that was changed.</p>
OldValue	<p><b>Type</b> anyType</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The most recent value of the field before it was changed.</p>
ParentId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The ID of the article.</p>
ParentObjectType	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Restricted picklist, Sort</p> <p><b>Description</b> The type of object that contains the field.</p>
VersionId	<p><b>Type</b> reference</p>

Field Name	Details
	<p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID assigned to a version of the article.</p>
VersionNumber	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number assigned to a version of an article. This field is available in API version 24.0 and later.</p>

## Usage

Use this object to query events in the history of an article. For example, you can retrieve the number of edits a particular user has made to an article, how many times the article has been published, and so on.

## KnowledgeArticleViewStat

Provides statistics on the number of views for the specified article across all article types. This object is read-only and available in API version 20 and later.

## Supported Calls

`describeSObjects()`, `query()`, `retrieve()`

## Special Access Rules

Users must have access to the published version of an article to retrieve its views. For more information on published article version, see the `PublishStatus` field in `KnowledgeArticleVersion`.

## Fields

Field Name	Details
Channel	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Restricted picklist</p> <p><b>Description</b> The channel where the article is viewed:</p>

Field Name	Details
	<ul style="list-style-type: none"> <li>• <code>AllChannels</code> for article views across all channels.</li> <li>• <code>App</code> for the internal Salesforce Knowledge application.</li> <li>• <code>Pkb</code> for article views in public knowledge base.</li> <li>• <code>Csp</code> for Customer Portal.</li> <li>• <code>Pm</code> for article view in partner portal.</li> </ul>
<code>NormalizedScore</code>	<p><b>Type</b> double</p> <p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> Article's weighted views in the selected channel. The article with most views has a score of 100. Other article views are then calculated relative to this highest view score. For example, if the best read article has 2000 views and another has 1000. The first one gets a score of 100 while the second gets 50.</p>
<code>ParentId</code>	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter</p> <p><b>Description</b> ID of the viewed article. This corresponds to a KnowledgeArticle record.</p>
<code>ViewCount</code>	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of unique views an article has received in the selected channel. An article with a high number of views may not always have a high normalized score. The normalized score for an article is calculated based on views over time, with more recent views earning a higher score. This field is available in API version 27.0 and later.</p>

## Usage

Use this object to query or retrieve statistics for article views.

Alternatively, client applications can use the article type `API Name` followed by `__ViewStat` to query or retrieve most viewed articles from a specific article type.

## SOQL Samples

The following SOQL clause uses KnowledgeArticleViewStat to query all the article views in Salesforce Knowledge and return the related articles:

```
SELECT Id, NormalizedScore, Parent.Id
      FROM KnowledgeArticleViewStat where Channel = 'App'
      ORDER BY NormalizedScore
```

Use the following clause to restrict your query to Offer articles for the Offer article type:

```
SELECT Id, NormalizedScore, Parent.Id
      FROM Offer__ViewStat where Channel = 'App'
      ORDER BY NormalizedScore
```

## KnowledgeArticleVoteStat

Provides the weighted rating for the specified article on a scale of 1 to 5 across all article types. This object is read-only and available in API version 20 and later.

## Supported Calls

`describeSObjects()`, `query()`, `retrieve()`

## Special Access Rules

Users must have access to the published version of an article to retrieve its votes. For more information on published article version, see the `PublishStatus` field in KnowledgeArticleVersion

## Fields

Field Name	Details
Channel	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Restricted picklist</p> <p><b>Description</b> The channel where the article is rated:</p> <ul style="list-style-type: none"> <li>• <code>AllChannels</code> for article views across all channels.</li> <li>• <code>App</code> for the internal Salesforce Knowledge application.</li> <li>• <code>Pkb</code> for article views in public knowledge base.</li> <li>• <code>Csp</code> for Customer Portal.</li> <li>• <code>Prm</code> for article view in partner portal.</li> </ul>
NormalizedScore	<p><b>Type</b> double</p>

Field Name	Details
	<p><b>Properties</b> Filter, Nillable</p> <p><b>Description</b> Article's weighted score on a scale of 1 to 5. A higher score means more votes. Articles without recent votes trend towards an average rating of three stars.</p>
ParentId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter</p> <p><b>Description</b> The rated article. This corresponds to a KnowledgeArticle record.</p>

## Usage

Use this object to query or retrieve the rating for an article.

Alternatively, client applications can use the article type `API Name` followed by `__VoteStat` to query or retrieve the rating for an article for a specific article type.

## SOQL Samples

See KnowledgeArticleViewStat.

## CaseArticle

Represents the association between a Case and a KnowledgeArticle. This object is available in API version 20.0 and later.

## Supported Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`

## Special Access Rules

Access to this object is controlled by the parent Case and KnowledgeArticle. However, when querying, access is only controlled by the parent Case.

Customer Portal users can't access this object.



## Fields

Field	Details
ArticleLanguage	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Restricted picklist</p> <p><b>Description</b> The language of the article associated with the case.</p>
ArticleVersionNumber	<p><b>Type</b> int</p> <p><b>Properties</b> Create, Group, Nillable</p> <p><b>Description</b> The number assigned to a version of an article. This field is available in API version 24.0 and later.</p>
CaseId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> ID of the Case associated with the KnowledgeArticle.</p>
IsSharedByEmail	<p><b>Type</b> int</p> <p><b>Properties</b> Create, Group, Nillable</p> <p><b>Description</b> Indicates that the article has been shared with the customer through an email.</p>
KnowledgeArticleId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> ID of the KnowledgeArticle associated with the Case.</p>

## Usage

This object represents the association of a knowledge article with a Case. An article is associated with a case when it's relevant to a specific issue, when it helps an agent solve the case, or when the agent sends the article to a customer.

You can use this object to include case-article associations in Apex and Visualforce.

You can't update this object via the API. If you attempt to create a record that matches an existing record, the create request simply returns the existing record.

## LinkedArticle

Represents a Knowledge article that is attached to a record. This object is available in API version 37.0 and later.

As of Summer '16, linked articles are supported only for work orders and work order line items.

## Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

## Special Access Rules

Knowledge must be set up in your org.

## Fields

Field Name	Details
CurrencyIsoCode	<p><b>Type</b> picklist</p> <p><b>Properties</b> Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update</p> <p><b>Description</b> Available only for orgs with the multicurrency feature enabled. Contains the ISO code for any currency allowed by the organization.</p>
KnowledgeArticleId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the Knowledge article attached to the record. The label in the user interface is Knowledge Article ID.</p>
KnowledgeArticleVersionId	<p><b>Type</b> reference</p>

Field Name	Details
	<p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p> <p><b>Description</b> The version of the Knowledge article attached to the record. This field lists the title of the attached version and links to the version. The label in the user interface is Article Version.</p> <p>When you attach an article to a work order, that version of the article stays associated with the work order, even if later versions are published. If needed, you can detach and reattach an article to a work order to link the latest version.</p>
LinkedEntityId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the record that the Knowledge article is attached to. The label in the user interface is Linked Record ID.</p>
Name	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The title of the article. The label in the user interface is Article Title.</p>
RecordTypeId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort, Update</p> <p><b>Description</b> The ID of the article's record type, if used.</p>
Type	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> (Read only) The type of record that the Knowledge article is attached to. For example, work order. The label in the user interface is Linked Object Type.</p>

## Usage

Admins can customize linked articles' page layouts, fields, validation rules, and more from the Linked Articles node in Setup under Knowledge.

## LinkedArticleFeed

Represents a single feed item on a linked Knowledge article attached to a record. This object is available in API version 37.0 and later. As of Summer '16, linked articles are supported only for work orders and work order line items.

## Supported Calls


`delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

## Special Access Rules

Knowledge must be set up in your org.

## Fields

Field Name	Details
Body	<p><b>Type</b> textarea</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The content of the feed item. Required when <code>Type</code> is <code>TextPost</code>. Optional when <code>Type</code> is <code>ContentPost</code> or <code>LinkPost</code>.</p>
CommentCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of comments associated with the feed item.</p>
InsertedById	<p><b>Type</b> reference</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> ID of the user who added this object to the feed. For example, if an application migrates posts and comments from another application into a feed, the <code>InsertedBy</code> value is set to the ID of the logged-in user.</p>

Field Name	Details
IsRichText	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether the feed item <code>Body</code> contains rich text. Set <code>IsRichText</code> to <code>true</code> if you post a rich text feed item via the SOAP API. Otherwise, the post is rendered as plain text.</p> <p>Rich text supports the following HTML tags.</p> <ul style="list-style-type: none"><li>• <code>&lt;p&gt;</code></li></ul> <p> <b>Tip:</b> Although the <code>&lt;br&gt;</code> tag isn't supported, you can use <code>&lt;p&gt;&amp;nbsp;&lt;/p&gt;</code> to create lines.</p> <ul style="list-style-type: none"><li>• <code>&lt;b&gt;</code></li><li>• <code>&lt;i&gt;</code></li><li>• <code>&lt;u&gt;</code></li><li>• <code>&lt;s&gt;</code></li><li>• <code>&lt;ul&gt;</code></li><li>• <code>&lt;ol&gt;</code></li><li>• <code>&lt;li&gt;</code></li><li>• <code>&lt;img&gt;</code></li></ul> <p>The <code>&lt;img&gt;</code> tag is accessible only via the API and must reference files in Salesforce similar to this example: <code>&lt;img src="sfdc://069B0000000omjh"&gt;&lt;/img&gt;</code></p>
LikeCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of Likes associated with the feed item.</p>
LinkUrl	<p><b>Type</b> url</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The URL of a <code>LinkPost</code>.</p>
NetworkScope	<p><b>Type</b> picklist</p>

Field Name	Details
	<p><b>Properties</b> Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> Specifies whether the feed item is available in the default community, a specific community, or all communities.</p>
ParentId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The ID of the object type to which the feed item is related. For example, set this field to a <code>UserId</code> to post to someone's profile feed. Or set it to an <code>AccountId</code> to post to a specific account.</p>
RelatedRecordId	<p><b>Type</b> reference</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> The ID of the <code>ContentVersion</code> object associated with a content post. This field is null for all posts, except <code>ContentPost</code>.</p>
Title	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> The title of the feed item. When the <code>Type</code> is <code>LinkPost</code>, the <code>LinkUrl</code> is the URL, and this field is the link name.</p>
Type	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> The type of feed item.</p>
Visibility	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Restricted picklist, Sort</p>

**Field Name****Details****Description**

Specifies whether this feed item is available to all users or only internal users.

## LinkedArticleHistory

Represents the history of changes made to tracked fields on a linked article. This object is available in API version 37.0 and later.

As of Summer '16, linked articles are supported only for work orders and work order line items.

## Supported Calls

`describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

## Special Access Rules

Knowledge must be set up in your org.

## Fields

**Field Name****Details**

Field

**Type**

picklist

**Properties**

Filter, Group, Restricted picklist, Sort

**Description**

The name of the field that was changed.

LinkedArticleId

**Type**

reference

**Properties**

Filter, Group, Sort

**Description**

The ID of the tracked linked article. The history is displayed on the detail page for this record.

NewValue

**Type**

anyType

**Properties**

Nullable, Sort

**Description**

The new value of the field that was changed.

Field Name	Details
OldValue	<p><b>Type</b> anyType</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The value of the field before it was changed.</p>

## NewsFeed


Represents a single feed item on a user's home tab. A Chatter feed shows recent changes to records that the user is following.

NewsFeed is available in API version 18.0 through API version 26.0. In API version 27.0 and later, NewsFeed is no longer available in the SOAP API. Use the Chatter REST API to access NewsFeed.

## Supported Calls

`delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

## Fields

Field	Details
Body	<p><b>Type</b> textarea</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The content of NewsFeed. Required when <code>Type</code> is <code>TextPost</code>. Optional when <code>Type</code> is <code>ContentPost</code> or <code>LinkPost</code>. This is the message that appears in the feed.</p>
CommentCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of FeedComments associated with this feed item.</p> <p> <b>Tip:</b> In a feed that supports pre-moderation, <code>CommentCount</code> isn't updated until a comment is published. For example, say you comment on a post that already has one published comment and your comment triggers moderation. Now there are two comments on the post, but the count says there's only one. In a moderated feed, your comment isn't counted until it's approved by an admin or a person with "CanApproveFeedPostAndComment" or "ModifyAllData" permission.</p>



Field	Details
	<p>This has implications for how you retrieve feed comments. In a moderated feed, rather than retrieving comments by looping through <code>CommentCount</code>, go through pagination until end of comments is returned.</p>
ConnectionId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> When a <code>PartnerNetworkConnection</code> makes a change to a record that is tracked, the <code>CreatedBy</code> field contains the ID of the system administrator while the <code>ConnectionId</code> contains the ID of the <code>PartnerNetworkConnection</code> that made the change. Available if Salesforce to Salesforce is enabled for your organization.</p>
ContentData	<p><b>Type</b> base64</p> <p><b>Properties</b> Nillable</p> <p><b>Description</b> Available in API version 36.0 and earlier only. Required if <code>Type</code> is <code>ContentPost</code>. Encoded file data in any format, and can't be 0 bytes. Setting this field automatically sets <code>Type</code> to <code>ContentPost</code>.</p>
ContentDescription	<p><b>Type</b> textarea</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The description of the file specified in <code>ContentData</code>.</p>
ContentFileName	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. This field is required if <code>Type</code> is <code>ContentPost</code>. The name of the file uploaded to the feed. Setting <code>ContentFileName</code> automatically sets <code>Type</code> to <code>ContentPost</code>.</p>
ContentSize	<p><b>Type</b> int</p>

Field	Details
	<p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The size of the file (in bytes) uploaded to the feed. This field is read-only and is automatically determined during insert.</p>
ContentType	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The MIME type of the file uploaded to the feed. This field is read-only and is automatically determined during insert.</p>
CreatedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Date and time when this record was created. This is a standard system field. Ordering by <code>CreatedDateDESC</code> sorts the feed by the most recent feed item.</p>
FeedPostId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> This field was removed in API version 22.0, and is available in earlier versions for backward compatibility only. ID of the associated FeedPost. A FeedPost represents the following types of changes in an FeedItem: changes to tracked fields, text posts, link posts, and content posts.</p>
InsertedById	<p><b>Type</b> reference</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> ID of the user who added this object to the feed. For example, if an application migrates posts and comments from another application into a feed, the <code>InsertedBy</code> value is set to the ID of the logged-in user.</p>

Field	Details
IsDeleted	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Standard system field. Indicates whether the record has been moved to the Recycle Bin (<code>true</code>) or not (<code>false</code>).</p>
LastModifiedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Date and time when a user last modified this record. This field is a standard system field.  When a feed item is created, <code>LastModifiedDate</code> is the same as <code>CreatedDate</code>. If a <code>FeedComment</code> is inserted on that feed item, then <code>LastModifiedDate</code> becomes the <code>CreatedDate</code> for that <code>FeedComment</code>. Deleting the <code>FeedComment</code> does not change the <code>LastModifiedDate</code>.  Ordering by <code>LastModifiedDate DESC</code> sorts the feed by both the most recent feed item or comment.</p>
LikeCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of <code>FeedLikes</code> associated with this feed item.</p>
LinkUrl	<p><b>Type</b> url</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The URL of a <code>LinkPost</code>.</p>
NetworkScope	<p><b>Type</b> picklist</p> <p><b>Properties</b> Group, Nillable, Restricted picklist, Sort</p>

Field	Details
	<p><b>Description</b></p> <p>Specifies whether this feed item is available in the default community, a specific community, or all communities. This field is available in API version 26.0 and later, if Salesforce Communities is enabled for your organization.</p> <p><code>NetworkScope</code> can have the following values:</p> <ul style="list-style-type: none"> <li><code>NetworkId</code>—The ID of the community in which the <code>FeedItem</code> is available. If left empty, the feed item is only available in the default community.</li> <li><code>AllNetworks</code>—The feed item is available in all communities.</li> </ul> <p>Note the following exceptions for <code>NetworkScope</code>:</p> <ul style="list-style-type: none"> <li>Only feed items with a <code>Group</code> or <code>User</code> parent can set a <code>NetworkId</code> or a null value for <code>NetworkScope</code>.</li> <li>For feed items with a record parent, users can set <code>NetworkScope</code> only to <code>AllNetworks</code>.</li> <li>You can't filter a <code>FeedItem</code> on the <code>NetworkScope</code> field.</li> </ul>
ParentId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Filter, Group, Sort</p> <p><b>Description</b></p> <p>ID of the newsfeed record that is tracked in the feed. The feed is displayed on the detail page for this record.</p>
RelatedRecordId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Group, Nillable, Sort</p> <p><b>Description</b></p> <p>ID of the <code>ContentVersion</code> object associated with a <code>ContentPost</code>. This field is null for all posts except <code>ContentPost</code>.</p>
Title	<p><b>Type</b></p> <p>string</p> <p><b>Properties</b></p> <p>Group, Nillable, Sort</p> <p><b>Description</b></p> <p>The title of the feed item. When the <code>Type</code> is <code>LinkPost</code>, the <code>LinkUrl</code> is the URL and this field is the link name.</p>
Type	<p><b>Type</b></p> <p>picklist</p>

**Field****Details****Properties**

Filter, Group, Nillable, Restricted picklist, Sort

**Description**

The type of feed item:


- **ActivityEvent**—indirectly generated event when a user or the API adds a Task associated with a feed-enabled parent record (excluding email tasks on cases). Also occurs when a user or the API adds or updates a Task or Event associated with a case record (excluding email and call logging).

For a recurring Task with CaseFeed disabled, one event is generated for the series only. For a recurring Task with CaseFeed enabled, events are generated for the series and each occurrence.

- **AdvancedTextPost**—created when a user posts a group announcement.
- **AnnouncementPost**—Not used.
- **ApprovalPost**—generated when a user submits an approval.
- **BasicTemplateFeedItem**—Not used.
- **CanvasPost**—a post made by a canvas app posted on a feed.
- **CollaborationGroupCreated**—generated when a user creates a public group.
- **CollaborationGroupUnarchived**—Not used.
- **ContentPost**—a post with an attached file.
- **CreatedRecordEvent**—generated when a user creates a record from the publisher.
- **DashboardComponentAlert**—generated when a dashboard metric or gauge exceeds a user-defined threshold.
- **DashboardComponentSnapshot**—created when a user posts a dashboard snapshot on a feed.
- **LinkPost**—a post with an attached URL.
- **PollPost**—a poll posted on a feed.
- **ProfileSkillPost**—generated when a skill is added to a user's Chatter profile.
- **QuestionPost**—generated when a user posts a question.
- **ReplyPost**—generated when Chatter Answers posts a reply.
- **RypplePost**—generated when a user creates a Thanks badge in Work.com.
- **TextPost**—a direct text entry on a feed.
- **TrackedChange**—a change or group of changes to a tracked field.
- **UserStatus**—automatically generated when a user adds a post. Deprecated.

The following values appear in the `Type` picklist for all feed objects but apply only to CaseFeed:

- **CaseCommentPost**—generated event when a user adds a case comment for a case object
- **EmailMessageEvent**—generated event when an email related to a case object is sent or received

Field	Details
	<ul style="list-style-type: none"> <li>• <code>CallLogPost</code>—generated event when a user logs a call for a case through the user interface. This event is also generated by CTI calls.</li> <li>• <code>ChangeStatusPost</code>—generated event when a user changes the status of a case</li> <li>• <code>AttachArticleEvent</code>—generated event when a user attaches an article to a case</li> </ul> <p> <b>Note:</b> If you set <code>Type</code> to <code>ContentPost</code>, you must specify <code>ContentData</code> and <code>ContentFileName</code>.</p>
<code>Visibility</code>	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> Specifies whether this feed item is available to all users or internal users only. This field is available in API version 26.0 and later, if Salesforce Communities is enabled for your organization.</p> <p><code>Visibility</code> can have the following values:</p> <ul style="list-style-type: none"> <li>• <code>AllUsers</code>—The feed item is available to all users who have permission to see the feed item.</li> <li>• <code>InternalUsers</code>—The feed item is available to internal users only.</li> </ul> <p>Note the following exceptions for <code>Visibility</code>:</p> <ul style="list-style-type: none"> <li>• For record posts, <code>Visibility</code> is set to <code>InternalUsers</code> for all internal users by default.</li> <li>• External users can set <code>Visibility</code> only to <code>AllUsers</code>.</li> <li>• On user and group posts, only internal users can set <code>Visibility</code> to <code>InternalUsers</code>.</li> </ul>


## Usage

Use this object to query and retrieve record changes tracked in a news feed. Note the following when working with news feeds:

- This object is read only in the API.
- Users can only query their own news feed.
- Queries retrieve feed items that include mentions from other users.
- Use this object to query and retrieve lead feed items that were associated with a converted lead record.

Users that do not have the “View All Data” permission have the following limitations when querying records:

- Must specify a `LIMIT` clause and the limit must be less than or equal to 1000.
- May include a `WHERE` clause that references `NewsFeed` fields, but may not include references to fields in related objects. For example, you can filter by `CreatedDate` or `ParentId`, but not by `Parent.Name`.
- May include an `ORDER BY` clause that references `NewsFeed` fields, but may not include references to fields in related objects. For example, you can `ORDER BY CreatedDate` or `ParentId`, but not by `Parent.Name`.

 **Tip:** To query for the most recent feed items, you should `ORDER BY CreatedDate DESC, Id DESC`.

Note the following SOQL restrictions.

- No SOQL limit if logged-in user has “View All Data” permission. If not, specify a LIMIT clause of 1,000 records or fewer.
- SOQL ORDER BY on fields using relationships is not available. Use ORDER BY on fields on the root object in the SOQL query.

## RecentlyViewed

Represents records that the current user has recently viewed or referenced (by viewing a related record).

## Supported Calls

`query()`, `update()`

## Special Usage Rules

The RecentlyViewed object does not support the Report, KnowledgeArticle, and Article objects.

## Fields

Field	Details
Alias	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The alias on the record.</p>
Email	<p><b>Type</b> email</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The email address on the record.</p>
FirstName	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The first name on the record. If the recently viewed record is a user, this is the user’s first name.</p>

Field	Details
Id	<p><b>Type</b> ID</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the recently viewed record.</p>
IsActive	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether the recently viewed record is an active user (<code>true</code>) or not (<code>false</code>). This field contains a value only if the recently viewed record is a user.</p>
LastName	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The last name on the record.</p>
LastReferencedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort, Update</p> <p><b>Description</b> The timestamp for when the current user last viewed a record related to this record.</p>
LastViewedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort, Update</p> <p><b>Description</b> The timestamp for when the current user last viewed this record. If this value is null, this record might only have been referenced (<code>LastReferencedDate</code>) and not viewed.</p>
Name	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p>



Field	Details
	<p><b>Description</b></p> <p>The name on the recently viewed record. If the recently viewed record is a user, contact, or lead, the value is a concatenation of the <code>firstname</code> and <code>lastname</code> field values.</p>
NetworkId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Filter, Group, Nillable, Sort</p> <p><b>Description</b></p> <p>ID of the community that this group is part of. This field is available only if Salesforce Communities is enabled in your organization.</p> <p>You can add a <code>NetworkId</code> only when creating a group. You can't change or add a <code>NetworkId</code> for an existing group. This field is available in API version 27.0 and later.</p>
Phone	<p><b>Type</b></p> <p>phone</p> <p><b>Properties</b></p> <p>Filter, Group, Nillable, Sort</p> <p><b>Description</b></p> <p>The phone number on the record.</p>
ProfileId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Filter, Group, Nillable, Sort</p> <p><b>Description</b></p> <p>If the recently viewed record is a user, this is the user's profile ID.</p>
Title	<p><b>Type</b></p> <p>string</p> <p><b>Properties</b></p> <p>Filter, Group, Nillable, Sort</p> <p><b>Description</b></p> <p>If the recently viewed record is a user, this is the title of the user; for example CFO or CEO.</p>
Type	<p><b>Type</b></p> <p>picklist</p> <p><b>Properties</b></p> <p>Filter, Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b></p> <p>The sObject type for this recently viewed record. Valid values include any standard or custom objects that RecentlyViewed supports.</p>

Field	Details
UserRoleId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the user role associated with this object.</p>

## Usage

This object provides a heterogeneous list of different object types and consists of recently viewed records or records that were recently referenced (a related record was viewed). A record is considered viewed when the user sees the details associated with it, but not when the user sees it in a list with other records. Use this object to programmatically construct a list of recently viewed items specific to the current user, for example, on a custom user interface or for search auto-complete options. You can also retrieve a filtered list of records by object type (`Type`). The `RecentlyViewed` data is periodically truncated down to 200 records per object.

Use this query in your code to retrieve a list of all the records that were recently viewed. The results are ordered from most to least recent.

```
SELECT Id, Name
FROM RecentlyViewed
WHERE LastViewedDate !=null
ORDER BY LastViewedDate DESC
```

Use this query to retrieve data that was either viewed or referenced, but only for a limited set of objects.

```
SELECT Id, Name
FROM RecentlyViewed
WHERE Type IN ('Account', 'Contact', 'Plan__c')
ORDER BY LastViewedDate DESC
```

This query retrieves a list of all recently viewed contacts with contact-specific fields, such as the contact's account name, and the custom website field. Records are ordered from most to least recent.

```
SELECT Account.Name, Title, Email, Phone, Website__c
FROM Contact
WHERE LastViewedDate != NULL
ORDER BY LastViewedDate DESC
```

## ReportFeed

Represents a single feed item in the feed displayed on a report. This object is available in API version 20.0 and later.

### Supported Calls

`delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

### Special Access Rules


You can delete all feed items you created. To delete feed items you didn't create, you must have one of these permissions:

- “Modify All Data”
- “Moderate Chatter”





**Note:** Users with the “Moderate Chatter” permission can delete only the feed items and comments they see.

## Fields

Field	Details
Body	<p><b>Type</b> textarea</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The content of ReportFeed. Required when <code>Type</code> is <code>TextPost</code>. Optional when <code>Type</code> is <code>ContentPost</code> or <code>LinkPost</code>. This field is the message that appears in the feed.</p>
CommentCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of FeedComments associated with this feed item.</p> <p> <b>Tip:</b> In a feed that supports pre-moderation, <code>CommentCount</code> isn't updated until a comment is published. For example, say you comment on a post that already has one published comment and your comment triggers moderation. Now there are two comments on the post, but the count says there's only one. In a moderated feed, your comment isn't counted until it's approved by an admin or a person with “CanApproveFeedPostAndComment” or “ModifyAllData” permission.</p> <p>This has implications for how you retrieve feed comments. In a moderated feed, rather than retrieving comments by looping through <code>CommentCount</code>, go through pagination until end of comments is returned.</p>
ConnectionId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> When a PartnerNetworkConnection modifies a record that is tracked, the <code>CreatedBy</code> field contains the ID of the system administrator. The <code>ConnectionId</code> contains the ID of the PartnerNetworkConnection. Available if Salesforce to Salesforce is enabled for your organization.</p>

Field	Details
ContentData	<p><b>Type</b> base64</p> <p><b>Properties</b> Nillable</p> <p><b>Description</b> Available in API version 36.0 and earlier only. Required if <code>Type</code> is <code>ContentPost</code>. Encoded file data in any format, and can't be 0 bytes. Setting this field automatically sets <code>Type</code> to <code>ContentPost</code>.</p>
ContentDescription	<p><b>Type</b> textarea</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The description of the file specified in <code>ContentData</code>.</p>
ContentFileName	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. This field is required if <code>Type</code> is <code>ContentPost</code>. The name of the file uploaded to the feed. Setting <code>ContentFileName</code> automatically sets <code>Type</code> to <code>ContentPost</code>.</p>
ContentSize	<p><b>Type</b> int</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The size of the file (in bytes) uploaded to the feed. This field is read-only and is automatically determined during insert.</p>
ContentType	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The MIME type of the file uploaded to the feed. This field is read-only and is automatically determined during insert.</p>


Field	Details
CreatedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Date and time when this record was created. This field is a standard system field. Ordering by <code>CreatedDateDESC</code> sorts the feed by the most recent feed item.</p>
FeedPostId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> This field was removed in API version 22.0, and is available in earlier versions for backward compatibility only. ID of the associated FeedPost. A FeedPost represents the following types of changes in an FeedItem: changes to tracked fields, text posts, link posts, and content posts.</p>
InsertedById	<p><b>Type</b> reference</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> ID of the user who added this object to the feed. For example, if an application migrates posts and comments from another application into a feed, the <code>InsertedBy</code> value is set to the ID of the logged-in user.</p>
IsDeleted	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Standard system field. Indicates whether the record has been moved to the Recycle Bin (<code>true</code>) or not (<code>false</code>).</p>
IsRichText	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether the feed item <code>Body</code> contains rich text. Set <code>IsRichText</code> to <code>true</code> if you post a rich text feed item via SOAP API. Otherwise, the post is rendered as plain text.</p>

Field	Details
	<p>Rich text supports the following HTML tags:</p> <ul style="list-style-type: none"> <li>• <code>&lt;p&gt;</code></li> <li>•  <b>Tip:</b> Though the <code>&lt;br&gt;</code> tag isn't supported, you can use <code>&lt;p&gt;&amp;nbsp;&lt;/p&gt;</code> to create lines.</li> <li>• <code>&lt;b&gt;</code></li> <li>• <code>&lt;code&gt;</code></li> <li>• <code>&lt;i&gt;</code></li> <li>• <code>&lt;u&gt;</code></li> <li>• <code>&lt;s&gt;</code></li> <li>• <code>&lt;ul&gt;</code></li> <li>• <code>&lt;ol&gt;</code></li> <li>• <code>&lt;li&gt;</code></li> <li>• <code>&lt;img&gt;</code></li> </ul> <p>The <code>&lt;img&gt;</code> tag is accessible only via the API and must reference files in Salesforce similar to this example: <code>&lt;img src="sfdc://069B0000000omjh"&gt;&lt;/img&gt;</code></p> <p> <b>Note:</b> In API version 35.0 and later, the system replaces special characters in rich text with escaped HTML. In API version 34.0 and prior, all rich text appears as a plain-text representation.</p>
LastModifiedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Date and time when a user last modified this record. This field is a standard system field.</p> <p>When a feed item is created, <code>LastModifiedDate</code> is the same as <code>CreatedDate</code>. If a <code>FeedComment</code> is inserted on that feed item, then <code>LastModifiedDate</code> becomes the <code>CreatedDate</code> for that <code>FeedComment</code>. Deleting the <code>FeedComment</code> does not change the <code>LastModifiedDate</code>.</p> <p>Ordering by <code>LastModifiedDate DESC</code> sorts the feed by both the most recent feed item or comment.</p>
LikeCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of <code>FeedLikes</code> associated with this feed item.</p>

Field	Details
LinkUrl	<p><b>Type</b> url</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The URL of a <code>LinkPost</code>.</p>
NetworkScope	<p><b>Type</b> picklist</p> <p><b>Properties</b> Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> Specifies whether this feed item is available in the default community, a specific community, or all communities. This field is available in API version 26.0 and later, if Salesforce Communities is enabled for your organization.</p> <p><code>NetworkScope</code> can have the following values:</p> <ul style="list-style-type: none"> <li>• <code>NetworkId</code>—The ID of the community in which the <code>FeedItem</code> is available. If left empty, the feed item is only available in the default community.</li> <li>• <code>AllNetworks</code>—The feed item is available in all communities.</li> </ul> <p>Note the following exceptions for <code>NetworkScope</code>:</p> <ul style="list-style-type: none"> <li>• Only feed items with a <code>Group</code> or <code>User</code> parent can set a <code>NetworkId</code> or a null value for <code>NetworkScope</code>.</li> <li>• For feed items with a record parent, users can set <code>NetworkScope</code> only to <code>AllNetworks</code>.</li> <li>• You can't filter a <code>FeedItem</code> on the <code>NetworkScope</code> field.</li> </ul>
ParentId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> ID of the account record that is tracked in the feed. The feed is displayed on the detail page for this record.</p>
RelatedRecordId	<p><b>Type</b> reference</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> ID of the <code>ContentVersion</code> object associated with a <code>ContentPost</code>. This field is null for all posts except <code>ContentPost</code>.</p>

Field	Details
Title	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> The title of the report record. When the <code>Type</code> is <code>LinkPost</code>, the <code>LinkUrl</code> is the URL and this field is the link name.</p>
Type	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> The type of feed item:</p> <ul style="list-style-type: none"> <li>• <code>ActivityEvent</code>—indirectly generated event when a user or the API adds a Task associated with a feed-enabled parent record (excluding email tasks on cases). Also occurs when a user or the API adds or updates a Task or Event associated with a case record (excluding email and call logging). For a recurring Task with <code>CaseFeed</code> disabled, one event is generated for the series only. For a recurring Task with <code>CaseFeed</code> enabled, events are generated for the series and each occurrence.</li> <li>• <code>AdvancedTextPost</code>—created when a user posts a group announcement.</li> <li>• <code>AnnouncementPost</code>—Not used.</li> <li>• <code>ApprovalPost</code>—generated when a user submits an approval.</li> <li>• <code>BasicTemplateFeedItem</code>—Not used.</li> <li>• <code>CanvasPost</code>—a post made by a canvas app posted on a feed.</li> <li>• <code>CollaborationGroupCreated</code>—generated when a user creates a public group.</li> <li>• <code>CollaborationGroupUnarchived</code>—Not used.</li> <li>• <code>ContentPost</code>—a post with an attached file.</li> <li>• <code>CreatedRecordEvent</code>—generated when a user creates a record from the publisher.</li> <li>• <code>DashboardComponentAlert</code>—generated when a dashboard metric or gauge exceeds a user-defined threshold.</li> <li>• <code>DashboardComponentSnapshot</code>—created when a user posts a dashboard snapshot on a feed.</li> <li>• <code>LinkPost</code>—a post with an attached URL.</li> <li>• <code>PollPost</code>—a poll posted on a feed.</li> <li>• <code>ProfileSkillPost</code>—generated when a skill is added to a user’s Chatter profile.</li> <li>• <code>QuestionPost</code>—generated when a user posts a question.</li> <li>• <code>ReplyPost</code>—generated when Chatter Answers posts a reply.</li> <li>• <code>RypplePost</code>—generated when a user creates a Thanks badge in Work.com.</li> </ul>



Field	Details
	<ul style="list-style-type: none"> <li>• <code>TextPost</code>—a direct text entry on a feed.</li> <li>• <code>TrackedChange</code>—a change or group of changes to a tracked field.</li> <li>• <code>UserStatus</code>—automatically generated when a user adds a post. Deprecated.</li> </ul> <p>The following values appear in the <code>Type</code> picklist for all feed objects but apply only to <code>CaseFeed</code>:</p> <ul style="list-style-type: none"> <li>• <code>CaseCommentPost</code>—generated event when a user adds a case comment for a case object</li> <li>• <code>EmailMessageEvent</code>—generated event when an email related to a case object is sent or received</li> <li>• <code>CallLogPost</code>—generated event when a user logs a call for a case through the user interface. CTI calls also generate this event.</li> <li>• <code>ChangeStatusPost</code>—generated event when a user changes the status of a case</li> <li>• <code>AttachArticleEvent</code>—generated event when a user attaches an article to a case</li> </ul> <p> <b>Note:</b> If you set <code>Type</code> to <code>ContentPost</code>, also specify <code>ContentData</code> and <code>ContentFileName</code>.</p>

## Visibility

### Type

picklist

### Properties

Filter, Group, Nillable, Restricted picklist, Sort

### Description

Specifies whether this feed item is available to all users or internal users only. This field is available in API version 26.0 and later, if Salesforce Communities is enabled for your organization.

`Visibility` can have the following values:

- `AllUsers`—The feed item is available to all users who have permission to see the feed item.
- `InternalUsers`—The feed item is available to internal users only.

Note the following exceptions for `Visibility`:

- For record posts, `Visibility` is set to `InternalUsers` for all internal users by default.
- External users can set `Visibility` only to `AllUsers`.
- On user and group posts, only internal users can set `Visibility` to `InternalUsers`.

## Usage

Use this object to retrieve the current contents of the feed fields, such as type of feed or feed ID.

## SearchPromotionRule

Represents a promoted search term, which is one or more keywords that you associate with a Salesforce Knowledge article. When a user's search query includes these keywords, the associated article is returned first in search results. This object is available in API version 31.0 and later.

### Supported Calls

`create()`, `delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

### Special Access Rules

A user must have the "Manage Promoted Search Terms" permission.

### Fields

Field Name	Details
PromotedEntityId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort, Update</p> <p><b>Description</b> The ID of the KnowledgeArticleVersion that the promoted search term is associated with. The article must be in published status.</p>
Query	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The text of the promoted search term. Maximum length: 100 characters. You can associate the same promoted search term with multiple articles. If the user's search matches the promoted term, all associated articles are promoted in search results, ordered by relevancy. For best results, create promoted search terms selectively and limit the number of articles that are promoted per term.</p>

### Usage

Use this object to optimize article search results in Salesforce Knowledge.

### TopicAssignment


Represents the assignment of a topic to a specific feed item, record, or file. This object is available in API version 28.0 and later.

Administrators must enable topics for objects before users can add topics to records of that object type. Topics for most objects are available in API version 30.0 and later. Topics for ContentDocument are available in API version 37.0 and later.

## Supported Calls

`create()`, `describeSObjects()`, `query()`, `retrieve()`

## Fields

Field Name	Details
<code>EntityId</code>	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> Identifier of the feed item, record, or file.</p>
<code>EntityKeyPrefix</code>	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, idLookup, Sort</p> <p><b>Description</b> The first three digits of the <code>EntityID</code> field, which identify the object type (account, opportunity, etc). This read-only field is available in API version 32.0 and later.  Interface label is "Record Key Prefix," which appears only in reports.</p>
<code>EntityType</code>	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The standard name for the object type (account, opportunity, etc). This read-only field is available in API version 33.0 and later.  Interface label is "Object Type," which appears only in reports.</p> <p> <b>Tip:</b> In most cases, you should use this field rather than <code>EntityKeyPrefix</code>, which exists primarily to support older reports.</p>
<code>NetworkId</code>	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p>

Field Name	Details
	<p><b>Description</b></p> <p>Identifier of the community to which the TopicAssignment belongs. This field is available only if Salesforce Communities is enabled in your organization.</p>
TopicId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Sort</p> <p><b>Description</b></p> <p>Identifier of the topic.</p>

## Usage

Use this object to query the assignments of topics to feed items, records, or files. To assign or remove topics, you must have the “Assign Topics” permission.

In SOQL `SELECT` syntax, this object supports nested semi-joins, allowing queries on Knowledge articles assigned to specific topics. For example:

```
SELECT parentId FROM KnowledgeArticleViewStat
  WHERE parentId in (SELECT KnowledgeArticleId FROM KnowledgeArticleVersion
  WHERE publishStatus = 'Online' AND language = 'en_US'
  AND Id in (select EntityId from TopicAssignment where TopicId ='0T0xx0000000xxx'))
```

No SOQL limit if logged-in user has “View All Data” permission. If not, do one of the following:

- Specify a `LIMIT` clause of 1,100 records or fewer.
- Filter on `Id` or `Entity` when using a `WHERE` clause with “=”.

## UserProfileFeed

Represents a user profile feed, which tracks all actions by a user on records that can be tracked in a feed. This feed is displayed on the user profile page.

UserProfileFeed is available in API version 18.0 through API 26.0. In API version 27.0 and later, UserProfileFeed is no longer available in the SOAP API. Use the Chatter REST API to access UserProfileFeed.

## Supported Calls

`delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

## Special Access Rules


You can delete all feed items you created. To delete feed items you didn’t create, you must have one of these permissions:

- “Modify All Data”
- “Moderate Chatter”

 **Note:** Users with the “Moderate Chatter” permission can delete only the feed items and comments they see.

- “Manage Unlisted Groups”  
Only users with this permission can delete items in unlisted groups.

## Fields

Field	Details
Body	<p><b>Type</b> textarea</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The content of UserProfileFeed. Required when <code>Type</code> is <code>TextPost</code>. Optional when <code>Type</code> is <code>ContentPost</code> or <code>LinkPost</code>. This field is the message that appears in the feed.</p>
CommentCount	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The number of FeedComments associated with this feed item.</p> <p> <b>Tip:</b> In a feed that supports pre-moderation, <code>CommentCount</code> isn't updated until a comment is published. For example, say you comment on a post that already has one published comment and your comment triggers moderation. Now there are two comments on the post, but the count says there's only one. In a moderated feed, your comment isn't counted until it's approved by an admin or a person with “CanApproveFeedPostAndComment” or “ModifyAllData” permission.</p> <p>This has implications for how you retrieve feed comments. In a moderated feed, rather than retrieving comments by looping through <code>CommentCount</code>, go through pagination until end of comments is returned.</p>
ContentData	<p><b>Type</b> base64</p> <p><b>Properties</b> Nillable</p> <p><b>Description</b> Required if <code>Type</code> is <code>ContentPost</code>. Encoded file data in any format, and can't be 0 bytes. Setting this field automatically sets <code>Type</code> to <code>ContentPost</code>.</p>
ContentDescription	<p><b>Type</b> textarea</p>


Field	Details
	<p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The description of the file specified in <code>ContentData</code>.</p>
<code>ContentFileName</code>	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> This field is required if <code>Type</code> is <code>ContentPost</code>. The name of the file uploaded to the feed. Setting <code>ContentFileName</code> automatically sets <code>Type</code> to <code>ContentPost</code>.</p>
<code>ContentSize</code>	<p><b>Type</b> int</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The size of the file (in bytes) uploaded to the feed. This field is read-only and is automatically determined during insert.</p>
<code>ContentType</code>	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> Available in API version 36.0 and earlier only. The MIME type of the file uploaded to the feed. This field is read-only and is automatically determined during insert.</p>
<code>CreatedDate</code>	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Date and time when this record was created. This field is a standard system field. Ordering by <code>CreatedDateDESC</code> sorts the feed by the most recent feed item.</p>
<code>FeedPostId</code>	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p>

Field	Details
	<p><b>Description</b></p> <p>This field was removed in API version 22.0, and is available in earlier versions for backward compatibility only.</p> <p>ID of the associated FeedPost. A FeedPost represents the following types of changes in an FeedItem: changes to tracked fields, text posts, link posts, and content posts.</p>
InsertedById	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Group, Nillable, Sort</p> <p><b>Description</b></p> <p>ID of the user who added this object to the feed. For example, if an application migrates posts and comments from another application into a feed, the <code>InsertedBy</code> value is set to the ID of the logged-in user.</p>
IsDeleted	<p><b>Type</b></p> <p>boolean</p> <p><b>Properties</b></p> <p>Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b></p> <p>Standard system field. Indicates whether the record has been moved to the Recycle Bin (<code>true</code>) or not (<code>false</code>).</p>
LastModifiedDate	<p><b>Type</b></p> <p>dateTime</p> <p><b>Properties</b></p> <p>Defaulted on create, Filter, Sort</p> <p><b>Description</b></p> <p>Date and time when a user last modified this record. This field is a standard system field.</p> <p>When a feed item is created, <code>LastModifiedDate</code> is the same as <code>CreatedDate</code>. If a <code>FeedComment</code> is inserted on that feed item, then <code>LastModifiedDate</code> becomes the <code>CreatedDate</code> for that <code>FeedComment</code>. Deleting the <code>FeedComment</code> does not change the <code>LastModifiedDate</code>.</p> <p>Ordering by <code>LastModifiedDate DESC</code> sorts the feed by both the most recent feed item or comment.</p>
LikeCount	<p><b>Type</b></p> <p>int</p> <p><b>Properties</b></p> <p>Filter, Group, Sort</p> <p><b>Description</b></p> <p>The number of <code>FeedLikes</code> associated with this feed item.</p>

Field	Details
LinkUrl	<p><b>Type</b> url</p> <p><b>Properties</b> Nillable, Sort</p> <p><b>Description</b> The URL of a <code>LinkPost</code>.</p>
NetworkScope	<p><b>Type</b> picklist</p> <p><b>Properties</b> Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> Specifies whether this feed item is available in the default community, a specific community, or all communities. This field is available in API version 26.0 and later, if Salesforce Communities is enabled for your organization.</p> <p><code>NetworkScope</code> can have the following values:</p> <ul style="list-style-type: none"> <li>• <code>NetworkId</code>—The ID of the community in which the <code>FeedItem</code> is available. If left empty, the feed item is only available in the default community.</li> <li>• <code>AllNetworks</code>—The feed item is available in all communities.</li> </ul> <p>Note the following exceptions for <code>NetworkScope</code>:</p> <ul style="list-style-type: none"> <li>• Only feed items with a <code>Group</code> or <code>User</code> parent can set a <code>NetworkId</code> or a null value for <code>NetworkScope</code>.</li> <li>• For feed items with a record parent, users can set <code>NetworkScope</code> only to <code>AllNetworks</code>.</li> <li>• You can't filter a <code>FeedItem</code> on the <code>NetworkScope</code> field.</li> </ul>
ParentId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> ID of the account record that is tracked in the feed. The feed is displayed on the detail page for this record.</p>
RelatedRecordId	<p><b>Type</b> reference</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> ID of the <code>ContentVersion</code> object associated with a <code>ContentPost</code>. This field is null for all posts except <code>ContentPost</code>.</p>



Field	Details
Title	<p><b>Type</b> string</p> <p><b>Properties</b> Group, Nillable, Sort</p> <p><b>Description</b> The title of the user profile record. When the <code>Type</code> is <code>LinkPost</code>, the <code>LinkUrl</code> is the URL and this field is the link name.</p>
Type	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> The type of feed item:</p> <ul style="list-style-type: none"> <li>• <code>ActivityEvent</code>—indirectly generated event when a user or the API adds a Task associated with a feed-enabled parent record (excluding email tasks on cases). Also occurs when a user or the API adds or updates a Task or Event associated with a case record (excluding email and call logging). For a recurring Task with <code>CaseFeed</code> disabled, one event is generated for the series only. For a recurring Task with <code>CaseFeed</code> enabled, events are generated for the series and each occurrence.</li> <li>• <code>AdvancedTextPost</code>—created when a user posts a group announcement.</li> <li>• <code>AnnouncementPost</code>—Not used.</li> <li>• <code>ApprovalPost</code>—generated when a user submits an approval.</li> <li>• <code>BasicTemplateFeedItem</code>—Not used.</li> <li>• <code>CanvasPost</code>—a post made by a canvas app posted on a feed.</li> <li>• <code>CollaborationGroupCreated</code>—generated when a user creates a public group.</li> <li>• <code>CollaborationGroupUnarchived</code>—Not used.</li> <li>• <code>ContentPost</code>—a post with an attached file.</li> <li>• <code>CreatedRecordEvent</code>—generated when a user creates a record from the publisher.</li> <li>• <code>DashboardComponentAlert</code>—generated when a dashboard metric or gauge exceeds a user-defined threshold.</li> <li>• <code>DashboardComponentSnapshot</code>—created when a user posts a dashboard snapshot on a feed.</li> <li>• <code>LinkPost</code>—a post with an attached URL.</li> <li>• <code>PollPost</code>—a poll posted on a feed.</li> <li>• <code>ProfileSkillPost</code>—generated when a skill is added to a user’s Chatter profile.</li> <li>• <code>QuestionPost</code>—generated when a user posts a question.</li> <li>• <code>ReplyPost</code>—generated when Chatter Answers posts a reply.</li> <li>• <code>RypplePost</code>—generated when a user creates a Thanks badge in Work.com.</li> </ul>

Field	Details
	<ul style="list-style-type: none"> <li>• <code>TextPost</code>—a direct text entry on a feed.</li> <li>• <code>TrackedChange</code>—a change or group of changes to a tracked field.</li> <li>• <code>UserStatus</code>—automatically generated when a user adds a post. Deprecated.</li> </ul> <p>The following values appear in the <code>Type</code> picklist for all feed objects but apply only to <code>CaseFeed</code>:</p> <ul style="list-style-type: none"> <li>• <code>CaseCommentPost</code>—generated event when a user adds a case comment for a case object</li> <li>• <code>EmailMessageEvent</code>—generated event when an email related to a case object is sent or received</li> <li>• <code>CallLogPost</code>—generated event when a user logs a call for a case through the user interface. CTI calls also generate this event.</li> <li>• <code>ChangeStatusPost</code>—generated event when a user changes the status of a case</li> <li>• <code>AttachArticleEvent</code>—generated event when a user attaches an article to a case</li> </ul> <p> <b>Note:</b> If you set <code>Type</code> to <code>ContentPost</code>, also specify <code>ContentData</code> and <code>ContentFileName</code>.</p>

<p><code>Visibility</code></p>	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Restricted picklist, Sort</p> <p><b>Description</b> Specifies whether this feed item is available to all users or internal users only. This field is available in API version 26.0 and later, if Salesforce Communities is enabled for your organization.</p> <p><code>Visibility</code> can have the following values:</p> <ul style="list-style-type: none"> <li>• <code>AllUsers</code>—The feed item is available to all users who have permission to see the feed item.</li> <li>• <code>InternalUsers</code>—The feed item is available to internal users only.</li> </ul> <p>Note the following exceptions for <code>Visibility</code>:</p> <ul style="list-style-type: none"> <li>• For record posts, <code>Visibility</code> is set to <code>InternalUsers</code> for all internal users by default.</li> <li>• External users can set <code>Visibility</code> only to <code>AllUsers</code>.</li> <li>• On user and group posts, only internal users can set <code>Visibility</code> to <code>InternalUsers</code>.</li> </ul>
--------------------------------	--

## Usage


Use this object to query and retrieve record changes tracked in a user profile feed. Note the following when working with user profile feeds:

- This object is read only in the API.

- Queries retrieve feed items that include mentions from other users.
- Use this object to query and retrieve lead feed items that were associated with a converted lead record.
- Include a `WITH` clause and specify the `UserId` of the user whose profile you want to query. The `WITH` clause must come after a `WHERE` clause.

Users that do not have the “View All Data” permission have the following limitations when querying records:

- Must specify a `LIMIT` clause and the limit must be less than or equal to 1000.
- Can include a `WHERE` clause that references `UserProfileFeed` fields, but cannot include references to fields in related objects. For example, you can filter by `CreatedDate` or `ParentId`, but not by `Parent.Name`.
- Can include an `ORDER BY` clause that references `UserProfileFeed` fields, but cannot include references to fields in related objects. For example, you can `ORDER BY CreatedDate` or `ParentId`, but not by `Parent.Name`.

 **Tip:** To query for the most recent feed items, you should `ORDER BY CreatedDate DESC, Id DESC`.

Note the following SOQL restrictions.

- No SOQL limit if logged-in user has “View All Data” permission. If not, specify a `LIMIT` clause of 1,000 records or fewer.
- SOQL `ORDER BY` on fields using relationships is not available. Use `ORDER BY` on fields on the root object in the SOQL query.

## Knowledge SOAP API Calls

---

SOAP API calls for working with Salesforce Knowledge.

IN THIS SECTION:

[describeKnowledge\(\)](#) Call Name

Retrieves the Knowledge language settings in the organization.

[describeDataCategoryGroups\(\)](#)

Retrieves available category groups for objects specified in the request.

[describeDataCategoryGroupStructures\(\)](#)

Retrieves available category groups along with their data category structure for objects specified in the request.

[search\(\)](#)

Executes a text search in your organization’s data.

### **describeKnowledge ()** Call Name

Retrieves the Knowledge language settings in the organization.

### Syntax

```
KnowledgeSettings result = _connection.describeKnowledgeSettings();
```

### Usage

Use this call to describe the existing Knowledge language settings, including the default Knowledge language, supported languages, and a list of Knowledge language information. You can also use `KnowledgeSettings` in the Metadata API to obtain similar information.

## Sample Code—Java

This sample shows how to retrieve the Knowledge language settings. It returns the default Knowledge language, a list of Knowledge supported language, including the language code and whether is an active Knowledge language.

```
public void describeKnowledgeSettingsSample() {
    try {

        // Make the describe call for KnowledgeSettings
        KnowledgeSettings result = connection.describeKnowledgeSettings();

        // Get the properties of KnowledgeSettings
        System.out.println("Knowledge default language: " + result.getDefaultLanguage());
        for (KnowledgeLanguageItem lang : result.getLanguages()) {
            System.out.println("Language: " + lang.getName());
            System.out.println("Active: " + lang.isActive());
        }
    } catch (ConnectionException ex) {
        ex.printStackTrace();
    }
}
```

## Sample Code—C#

This sample shows how to retrieve the Knowledge language settings. It returns the default Knowledge language, a list of Knowledge supported language, including the language code and whether is an active Knowledge language.

```
public void describeKnowledgeSettingsSample() {
    try {

        // Make the describe call for KnowledgeSettings
        KnowledgeSettings result = connection.describeKnowledgeSettings();

        // Get the properties of KnowledgeSettings
        Console.WriteLine("Knowledge default language: " + result.getDefaultLanguage());
        for (KnowledgeLanguageItem lang : result.getLanguages()) {
            Console.WriteLine("Language: " + lang.GetName());
            Console.WriteLine("Active: " + lang.IsActive());
        }
    } catch (SoapException ex) {
        ex.printStackTrace();
    }
}
```

## Response

KnowledgeSettings

## **describeDataCategoryGroups ()**

Retrieves available category groups for objects specified in the request.

## Syntax

```
DescribeDataCategoryGroupResult[] = connection.describeDataCategoryGroups() (string[]
sObjectTypes);
```

## Usage

Use this call to describe the available category groups for the objects specified in the request. This call can be used with the [describeDataCategoryGroupStructures\(\)](#) call to describe all the categories available for a specific object. For additional information about data categories, see “Data Categories in Salesforce.com” in the Salesforce online help.

## Sample Code—Java

This sample shows how to retrieve the data category groups associated with:

- Salesforce Knowledge articles
- Questions from the Answers feature

It returns the name, label and description of a category group and the name of the associated `sObject` (article or question). It also returns the number of data categories in the data category group.

```
public void describeDataCategoryGroupsSample() {
    try {
        // Make the describe call for data category groups
        DescribeDataCategoryGroupResult[] results =
            connection.describeDataCategoryGroups(new String[] {
                "KnowledgeArticleVersion", "Question"});

        // Get the properties of each data category group
        for (int i = 0; i < results.length; i++) {
            System.out.println("sObject: " +
                results[i].getSObject());
            System.out.println("Group name: " +
                results[i].getName());
            System.out.println("Group label: " +
                results[i].getLabel());
            System.out.println("Group description: " +
                (results[i].getDescription()==null? "" :
                results[i].getDescription()));
            System.out.println("Number of categories: " +
                results[i].getCategoryCount());
        }
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

## Sample Code—C#

This sample shows how to retrieve the data category groups associated with:

- Salesforce Knowledge articles
- Questions from the Answers feature

It returns the name, label and description of a category group and the name of the associated `subject` (article or question). It also returns the number of data categories in the data category group.

```
public void describeDataCategoryGroups() {
    try {
        // Make the describe call for data category groups
        DescribeDataCategoryGroupResult[] results =
            binding.describeDataCategoryGroups(new String[] {
                "KnowledgeArticleVersion", "Question"});

        // Get the properties of each data category group
        for (int i = 0; i < results.Length; i++) {
            Console.WriteLine("sObject: " +
                results[i].subject);
            Console.WriteLine("Group name: " +
                results[i].name);
            Console.WriteLine("Group label: " +
                results[i].label);
            Console.WriteLine("Group description: " +
                (results[i].description==null? "" :
                results[i].description));
            Console.WriteLine("Number of categories: "+
                results[i].categoryCount);
        }
    } catch (SoapException e) {
        Console.WriteLine("An unexpected error has occurred: " +
            e.Message + "\n" + e.StackTrace);
    }
}
```

## Arguments

Name	Type	Description
<code>sObjectTypes</code>	<code>string[]</code>	<p>The specified value can be:</p> <ul style="list-style-type: none"> <li><code>KnowledgeArticleVersion</code>—to retrieve category groups associated with article types.</li> <li><code>Question</code>—to retrieve category groups associated with questions.</li> </ul> <p>For additional information about articles and questions, see “Work with Articles and Translations” and “Answers Overview” in the Salesforce online help.</p>

## Response

[DescribeDataCategoryGroupResult](#)

## Faults

`InvalidSObjectFault`

`UnexpectedErrorFault`

**IN THIS SECTION:**[DescribeDataCategoryGroupResult](#)

The `describeDataCategoryGroups()` call returns a `DescribeDataCategoryGroupResult` object containing the list of the category groups associated with the specified objects.

**DescribeDataCategoryGroupResult**

The `describeDataCategoryGroups()` call returns a `DescribeDataCategoryGroupResult` object containing the list of the category groups associated with the specified objects.

Name	Type	Description
categoryCount	int	The number of visible data categories in the data category group.
description	string	The description of the data category group.
label	string	Label for the data category group in the Salesforce user interface.
name	string	The unique name used for API access to the data category group .
subject	string	The object associated with the data category group.

**describeDataCategoryGroupStructures ()**

Retrieves available category groups along with their data category structure for objects specified in the request.

**Syntax**

```
describeDataCategoryGroupStructures() [] = connection.
    describeDataCategoryGroupStructures() (DataCategoryGroupSObjectTypePair[]
    pairs, boolean topCategoriesOnly)
```

**Usage**

Use this call to return the visible data category structure for the given object category group pairs. First use [describeDataCategoryGroups \(\)](#) to find the available category groups for the objects specified. From the returned list, choose the object category group pairs to pass as the input in `describeDataCategoryGroupStructures ()`. This call returns all the visible categories and data category structure as output. For additional information about data categories and data category visibility, see “Data Categories in Salesforce.com” and “Data Category Visibility” in the Salesforce online help.

**Sample Code—Java**

This sample shows how to use `sObject` and data category group pairs to retrieve data categories for each pair. It calls `describeDataCategoryGroupStructures ()` with two pairs, `KnowledgeArticleVersion/Regions` and `Question/Regions`, and iterates through the results of this call. It gets the top categories for each result, which is “All”, and then gets the first-level child categories. The sample requires that you set up a data category group called `Regions` with some child categories and associate it

with a knowledge article and questions. Alternatively, you can replace the data category group name in the sample if you want to use an existing data category group in your org that has a different name.

```
public void describeDataCategoryGroupStructuresSample() {
    try {
        // Create the data category pairs
        DataCategoryGroupSubjectTypePair pair1 =
            new DataCategoryGroupSubjectTypePair();
        DataCategoryGroupSubjectTypePair pair2 =
            new DataCategoryGroupSubjectTypePair();
        pair1.setSubject("KnowledgeArticleVersion");
        pair1.setDataCategoryGroupName("Regions");
        pair2.setSubject("Question");
        pair2.setDataCategoryGroupName("Regions");

        DataCategoryGroupSubjectTypePair[] pairs =
            new DataCategoryGroupSubjectTypePair[] {
                pair1,
                pair2
            };

        // Get the list of top level categories using the describe call
        DescribeDataCategoryGroupStructureResult[] results =
            connection.describeDataCategoryGroupStructures(
                pairs,
                false
            );

        // Iterate through each result and get some properties
        // including top categories and child categories
        for (int i = 0; i < results.length; i++) {
            DescribeDataCategoryGroupStructureResult result =
                results[i];
            String sObject = result.getSubject();
            System.out.println("sObject: " + sObject);
            System.out.println("Group name: " + result.getName());
            System.out.println("Group label: " + result.getLabel());
            System.out.println("Group description: " +
                result.getDescription());

            // Get the top-level categories
            DataCategory[] topCategories = result.getTopCategories();

            // Iterate through the top level categories and retrieve
            // some information
            for (int j = 0; j < topCategories.length; j++) {
                DataCategory topCategory = topCategories[j];
                System.out.println("Category name: " +
                    topCategory.getName());
                System.out.println("Category label: " +
                    topCategory.getLabel());
                DataCategory [] childCategories =
                    topCategory.getChildCategories();
                System.out.println("Child categories: ");
                for (int k = 0; k < childCategories.length; k++) {
```



```

System.out.println("\t" + k + ". Category name: " +
childCategories[k].getName());
System.out.println("\t" + k + ". Category label: " +
childCategories[k].getLabel());
}
}
}
} catch (ConnectionException ce) {
ce.printStackTrace();
}
}
}

```

## Sample Code—C#

This sample shows how to use `sObject` and data category group pairs to retrieve data categories for each pair. It calls `describeDataCategoryGroupStructures()` with two pairs, `KnowledgeArticleVersion/Regions` and `Question/Regions`, and iterates through the results of this call. It gets the top categories for each result, which is "All", and then gets the first-level child categories. The sample requires that you set up a data category group called *Regions* with some child categories and associate it with a knowledge article and questions. Alternatively, you can replace the data category group name in the sample if you want to use an existing data category group in your org that has a different name.

```

public void describeDataCategoryGroupStructuresSample() {
    try {
        // Create the data category pairs
        DataCategoryGroupSubjectTypePair pair1 =
        new DataCategoryGroupSubjectTypePair();
        DataCategoryGroupSubjectTypePair pair2 =
        new DataCategoryGroupSubjectTypePair();
        pair1.subject = "KnowledgeArticleVersion";
        //pair1.setDataCategoryGroupName("Regions");
        pair1.dataCategoryGroupName = "KBArticleCategories";
        pair2.subject = "Question";
        //pair2.setDataCategoryGroupName("Regions");
        pair2.dataCategoryGroupName = "KBArticleCategories";

        DataCategoryGroupSubjectTypePair[] pairs =
        new DataCategoryGroupSubjectTypePair[] {
            pair1,
            pair2
        };

        // Get the list of top level categories using the describe call
        DescribeDataCategoryGroupStructureResult[] results =
        binding.describeDataCategoryGroupStructures(
            pairs,
            false
        );

        // Iterate through each result and get some properties
        // including top categories and child categories
        for (int i = 0; i < results.Length; i++) {
            DescribeDataCategoryGroupStructureResult result =
            results[i];

```

```

String sObject = result.sobject;
Console.WriteLine("sObject: " + sObject);
Console.WriteLine("Group name: " + result.name);
Console.WriteLine("Group label: " + result.label);
Console.WriteLine("Group description: " +
result.description);

// Get the top-level categories
DataCategory[] topCategories = result.topCategories;

// Iterate through the top level categories and retrieve
// some information
for (int j = 0; j < topCategories.Length; j++) {
DataCategory topCategory = topCategories[j];
Console.WriteLine("Category name: " +
topCategory.name);
Console.WriteLine("Category label: " +
topCategory.label);
DataCategory [] childCategories =
topCategory.childCategories;
Console.WriteLine("Child categories: ");
for (int k = 0; k < childCategories.Length; k++) {
Console.WriteLine("\t" + k + ". Category name: " +
childCategories[k].name);
Console.WriteLine("\t" + k + ". Category label: " +
childCategories[k].label);
}
}
}
}
}
catch (SoapException e)
{
Console.WriteLine("An unexpected error has occurred: " +
e.Message + "\n" + e.StackTrace);
}
}

```

## Arguments

Name	Type	Description
pairs	<a href="#">DataCategoryGroupSObjectTypePair[]</a>	Specifies a category group and an object to query. Visible data categories are retrieved for that object.
topCategoriesOnly	boolean	Indicates whether the call returns only the top ( <code>true</code> ) or all the categories ( <code>false</code> ) visible depending on the user's data category group visibility settings. For more information on data category group visibility, see <a href="#">Data Category Visibility</a> in the Salesforce online help.

[DataCategoryGroupSObjectTypePair](#) contains the following fields:

Name	Type	Description
dataCategoryGroupName	string	The unique name used for API access to the data category group.
subject	string	The object associated with the data category group

## Response

[describeDataCategoryGroupStructures\(\)](#) []

## Faults

InvalidObjectFault

UnexpectedErrorFault

### IN THIS SECTION:

[describeDataCategoryGroupStructures\(\)](#)

The describeDataCategoryGroupStructures() call returns an array of DescribeDataCategoryGroupStructureResult objects containing the category groups and categories associated with the specified objects.

## describeDataCategoryGroupStructures ()

The describeDataCategoryGroupStructures() call returns an array of DescribeDataCategoryGroupStructureResult objects containing the category groups and categories associated with the specified objects.

Name	Type	Description
description	string	The description of the data category group.
label	string	The label for the data category group in the Salesforce user interface.
name	string	The unique name used for API access to the data category group.
subject	string	The object associated with the data category group.
topCategories	<a href="#">DataCategory</a> []	A list of top level categories visible depending on the user's data category group visibility settings. For more information on data category group visibility, see "Data Category Visibility" in the Salesforce online help.

## DataCategory

Name	Type	Description
childDataCategories	<a href="#">DataCategory</a> []	A recursive list of visible sub categories in the data category.
label	string	The label for the data category in the Salesforce user interface.
name	string	The unique name used for API access to the data category.

## search ()

Executes a text search in your organization's data.

## Syntax

```
SearchResult = connection.search(String searchString);
```

## Usage

Use `search ()` to search for records based on a search string. The `search ()` call supports searching custom objects. For an extensive discussion about the syntax and rules used for text searches, see the [Salesforce SOQL and SOSL Reference Guide](#).

Certain objects cannot be searched via the API, such as Attachment objects. To search an object via the `search ()` call, the object must be configured as searchable (`isSearchable` is `true`). To determine whether an object can be searched, your client application can invoke the `describeSObjects ()` call on the object and inspect its `searchable` property.

## Sample Code—Java

This sample makes the `search ()` call by passing it a SOSL query, which returns contacts, leads, and accounts whose phone fields contain a specified value. Next, it gets the `sObject` records from the results and stores the records in arrays depending on the record type. Finally, it writes the fields of the returned contacts, leads, and accounts to the console.

```
public void searchSample() {
    try {
        // Perform the search using the SOSL query.
        SearchResult sr = connection.search(
            "FIND {4159017000} IN Phone FIELDS RETURNING "
            + "Contact(Id, Phone, FirstName, LastName), "
            + "Lead(Id, Phone, FirstName, LastName), "
            + "Account(Id, Phone, Name)");

        // Get the records from the search results.
        SearchRecord[] records = sr.getSearchRecords();

        ArrayList<Contact> contacts = new ArrayList<Contact>();
        ArrayList<Lead> leads = new ArrayList<Lead>();
        ArrayList<Account> accounts = new ArrayList<Account>();

        // For each record returned, find out if it's a
        // contact, lead, or account and add it to the
        // appropriate array, then write the records
        // to the console.
        if (records.length > 0) {
            for (int i = 0; i < records.length; i++) {
                SObject record = records[i].getRecord();
                if (record instanceof Contact) {
                    contacts.add((Contact) record);
                } else if (record instanceof Lead) {
                    leads.add((Lead) record);
                } else if (record instanceof Account) {
                    accounts.add((Account) record);
                }
            }
        }
    }
}
```

```

    }
    }

    System.out.println("Found " + contacts.size() + " contacts.");
    for (Contact c : contacts) {
        System.out.println(c.getId() + ", " + c.getFirstName() + ", "
            + c.getLastName() + ", " + c.getPhone());
    }
    System.out.println("Found " + leads.size() + " leads.");
    for (Lead d : leads) {
        System.out.println(d.getId() + ", " + d.getFirstName() + ", "
            + d.getLastName() + ", " + d.getPhone());
    }
    System.out.println("Found " + accounts.size() + " accounts.");
    for (Account a : accounts) {
        System.out.println(a.getId() + ", " + a.getName() + ", "
            + a.getPhone());
    }
    } else {
        System.out.println("No records were found for the search.");
    }
    } catch (Exception ce) {
        ce.printStackTrace();
    }
    }
}

```

## Sample Code—C#

This sample makes the `search()` call by passing it a SOSL query, which returns contacts, leads, and accounts whose phone fields contain a specified value. Next, it gets the `sObject` records from the results and stores the records in arrays depending on the record type. Finally, it writes the fields of the returned contacts, leads, and accounts to the console.

```

public void searchSample()
{
    try
    {
        // Perform the search using the SOSL query.
        SearchResult sr = binding.search(
            "FIND {4159017000} IN Phone FIELDS RETURNING "
            + "Contact(Id, Phone, FirstName, LastName), "
            + "Lead(Id, Phone, FirstName, LastName), "
            + "Account(Id, Phone, Name)");

        // Get the records from the search results.
        SearchRecord[] records = sr.searchRecords;

        List<Contact> contacts = new List<Contact>();
        List<Lead> leads = new List<Lead>();
        List<Account> accounts = new List<Account>();

        // For each record returned, find out if it's a
        // contact, lead, or account and add it to the
        // appropriate array, then write the records
        // to the console.
    }
}

```

```
if (records.Length > 0)
{
for (int i = 0; i < records.Length; i++)
{
sObject record = records[i].record;
if (record is Contact)
{
contacts.Add((Contact)record);
}
else if (record is Lead)
{
leads.Add((Lead)record);
}
else if (record is Account)
{
accounts.Add((Account)record);
}
}

Console.WriteLine("Found " + contacts.Count + " contacts.");
foreach (Contact c in contacts)
{
Console.WriteLine(c.Id + ", " +
c.FirstName + ", " +
c.LastName + ", " +
c.Phone);
}
Console.WriteLine("Found " + leads.Count + " leads.");
foreach (Lead d in leads)
{
Console.WriteLine(d.Id + ", " +
d.FirstName + ", " +
d.LastName + ", " +
d.Phone);
}
Console.WriteLine("Found " + accounts.Count + " accounts.");
foreach (Account a in accounts)
{
Console.WriteLine(a.Id + ", " +
a.Name + ", " +
a.Phone);
}
}
else
{
Console.WriteLine("No records were found for the search.");
}
}
catch (SoapException e)
{
Console.WriteLine("An unexpected error has occurred: " +
e.Message + "\n" + e.StackTrace);
}
}
```

## Arguments

Name	Type	Description
search	string	Search string that specifies the text expression to search for, the scope of fields to search, the list of objects and fields to retrieve, and the maximum number of records to return. For more information, see the <a href="#">Salesforce SOQL and SOSL Reference Guide</a> .

## Response

[SearchResult](#)

## Fault

InvalidFieldFault

InvalidSObjectFault

MalformedSearchFault

UnexpectedErrorFault

### IN THIS SECTION:

[SearchResult](#)

The describeDataCategoryGroupStructures() call returns a SearchResult object, which has the following properties.

## SearchResult

The describeDataCategoryGroupStructures() call returns a SearchResult object, which has the following properties.

Name	Type	Description
searchRecords	SearchRecord[]	Array of SearchRecord objects, each of which contains an sObject.
searchResultsMetadata	SearchResultsMetadata	Metadata for searchRecords.
queryId	string	Unique identifier for the SOSL search.

# CHAPTER 3 Salesforce Knowledge REST APIs

## In this chapter ...

- [Manage Knowledge with REST APIs](#)
- [Support Knowledge with REST API](#)

You can manage articles and get support information with REST APIs.

REST API uses the same underlying data model and standard objects as those in SOAP API. For generic REST API information see the [Force.com REST API Developer Guide](#).



# Manage Knowledge with REST APIs

---

REST APIs provide programmatic access to many actions you can perform on your Knowledge base, an article, and its translations.

## IN THIS SECTION:

### [Archive the Master Version of an Article](#)

Archives the master version of an article. The actions are defined by the field change you request on the resource. To archive the masterversion, use `"publishStatus": "Archived"`. To schedule a date for archiving, use `"archiveScheduleDate": <date>`.

### [Assign a Task Related to a Master Article](#)

Assigns a task to a user for a master article, including due date and instructions. The actions are defined by the field change you request on the resource.

### [Assign a Task Related to a Translation](#)

Assigns a task to a user for a translated article, including due date and instructions. The actions are defined by the field change you request on the resource.

### [Delete a Master Version of an Article](#)

Deletes the master version of an article.

### [Delete a Translated Version of an Article](#)

Deletes a translated version of an article.

### [Edit an Online Version of a Master Article](#)

Creates a draft copy of the online version of a master article. This does not unpublish the online version.

### [Publish the Master Version of an Article](#)

Publishes the master version of an article.

### [Restore an Archived Version of an Article](#)

Restores an archived version of the article. If `versionNumber` isn't specified, restores the latest version of the archived article.

### [Retrieve Article Metadata](#)

Retrieves the metadata of an article.

### [Retrieve Knowledge Language Settings](#)

Returns the existing Knowledge language settings, including the default knowledge language and a list of supported Knowledge language information.

### [Retrieve a Version of an Article](#)

Retrieves the version ID of an article.

### [Search for Metadata Elements of a Master Version](#)

Searches for metadata elements of the online master version of an article.

### [Search for Metadata Elements of a Translated Version](#)

Searches for metadata elements of a translated version of an article.

### [Set a Translated Article Version to Complete](#)

Sets a translated article version to complete.

### [Set a Translated Article Version to Incomplete](#)

Sets a translated article version to incomplete.

### [Submit an Article for Translation](#)

Submits an article for translation and keeps the online version. Copies or creates a new draft translation if there is already a translation in the specified language. The publishing actions are defined by the field change you request on the resource.

### [Unpublish the Master Version of an Article](#)

Unpublishes the online master version of an article when there isn't an existing draft article.

### [Unpublish the Online Version of a Translated Article](#)

Unpublishes the online version of a translated article.

### [Parameterized Search](#)

Executes a simple RESTful search using parameters instead of a SOSL clause. Indicate parameters in a URL in the GET method. Or, use POST for more complex JSON searches.

### [Search](#)

Executes the specified SOSL search. The search string must be URL-encoded.

### [Search Scope and Order](#)

Returns an ordered list of objects in the default global search scope of a logged-in user. Global search keeps track of which objects the user interacts with and how often and arranges the search results accordingly. Objects used most frequently appear at the top of the list.

### [Search Result Layouts](#)

Returns search result layout information for the objects in the query string. For each object, this call returns the list of fields displayed on the search results page as columns, the number of rows displayed on the first page, and the label used on the search results page.

### [Search Suggested Records](#)

Returns a list of suggested records whose names match the user's search string. The suggestions resource provides a shortcut for users to navigate directly to likely relevant records, before performing a full search.

### [Search Suggested Article Title Matches](#)

Returns a list of Salesforce Knowledge article titles that match the user's search query string. Provides a shortcut to navigate directly to likely relevant articles before the user performs a search.

### [Search Suggested Queries](#)

Returns a list of suggested searches based on the user's query string text matching searches that other users have performed in Salesforce Knowledge. Provides a way to improve search effectiveness, before the user performs a search.

## Archive the Master Version of an Article

Archives the master version of an article. The actions are defined by the field change you request on the resource. To archive the master version, use "publishStatus": "Archived". To schedule a date for archiving, use "archiveScheduleDate" : <date>.

## Syntax

### URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionID>
```

### Formats

JSON, XML

**HTTP Method**

PATCH

**Authentication**Authorization: OAuth *access\_token***Parameters**

Parameter	Description
publishStatus	Publishing status of the article. Use <code>archived</code> .
archiveScheduleDate	Date to archive the article.

**Input:**

Archive:

```
{
  "publishStatus": "Archived"
}
```

Schedule for archiving (using GMT date format):

```
{
  "archiveScheduleDate" : "2012-04-19T07:00:00.000+0000"
}
```

**Output:**

HTTP status code 204 is returned when an existing record is updated.

## Assign a Task Related to a Master Article

Assigns a task to a user for a master article, including due date and instructions. The actions are defined by the field change you request on the resource.

### Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionID>
```

**Formats**

JSON, XML

**HTTP Method**

PATCH

**Authentication**Authorization: OAuth *access\_token***Parameters**

Parameter	Description
assigneeId	Assigns the master article to a user ID.

Parameter	Description
dueDate	Date that the task is due.
instruction	Instructions for the task.

**Input:**

```
{
  "assigneeId": "05Dxx0000dsads"
  "dueDate": "2012-04-19T07:00:00.000+0000"
  "instruction": "Please review."
}
```

**Output:**

HTTP status code 204 is returned when an existing record is updated.

## Assign a Task Related to a Translation

Assigns a task to a user for a translated article, including due date and instructions. The actions are defined by the field change you request on the resource.

### Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<translationVersionId>
```

**Formats**

JSON, XML

**HTTP Method**

PATCH

**Authentication**

Authorization: OAuth *access\_token*

**Parameters**

Parameter	Description
assigneeId	Assigns the master article to a user ID.
dueDate	Date that the task is due.
instruction	Instructions for the task.

**Input:**

```
{
  "assigneeId": "05Dxx0000dsads"
  "dueDate": "2012-04-19T07:00:00.000+0000"
```

```
"instruction": "Please review."  
}
```

**Output:**

HTTP status code 204 is returned when an existing record is updated.

## Delete a Master Version of an Article

Deletes the master version of an article.

### Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionID>
```

**Formats**

JSON, XML

**HTTP Method**

DELETE

**Authentication**

Authorization: OAuth *access\_token*

**Parameters**

None

**Input:**

None required

**Output:**

HTTP status code 204 is returned when an existing record is deleted.

## Delete a Translated Version of an Article

Deletes a translated version of an article.

### Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<versionID>
```

**Formats**

JSON, XML

**HTTP Method**

DELETE

**Authentication**

Authorization: OAuth *access\_token*

**Parameters**

None

**Input**

None needed

**Output**

HTTP status code 204 is returned when an existing record is deleted.

## Edit an Online Version of a Master Article

Creates a draft copy of the online version of a master article. This does not unpublish the online version.

### Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions
```

**Formats**

JSON, XML

**HTTP Method**

POST

**Authentication**Authorization: OAuth *access\_token***Parameters**

Parameter	Description
articleId	The ID of the article.

**Input:**

```
{
  "articleId": <articleID>
}
```

## Publish the Master Version of an Article

Publishes the master version of an article.

The publishing actions are defined by the field change you request on the resource. To publish a minor version, use "publishStatus": "Online". To publish a major version, use "publishStatus": "Online" and "versionNumber": "NextVersion". To schedule a publication date, use "publishScheduleDate" : <date>.

### Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionId>
```

**Formats**

JSON, XML

**HTTP Method**

PATCH

**Authentication**Authorization: OAuth *access\_token***Parameters**

Parameter	Description
<code>publishStatus</code>	Publishing status of the article. Use <code>online</code> .
<code>versionNumber</code>	Version of the article.
<code>publishScheduleDate</code>	Date to publish the article.

**Input**

Publish a minor version:

```
{
  "publishStatus": "Online"
}
```

Publish a major version:

```
{
  "publishStatus": "Online"
  "versionNumber": "NextVersion"
}
```

Schedule for publication (using GMT date format):

```
{
  "publishScheduleDate" : "2012-05-19T07:00:00.000+0000"
}
```

**Output**

HTTP status code 204 is returned when an existing record is updated.

## Restore an Archived Version of an Article

Restores an archived version of the article. If `versionNumber` isn't specified, restores the latest version of the archived article.

### Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/
```

**Formats**

JSON, XML

**HTTP Method**

POST

**Authentication**

Authorization: OAuth *access\_token*

**Parameters**

Parameter	Description
articleId	The ID of the article.
versionNumber	Version of the article. If this field is not specified, the latest version of the archived article is restored.

**Input:**

```
{
  "articleId": "<articleID>",
  "versionNumber": <number>
}
```

## Retrieve Article Metadata

Retrieves the metadata of an article.

### Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articles/<articleId>
```

**Formats**

JSON, XML

**HTTP Method**

GET

**Authentication**

Authorization: OAuth *access\_token*

**Parameters**

None

**Input**

None required

## Retrieve Knowledge Language Settings

Returns the existing Knowledge language settings, including the default knowledge language and a list of supported Knowledge language information.



## Syntax

### URI

/services/data/v31.0/knowledge Management/settings

### Available since release

31.0

### Formats

JSON, XML

### HTTP methods

GET

### Authentication

Authorization: Bearer *token*

### Request body

None required

### Request parameters

None

## Example for Getting KnowledgeSettings

```
curl
https://https://yourInstance.salesforce.com/services/data/v31.0/knowledgeManagement/settings
-H "Authorization: Bearer token"
```

## Example JSON Response Body

```
{
  "defaultLanguage" : "en_US",
  "knowledgeEnabled" : true,
  "languages" : [ {
    "active" : true,
    "name" : "en_US"
  }, {
    "active" : true,
    "name" : "it"
  }, {
    "active" : true,
    "name" : "zh_CN"
  }, {
    "active" : true,
    "name" : "fr"
  } ]
}
```

## Usage

Salesforce Knowledge must be enabled in your organization. The user must have the “Knowledge User” license on their profile. This resource can be used in API version 31.0 and later. It retrieves the Knowledge language settings, including the default knowledge language and a list of supported Knowledge language information.

## Retrieve a Version of an Article

Retrieves the version ID of an article.

### Syntax

#### URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<articleVersionId>
```

#### Formats

JSON, XML

#### HTTP Method

GET

#### Authentication

Authorization: OAuth *access\_token*

#### Parameters

None

#### Input

None required

## Search for Metadata Elements of a Master Version

Searches for metadata elements of the online master version of an article.

### Syntax

#### URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions?  
filterArticleId=value1&FilterPublishStatus=value2
```

#### Formats

JSON, XML

#### HTTP Method

GET

#### Authentication

Authorization: OAuth *access\_token*

#### Example

This example searches for the online version of the master article 'kA0x50000000jsh':

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions?  
filterArticleId=kA0x50000000jsh&filterPublishStatus=online"
```

## Search for Metadata Elements of a Translated Version

Searches for metadata elements of a translated version of an article.

## Syntax

### URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations?filterArticleId=value1&filterLanguage=value2&FilterPublishStatus=value3";
```

### Formats

JSON, XML

### HTTP Method

GET

### Authentication

Authorization: OAuth *access\_token*

### Example

This example searches for the German online translation of the article 'kA0x5000000jsh':

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations?filterArticleId=kA0x5000000jsh&filterLanguage=de&filterPublishStatus=online"
```

## Set a Translated Article Version to Complete

Sets a translated article version to complete.

## Syntax

### URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<translationVersionID>
```

### Formats

JSON, XML

### HTTP Method

PATCH

### Authentication

Authorization: OAuth *access\_token*

### Parameters

Parameter	Description
complete	Set this value to <code>true</code> for translations that are complete.

### Input:

```
{
  "complete": "true"
}
```

### Output:

HTTP status code 204 is returned when an existing record is updated.

## Set a Translated Article Version to Incomplete

Sets a translated article version to incomplete.

### Syntax

#### URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<translationVersionID>
```

#### Formats

JSON, XML

#### HTTP Method

PATCH

#### Authentication

Authorization: OAuth *access\_token*

#### Parameters

Parameter	Description
complete	Set this value to <code>false</code> to set a translation to incomplete.

#### Input:

```
{
  "complete": "false"
}
```

#### Output:

HTTP status code 204 is returned when an existing record is updated.

## Submit an Article for Translation

Submits an article for translation and keeps the online version. Copies or creates a new draft translation if there is already a translation in the specified language. The publishing actions are defined by the field change you request on the resource.

### Syntax

#### URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations
```

#### Formats

JSON, XML

#### HTTP Method

POST

#### Authentication

Authorization: OAuth *access\_token*

**Parameters**

Parameter	Description
articleID	ID of the article
language	ISO code for the language

**Input:**

```
{
  "articleId":<articleID>
  "language":"fr"
}
```

## Unpublish the Master Version of an Article

Unpublishes the online master version of an article when there isn't an existing draft article.

The publishing actions are defined by the field change you request on the resource. To unpublish the master version, use "publishStatus":"draft".

## Syntax

**URI**

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionId>
```

**Formats**

JSON, XML

**HTTP Method**

PATCH

**Authentication**

Authorization: OAuth *access\_token*

**Parameters**

Parameter	Description
publishStatus	Publishing status of the article. Use <code>draft</code> .

**Input**

```
{
  "publishStatus":"draft"
}
```

**Output:**

HTTP status code 204 is returned when an existing record is updated.

## Unpublish the Online Version of a Translated Article

Unpublishes the online version of a translated article.

The publishing actions are defined by the field change you request on the resource. To edit and remove a translation from online state, use `"publishStatus": "draft"`.

### Syntax

#### URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<translationVersionID>
```

#### Formats

JSON, XML

#### HTTP Method

PATCH

#### Authentication

Authorization: OAuth *access\_token*

#### Parameters

Parameter	Description
<code>publishStatus</code>	Publishing status of the article. Use <code>draft</code> .

#### Input:

```
{
  "publishStatus": "draft"
}
```

## Parameterized Search

Executes a simple RESTful search using parameters instead of a SOSL clause. Indicate parameters in a URL in the GET method. Or, use POST for more complex JSON searches.

### Syntax

#### URI

```
/vXX.X/parameterizedSearch/?q=search string
```

#### Formats

JSON, XML


#### HTTP Method

GET, POST

#### Authentication

Authorization: Bearer *token*

## Required Global Parameters

Name	Description
q	<p>A search string that is properly URL-encoded.</p> <p> <b>Note:</b> SOSL clauses aren't supported.</p> <p>Available in version 36.0 and later.</p>

## Optional Global Parameters

Name	Type	Supported Methods	Description
dataCategory	string	GET	<p>Single value. If an organization uses Salesforce Knowledge articles or answers, <code>dataCategory</code> filters all search results based on one data category.</p> <p>For example, <code>dataCategory=GlobalCategory__c</code> below <code>NorthAmerica__c</code>.</p> <p>When using <code>dataCategories</code>, specify a Salesforce Knowledge article or answer type with <code>subject</code> and all the required parameters.</p> <p>For example:</p> <pre>q=tourism&amp;subject=KnowledgeArticleVersion&amp;KnowledgeArticleVersion.where=language='en_US'+and+publishStatus='online'&amp;KnowledgeArticleVersion.fields=id,title&amp;dataCategory=Location__c+Below+North_America__c</pre> <p>If you require multiple <code>dataCategory</code> filters, use <code>dataCategories</code> with the POST method.</p>
dataCategories	<a href="#">dataCategoriesFilter[]</a>	POST	<p>If an organization uses Salesforce Knowledge articles or answers, filters all search results based on one or more data categories.</p> <p>When using <code>dataCategories</code>, specify a Salesforce Knowledge article or answer type with <code>subjects</code> and the required parameters.</p> <p>For example:</p> <pre>{   "q": "Acme",   "fields": ["id", "title"],   "subjects": [{"name": "KnowledgeArticleVersion",     "where": "language='en_US' and publishstatus='draft'"}],   "dataCategories": [     {"groupName": "location__c", "operator": "below",     "categories": ["North_America__c"]}   ] }</pre>

Name	Type	Supported Methods	Description
defaultLimit	string	GET, POST	<p>Single value. The maximum number of results to return for each <code>subject</code> (GET) or <code>subjects</code> (POST) specified.</p> <p>The maximum <code>defaultLimit</code> is 2000.</p> <p>At least one <code>subject</code> must be specified.</p> <p>GET example:  <code>defaultLimit=10&amp;subject=Account&amp;subject=Contact</code>.</p> <p>When an <code>subject</code> limit is specified using <code>subject.limit=value</code>, such as <code>Account.limit=10</code>, this parameter is ignored for that object.</p>
division	string	GET, POST	<p>Single value. Filters search results based on the division field.</p> <p>For example in the GET method, <code>division=global</code>.</p> <p>Specify a division by its name rather than ID.</p> <p>All searches within a specific division also include the <code>global</code> division.</p>
fields	string	GET	<p>Comma-separated list of one or more fields to return in the response for each <code>subject</code> specified. At least one <code>subject</code> must be specified at the global level.</p> <p>For example: <code>fields=id&amp;subject=Account&amp;subject=Contact</code>.</p> <p>The global <code>fields</code> parameter is overridden when <code>subject</code> are specified using <code>subject.fields=field names</code>. For example, <code>Contact.fields=id,FirstName,LastName</code> would override the global setting of just returning the <code>id</code>.</p> <p>If unspecified, then the search results contain the IDs of records matching all fields for the specified object.</p> <p><b>Functions</b></p> <p>The following optional functions can be used within the <code>fields</code> parameter.</p> <ul style="list-style-type: none"> <li>• <code>toLabel</code>: Translates response field value into the user's language. For example, <code>Lead.fields=id,toLabel (Status)</code>. This function requires extra setup.</li> <li>• <code>convertCurrency</code>: Converts response currency fields to the user's currency. For example, <code>Opportunity.fields=id,convertCurrency (Amount)</code>. This function requires extra setup. Multi-currency must be enabled for your org.</li> <li>• <code>format</code>: Applies localized formatting to standard and custom number, date, time, and currency fields. For example, <code>Opportunity.fields=id,format (Amount)</code>.</li> </ul> <p>Aliasing is support within <code>fields</code> for <code>toLabel</code>, <code>convertCurrency</code>, and <code>format</code>. In addition, aliasing is required when the query includes the same field multiple times. For example,  <code>Opportunity.fields=id,format (Amount) AliasAmount</code></p>



Name	Type	Supported Methods	Description
fields	string[]	POST	<p>Array of one or more fields to return in the response for each <code>subjects</code> specified. At least one <code>subjects</code> must be specified at the global level.</p> <p>For example:</p> <pre data-bbox="667 415 1446 709"> {   "q": "Acme",   "fields": ["Id", "Name", "Phone"],   "subjects": [{"name": "Account"},                {"name": "Contact", "fields": ["Id",                "FirstName", "LastName"]},                {"name": "Lead"}] } </pre> <p>The global <code>fields</code> parameter is overridden when <code>subjectsFilter[]</code> fields are specified. Such as, in the previous example, <code>Id</code>, <code>FirstName</code>, and <code>LastName</code> is returned for <code>Contact</code> instead of the global fields of <code>Id</code>, <code>Name</code> and <code>Phone</code>.</p> <p>If unspecified, then the search results contain the IDs of records matching all fields for the specified object.</p> <p><b>Functions</b></p> <p>The following optional functions can be used within the <code>fields</code> parameter.</p> <ul style="list-style-type: none"> <li> <p><code>toLabel</code>: Translates response field value into the user's language. This function requires extra setup. For example:</p> <pre data-bbox="703 1129 1446 1339"> {   ...   "subjects": [ {"name": "Lead", "fields": ["Id",   "toLabel (Status)"} ],   ... } </pre> </li> <li> <p><code>convertCurrency</code>: Converts response currency fields to the user's currency. This function requires extra setup. Multi-currency must be enabled in the org. For example:</p> <pre data-bbox="703 1472 1446 1682"> {   ...   "subjects": [ {"name": "Opportunity",   "fields": ["Id", "convertCurrency (Amount)"} ]   ... } </pre> </li> <li> <p><code>format</code>: Applies localized formatting to standard and custom number, date, time, and currency fields. For example:</p> <pre data-bbox="703 1780 1446 1890"> {   ...   "subjects": [ {"name": "Opportunity", </pre> </li> </ul>

Name	Type	Supported Methods	Description
			<pre>"fields":["Id", "format(Amount)"]}] ... }</pre> <p>Aliasing is supported within <code>fields</code> for <code>toLabel</code>, <code>convertCurrency</code>, and <code>format</code>. In addition, aliasing is required when the query includes the same field multiple times. For example:</p> <pre>{ ... "subjects":[ {"name": "Opportunity", "fields":["Id", "format(Amount) AliasAmount" ]}] ... }</pre>
<code>in</code>	string	GET, POST	<p>Scope of fields to search. If you specify one or more scope values, the fields are returned for all found objects.</p> <p>Use one of the following values:</p> <ul style="list-style-type: none"> <li>• ALL</li> <li>• NAME</li> <li>• EMAIL</li> <li>• PHONE</li> <li>• SIDEBAR</li> </ul> <p>This clause doesn't apply to articles, documents, feed comments, feed items, files, products, and solutions. If any of these objects are specified, the search is not limited to specific fields; all fields are searched.</p>
<code>metadata</code>	string	GET, POST	<p>Specifies if metadata should be returned in the response. No metadata is returned by default. To include metadata in the response, use the <code>LABELS</code> value, which returns the display label for the fields returned in search results. For example:</p> <pre>?q=Acme&amp;metadata=LABELS</pre>
<code>networkIds</code>	string	GET	<p>Filters search results by a comma-separated list.</p> <p>A network ID represents the community ID.</p>
<code>networkIds</code>	string[]	POST	<p>Filters search results by an array.</p> <p>A network ID represents the community ID.</p>
<code>offset</code>	string	GET, POST	<p>Single value. The starting row offset into the result set returned.</p> <p>The maximum <code>offset</code> is 2000.</p> <p>Only one <code>subject</code> can be specified when using this parameter.</p>

Name	Type	Supported Methods	Description
overallLimit	string	GET, POST	<p>Single value. The maximum number of results to return across all <code>subject</code> parameters specified.</p> <p>The maximum <code>overallLimit</code> is 2000.</p>
pricebookId	string	GET, POST	<p>Single value. Filters product search results by a price book ID for only the Product2 object. The price book ID must be associated with the product that you're searching for. For example,</p> <pre>?q=laptop&amp;subject=product2&amp;pricebookId=01sxx000002MfFAAE</pre>
snippet	string	GET, POST	<p>The target length (maximum number of snippet characters) to return in Salesforce Knowledge article, case, case comment, feed, feed comment, idea, and idea comment search results. The <code>snippet</code> parameter displays contextual excerpts and highlights the search term for each article in the search results. Snippet results are used to differentiate matches to the search term in article search results. The target length can be from 50 to 1000 characters.</p> <p>Snippet and highlights are generated from email, text, and text area (long and rich) fields. Snippets aren't displayed for partially matching searches or if the user doesn't have access to the field that contains the snippet. Snippets are only displayed when 20 or fewer results are returned on a page.</p> <p>At least one of the following <code>subject</code> values must be specified.</p> <ul style="list-style-type: none"> <li>To search a specific article type, use the article type name with the suffix <code>__kav</code>.</li> <li>To search all article types, use <code>KnowledgeArticleVersion</code>.</li> <li>To search case, case comment, feed, feed comment, idea, and idea comment types, use <code>Case</code>, <code>CaseComment</code>, <code>FeedItem</code>, <code>FeedComment</code>, <code>Idea</code>, and <code>IdeaComment</code>.</li> </ul> <p>For example, <code>q=tourism&amp;subject=Case&amp;snippet=500</code>.</p>
subject	string	GET	<p>Objects to return in the response. Must be a valid object type.</p> <p>You can use multiple <code>subject</code> values, such as <code>subject=Account&amp;subject=Contact</code>.</p> <p>If unspecified, then the search results contain the IDs of all objects.</p>
subjects	<a href="#">subjectsFilter[]</a>	POST	<p>Objects to return in the response. Must contain valid object types. Use with the required parameters.</p> <p>For example:</p> <pre>{   "q": "Acme",   "fields": ["id", "title"],   "subjects": [{"name": "Solution__kav"},   "where": "language='en_US' and publishstatus='draft'"},</pre>

Name	Type	Supported Methods	Description
			<pre> {"name": "FAQ__kav", "where": "language='en_US' and publishstatus='draft'"} } </pre> <p>If unspecified, then the search results contain the IDs of all objects.</p>
updateTracking	string	GET, POST	<p>Specifies a value of <code>true</code> to track keywords that are used in Salesforce Knowledge article searches only.</p> <p>If unspecified, the default value of <code>false</code> is applied.</p>
updateViewStat	string	GET, POST	<p>Specifies a value of <code>true</code> to update an article's view statistics. Valid only for Salesforce Knowledge article searches.</p> <p>If unspecified, the default value of <code>false</code> is applied.</p>

#### dataCategoriesFilter[] Parameters

Parameters must be specified in the order presented in the table (`groupName`, `operator`, `categories`).

Name	Type	Description
groupName	string	The name of the data category group to filter by.
operator	string	Valid values: <ul style="list-style-type: none"> <li>• ABOVE</li> <li>• ABOVE_OR_BELOW</li> <li>• AT</li> <li>• BELOW</li> </ul>
categories	string[]	The name of the categories to filter by.

#### subjectsFilter[] Parameters (POST Method Only)

Name	Type	Description
fields	string[]	Array of one or more fields to return in the response for the <code>subject</code> .
limit	string	Specify the maximum number of rows that are returned for the <code>subject</code> .
name	string	Name of the <code>subject</code> to return in the response.

Name	Type	Description
orderBy	string	<p>Controls the field order of the results using the following syntax "orderBy" : "field {ASC DESC} [NULLS_{FIRST LAST}]"</p> <p>For example:</p> <pre>{   ...   "subjects":[ {"name": "Account", "fields":["Id", "name"], "orderBy":     "Name DESC Nulls_last"}]   ... }</pre> <ul style="list-style-type: none"> <li>• ASC: ascending. Default.</li> <li>• DESC: descending.</li> <li>• NULLS_FIRST: Null records at the beginning of the results. Default.</li> <li>• NULLS_LAST: Null records at the end of the results.</li> </ul>
where	string	<p>Filter search results for this object by specific field values.</p> <p>For example, where : conditionExpression. Here the conditionExpression of the WHERE clause uses the following syntax: fieldExpression [logicalOperator fieldExpression2 ... ].</p> <p>Add multiple field expressions to a condition expression by using logical and comparison operators.</p>

### subject-level Parameters (GET Method Only)

The following optional parameters can be used with the `subject` parameter in a GET method to further refine search results. These settings would override any settings specified at the global level.

The format is `subject.parameter`, such as `Account.fields`. An `subject` must be specified to use these parameters, for example, `subject=Account&Account.fields=id,name`.

Name	Type	Description
fields	string	<p>Comma-separated list of one or more fields to return in the response.</p> <p>For example, <code>KnowledgeArticleVersion.fields=id,title</code>.</p>
limit	string	<p>Specifies the maximum number of rows that are returned for the <code>subject</code>.</p> <p>For example, <code>Account.limit=10</code>.</p>
orderBy	string	<p>Controls the field order of the results using the following syntax <code>orderBy = field {ASC DESC} [NULLS_{FIRST LAST}]</code></p> <p>For example: <code>Account.orderBy=Name</code></p> <ul style="list-style-type: none"> <li>• ASC: ascending. Default.</li> <li>• DESC: descending.</li> <li>• NULLS_FIRST: Null records at the beginning of the results. Default.</li> <li>• NULLS_LAST: Null records at the end of the results.</li> </ul>

Name	Type	Description
where	string	<p>Filter search results for this object by specific field values.</p> <p>Forexample, <code>Account.where = conditionExpression</code>. Here the <code>conditionExpression</code> of the WHERE clause uses the following syntax: <code>fieldExpression [logicalOperator fieldExpression2 ... ]</code>.</p> <p>Add multiple field expressions to a condition expression by using logical and comparison operators. For example, <code>KnowledgeArticleVersion.where=publishstatus='online' and language='en_US'</code>.</p>

### Example GET Method

```
.../v37.0/parameterizedSearch/?q=Acme&subject=Account&Account.fields=id,name&Account.limit=10
```

### Example POST Method

```
{
  "q": "Smith",
  "fields" : ["id", "firstName", "lastName"],
  "subjects": [{"fields": ["id", "NumberOfEmployees"],
    "name": "Account",
    "limit": 20},
    {"name": "Contact"}],
  "in": "ALL",
  "overallLimit": 100,
  "defaultLimit": 10
}
```

## Search

Executes the specified SOSL search. The search string must be URL-encoded.

For more information on SOSL see the [Force.com SOQL and SOSL Reference](#).

## Syntax

### URI

```
/vXX.X/search/?q=SOSL search string
```

### Formats

JSON, XML

### HTTP Method

GET

### Authentication

Authorization: Bearer **token**

**Parameters**

Parameter	Description
q	A SOSL statement that is properly URL-encoded.

**Example**

See Search for a String.

## Search Scope and Order

Returns an ordered list of objects in the default global search scope of a logged-in user. Global search keeps track of which objects the user interacts with and how often and arranges the search results accordingly. Objects used most frequently appear at the top of the list.

The returned list reflects the object order in the user's default search scope, including any pinned objects on the user's search results page. This call is useful if you want to implement a custom search results page using the optimized global search scope. The search string must be URL-encoded.

## Syntax

**URI**

`/vXX.X/search/scopeOrder`

**Formats**

JSON, XML

**HTTP Method**

GET

**Authentication**

Authorization: Bearer *token*

**Example**

See Get the Default Search Scope and Order.

## Search Result Layouts

Returns search result layout information for the objects in the query string. For each object, this call returns the list of fields displayed on the search results page as columns, the number of rows displayed on the first page, and the label used on the search results page.

This call supports bulk fetch for up to 100 objects in a query.

## Syntax

**URI**

`/vXX.X/search/layout/?q=Comma delimited object list`

**Formats**

JSON, XML

**HTTP Method**

GET

**Authentication**

Authorization: Bearer *token*

**Example**

Get Search Result Layouts for Objects

## Search Suggested Records

Returns a list of suggested records whose names match the user's search string. The suggestions resource provides a shortcut for users to navigate directly to likely relevant records, before performing a full search.

### Syntax

**URI**

vXX.X/search/suggestions?q=*search string*&subject=*object types*

**Available since release**

32.0

**Formats**

JSON, XML

**HTTP methods**

GET

**Authentication**

Authorization: Bearer *token*

**Request body**

None required

**Request parameters**

Parameter	Description
<code>fields</code>	Optional. Used for creating lookup queries. Specify multiple fields using a comma-separated list. Specifies which lookup fields to be returned in the response.
<code>groupId</code>	Optional. Specifies one or more unique identifiers of one or more groups that the question to return was posted to. Specify multiple groups using a comma-separated list. This parameter is only applicable when the parameter type equals <code>question</code> . Don't use with the <code>userId</code> .
<code>ignoreUnsupportedSObjects</code>	Optional. Specifies what to do if unsupported objects are included in the request. If <code>false</code> and an unsupported object is included, an error is returned. If <code>true</code> and an unsupported object is included, the object is ignored and no error is returned. See the Unsupported Objects section for reference. The default is <code>false</code> .
<code>limit</code>	Optional. Specifies the maximum number of suggested records to return. If a limit isn't specified, 5 records are returned by default. If there are more suggested records than the limit specified, the response body's <code>hasMoreResults</code> property is <code>true</code> .
<code>networkId</code>	Optional. Specifies one or more unique identifiers for the community(ies) that the question to return is associated to. Specify multiple communities using a





Parameter	Description
	comma-separated list. This parameter is only applicable when the parameter <code>type</code> equals <code>question</code> or parameter <code>subject</code> equals <code>user</code> .
<code>q</code>	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: one character for queries in Chinese, Japanese, Korean, and Thai; three characters for all other languages. Query strings that exceed the maximum length of 255 characters (or 200 consecutive characters without a space break) return an error.
<code>subject</code>	<p>Required. The objects that the search is scoped to, such as <code>Account</code> or <code>offer__c</code>.</p> <p>If the <code>subject</code> value is <code>feedItem</code>, it is required to have the <code>type</code> parameter with a value of <code>question</code>.</p> <p>Specify up to 10 objects with a comma-separated list. For example: <code>subject=Account, Contact, Lead</code>. To take advantage of this feature, activate the <b>CrossObjectTypeahead</b> permission.</p> <p>To specify the specific fields to return by object, use the following syntax with multiple fields in a comma-separated list. The <code>subject</code> is lowercase.</p> <pre>subject=<b>subject</b>.fields=<b>fields</b></pre> <p>For example:</p> <pre>&amp;subject=Account, Contact, Lead&amp;account.fields=Website, Phone &amp;contact.fields=Phone</pre>
<code>topicId</code>	Optional. Specifies the unique identifier of the single topic that the question to return was tagged as. This parameter is only applicable when the parameter <code>type</code> equals <code>question</code> .
<code>type</code>	Required when the <code>subject</code> value is <code>feedItem</code> . Including this parameter for all other <code>subject</code> values doesn't affect the query. Specifies that the type of Feed is questions. Valid value: <code>question</code> .
<code>userId</code>	Optional. Specifies one or more unique identifiers of one or more users who authored the question to return. Specify multiple users using a comma-separated list. This parameter is only applicable when the parameter <code>type</code> equals <code>question</code> . Don't use with the <code>groupId</code> .
<code>where</code>	<p>Optional. A filter constraint following the same syntax as the SOQL <code>where</code> clause. For example: <code>my_field__c LIKE 'foo%' AND RecordType='bar'</code> This expression needs to be properly URL encoded. Can't be used for <code>Question</code> object.</p> <p>You can set the clause on a per-entity basis or globally for all compatible entities. An object-specific clause example is: <code>account.where=name%20LIKE%20%27Smith%25%27</code>. And the example global clause is: <code>where=name%20LIKE%20%27Smith%25%27</code>. The parameter must be lower case. Any object-specific <code>where</code> clauses overrides the global <code>where</code> clause.</p>

Parameter	Description
	<p>To specify multiple entities, see the following example. This feature is available in version 38.0 and later.</p> <pre> ...search/suggestions?q=Smith &amp;subject=Account,Contact,KnowledgeArticleVersion,CollaborationGroup,Topic,FeedItem  // Specifies a global where clause (to filter Account and Contact) &amp;where=name%20LIKE%20%27Smith%25%27 // Overrides the global where clause for Knowledge Article (filtering by PublishStatus and Language is required for KnowledgeArticle) &amp;knowledgearticleversion.where=PublishStatus='online'+and+language='en_US' // Overrides the global where clause for Topic &amp;topic.where=networkid=&lt;1234567891&gt; // Overrides the global where clause for CollaborationGroup &amp;collaborationgroup.where=networkid=&lt;1234567891&gt; // FeedItem-Question doesn't support where clauses, but we can filter the type and networkId &amp;type=question &amp;networkId==&lt;1234567891&gt; </pre>

## Usage

The suggestions resource returns records when the record's name field includes the exact text in the search string. The last term in the search string can match the beginning of a word. Records that contain the search string within a word aren't considered a match.

-  **Note:** If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI.
-  **Example:** The text string *national u* is treated as *national u\** and returns "National Utility", "National Urban Company", and "First National University".

## Suggested Records Response

The suggestions resource returns display-ready data about likely relevant records that the user can access.

The order of results is determined by a relevance algorithm.

Each suggested record in the results contains these elements:

Element	Description
Attributes	<p>The record's object type and the URL for accessing the record.</p> <p>Also includes the requested lookup fields' values. For example, if you requested <code>fields=Id,Name</code>, the result would include the ID and name.</p>

Element	Description
Name (or Title)	<p>The record's Name field. In the absence of a standard Name field, the Title field is used for these objects:</p> <ul style="list-style-type: none"> <li>• Dashboard</li> <li>• Idea</li> <li>• IdeaTheme</li> <li>• Note</li> <li>• Question</li> </ul> <p>In the absence of a standard Name or Title field, the main identifying field is used. For example, in cases, the Case Number is used.</p>
Id	The record's unique identifier.

## Example JSON Response Body

```
[ {
  "attributes" : {
    "type" : "Account",
    "url" : "/services/data/v32.0/subjects/Account/001xx000003DH6WAAW"
  },
  "Id" : "001xx000003DH6WAAW"
  "Name" : "National Utility Service"
}, {
  {
    "attributes" : {
      "type" : "Account",
      "url" : "/services/data/v32.0/subjects/Account/001xx000003DHJ4AAO"
    },
    "Id" : "001xx000003DHJ4AAO"
    "Name" : "National Utility Service"
  }, {
  {
    "attributes" : {
      "type" : "Account",
      "url" : "/services/data/v32.0/subjects/Account/001xx000003DHscAAG"
    },
    "Id" : "001xx000003DHscAAG"
    "Name" : "National Urban Technology Center"
  }
} ]
```

## Example JSON Response Body for a Multiple Object Request

```
[ {
  "attributes" : {
    "type" : "Account",
    "url" : "/services/data/v38.0/subjects/Account/001xx000003DMEKAA4"
  },
  "Id" : "001xx000003DMEKAA4"
}
```

```

    "Name" : "Joe Doe Printing"
  }, {
  {
    "attributes" : {
      "type" : "Account",
      "url" : "/services/data/v38.0/subjects/Account/001xx000003DLjvAAG"
    },
    "Id" : "001xx000003DLjvAAGO"
    "Name" : "Joe Doe Plumbing"
  }, {
  {
    "attributes" : {
      "type" : "Contact",
      "url" : "/services/data/v38.0/subjects/Contact/003xx000004U9Y9AAK"
    },
    "Id" : "003xx000004U9Y9AAK"
    "Name" : "John Doe"
  } ]

```

## Example XML Response Body

```

<?xml version="1.0" encoding="UTF-8"?
<suggestions>
  <autoSuggestResults type="Account"
url="/services/data/v32.0/subjects/Account/001xx000003DH6WAAW">
    <Id>001xx000003DH6WAAW</Id>
    <Name>National Utility Service</Name>
  </autoSuggestResults>
  <autoSuggestResults type="Account"
url="/services/data/v32.0/subjects/Account/001xx000003DHJ4AAO">
    <Id>001xx000003DHJ4AAO</Id>
    <Name>National Utility Service</Name>
  </autoSuggestResults>
  <autoSuggestResults type="Account"
url="/services/data/v32.0/subjects/Account/001xx000003DHscAAG">
    <Id>001xx000003DHscAAG</Id>
    <Name>National Urban Technology Center</Name>
  </autoSuggestResults>
  <hasMoreResults>true</hasMoreResults>
</suggestions>

```

## Unsupported Objects

The suggestions resource supports all searchable objects except the following.

- ContentNote
- Event
- FeedComment
- FeedPost
- IdeaComment
- Pricebook2

- Reply
- TagDefinition
- Task

## Search Suggested Article Title Matches

Returns a list of Salesforce Knowledge article titles that match the user's search query string. Provides a shortcut to navigate directly to likely relevant articles before the user performs a search.

### Syntax

#### URI

```
/vXX.X/search/suggestTitleMatches?q=search string&language=article language&publishStatus=article publication status
```

#### Available since release

30.0

#### Formats

JSON, XML

#### HTTP methods

GET

#### Authentication

Authorization: Bearer **token**

#### Request body

None required

#### Request parameters

Parameter	Description
<code>articleTypes</code>	Optional. Three-character ID prefixes indicating the desired article types. You can specify multiple values for this parameter in a single REST call, by repeating the parameter name for each value. For example, <code>articleTypes=ka0&amp;articleTypes=ka1</code> .
<code>categories</code>	Optional. The name of the data category group and name of the data category for desired articles, expressed as a JSON mapping. You can specify multiple data category group and data category pairs in this parameter. For example, <code>categories={"Regions": "Asia", "Products": "Laptops"}</code> . Characters in the URL might need to be encoded. For this example, <code>categories=%7B%22Regions%22%3A%22Asia%22%2C%22Products%22%3A%22Laptops%22%7D</code> .
<code>channel</code>	Optional. The channel where the matching articles are visible. Valid values: <ul style="list-style-type: none"> <li>• <code>AllChannels</code>—Visible in all channels the user has access to</li> <li>• <code>App</code>—Visible in the internal Salesforce Knowledge application</li> <li>• <code>Pkb</code>—Visible in the public knowledge base</li> <li>• <code>Csp</code>—Visible in the Customer Portal</li> <li>• <code>Ppm</code>—Visible in the Partner Portal</li> </ul>

Parameter	Description
	<p>If <code>channel</code> isn't specified, the default value is determined by the type of user.</p> <ul style="list-style-type: none"> <li>• <code>Pkb</code> for a guest user</li> <li>• <code>Csp</code> for a Customer Portal user</li> <li>• <code>Prm</code> for a Partner Portal user</li> <li>• <code>App</code> for any other type of user</li> </ul> <p>If <code>channel</code> is specified, the specified value may not be the actual value requested, because of certain requirements.</p> <ul style="list-style-type: none"> <li>• For guest, Customer Portal, and Partner Portal users, the specified value must match the default value for each user type. If the values don't match or <code>AllChannels</code> is specified, then <code>App</code> replaces the specified value.</li> <li>• For all users other than guest, Customer Portal, and Partner Portal users: <ul style="list-style-type: none"> <li>– If <code>Pkb</code>, <code>Csp</code>, <code>Prm</code>, or <code>App</code> are specified, then the specified value is used.</li> <li>– If <code>AllChannels</code> is specified, then <code>App</code> replaces the specified value.</li> </ul> </li> </ul>
<code>language</code>	Required. The language of the user's query. Specifies the language that matching articles are written in.
<code>limit</code>	Optional. Specifies the maximum number of articles to return. If there are more suggested articles than the limit specified, the response body's <code>hasMoreResults</code> property is <code>true</code> .
<code>publishStatus</code>	Required. The article's publication status. Valid values: <ul style="list-style-type: none"> <li>• <code>Draft</code>—Articles aren't published in Salesforce Knowledge.</li> <li>• <code>Online</code>—Articles are published in Salesforce Knowledge.</li> <li>• <code>Archived</code>—Articles aren't published and are available in Archived Articles view.</li> </ul>
<code>q</code>	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: one character for queries in Chinese, Japanese, and Korean, and three characters for all other languages. Query strings exceeding the maximum length of 250 characters return an error.
<code>topics</code>	Optional. The topic of the returned articles. For example: <code>topics=outlook&amp;topics=email</code> .
<code>validationStatus</code>	Optional. The validation status of returned articles.

## Example for getting suggested articles with matching titles

```
curl https://yourInstance.salesforce.com/services/data/v30.0/search/suggestTitleMatches?
q=orange+banana&language=en_US&publishStatus=Online -H "Authorization: Bearer token"
```

## Example JSON response body


```
{
  "autoSuggestResults" : [ {
    "attributes" : {
      "type" : "KnowledgeArticleVersion",
      "url" : "/services/data/v30.0/subjects/KnowledgeArticleVersion/ka0D00000004CcQ"
    },
    "Id" : "ka0D00000004CcQ",
    "UrlName" : "orange-banana",
    "Title" : "orange banana",
    "KnowledgeArticleId" : "kA0D00000004Cfz"
  } ],
  "hasMoreResults" : false
}
```

## Usage

Salesforce Knowledge must be enabled in your organization. The user must have the “View Articles” permission enabled. The articles suggested include only the articles the user can access, based on the data categories and article types the user has permissions to view.


The Suggest Article Title Matches resource is designed to return display-ready data about likely relevant articles. Articles are suggested if their titles contain the entire query string, except stopwords, such as “a,” “for,” and “the.”

For example, a search for *Backpacking for desert* returns the article, “Backpacking in the desert.”

 **Note:** Articles with titles that include stopwords from the query string, such as “Backpacking for desert survival” in this example, appear before matching articles with titles that don’t include the stopwords.

Stopwords at the end of the query string are treated as search terms.

A wildcard is automatically appended to the last token in the query string.

 **Note:** If the user’s search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI along with any other special characters.

If the number of suggestions returned exceeds the limit specified in the request, the end of the response contains a field called `hasMoreResults`. Its value is `true` if the suggestions returned are only a subset of the suggestions available, and `false` otherwise.

## Search Suggested Queries

Returns a list of suggested searches based on the user’s query string text matching searches that other users have performed in Salesforce Knowledge. Provides a way to improve search effectiveness, before the user performs a search.

## Syntax

### URI

`vXX.X/search/suggestSearchQueries?q=search string&language=language of query`

### Available since release

30.0

### Formats

JSON, XML

**HTTP methods**

GET

**Authentication**Authorization: Bearer *token***Request body**

None required

**Request parameters**

Parameter	Description
<code>channel</code>	<p>Optional. Specifies the Salesforce Knowledge channel where the article can be viewed. Valid values:</p> <ul style="list-style-type: none"> <li>• <code>AllChannels</code>—Visible in all channels the user has access to</li> <li>• <code>App</code>—Visible in the internal Salesforce Knowledge application</li> <li>• <code>Pkb</code>—Visible in the public knowledge base</li> <li>• <code>Csp</code>—Visible in the Customer Portal</li> <li>• <code>Prm</code>—Visible in the Partner Portal</li> </ul> <p>If <code>channel</code> isn't specified, the default value is determined by the type of user.</p> <ul style="list-style-type: none"> <li>• <code>Pkb</code> for a guest user</li> <li>• <code>Csp</code> for a Customer Portal user</li> <li>• <code>Prm</code> for a Partner Portal user</li> <li>• <code>App</code> for any other type of user</li> </ul> <p>If <code>channel</code> is specified, the specified value may not be the actual value requested, because of certain requirements.</p> <ul style="list-style-type: none"> <li>• For guest, Customer Portal, and Partner Portal users, the specified value must match the default value for each user type. If the values don't match or <code>AllChannels</code> is specified, then <code>App</code> replaces the specified value.</li> <li>• For all users other than guest, Customer Portal, and Partner Portal users: <ul style="list-style-type: none"> <li>– If <code>Pkb</code>, <code>Csp</code>, <code>Prm</code>, or <code>App</code> are specified, then the specified value is used.</li> <li>– If <code>AllChannels</code> is specified, then <code>App</code> replaces the specified value.</li> </ul> </li> </ul>
<code>language</code>	Required. The language of the user's query.
<code>limit</code>	Optional. Specifies the maximum number of suggested searches to return. If there are more suggested queries than the limit specified, the response body's <code>hasMoreResults</code> property is <code>true</code> .
<code>q</code>	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: one character for queries in Chinese, Japanese, and Korean, and three characters for all other languages. Query strings exceeding the maximum length of 250 characters return an error.



## Example for getting suggested queries

```
curl https://yourInstance.salesforce.com/services/data/v30.0/search/suggestSearchQueries?q=app&language=en_US -H "Authorization: Bearer token"
```

## Example JSON response body

```
{
  "autoSuggestResults" : [ {
    "0" : "apple",
    "1" : "apple banana",
  } ],
  "hasMoreResults" : false
}
```

## Usage

Salesforce Knowledge must be enabled in your organization.

Queries are suggested if they exactly match the query string text. The text string must be a prefix within the query; it's not considered a match if it appears within a word. For example, the text string *app* would return suggested queries *apple banana* and *banana apples* but not *pineapple*.

If the number of suggestions returned exceeds the limit specified in the request, the end of the response contains a field called `hasMoreResults`. Its value is `true` if the suggestions returned are only a subset of the suggestions available, and `false` otherwise.

If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI.

## Support Knowledge with REST API

---

Knowledge Support REST APIs allow both authorized and guest users to retrieve the user's visible data categories and their associated articles.

Authenticated users need the `UserProfile.apiEnabled` permission, Knowledge enabled in the organization, read rights on the article type, and any other knowledge specific permission or preference that controls visibility to articles.

Guest users need the `Guest Access to the Support API` preference enabled on the relevant Site, Knowledge enabled in the organization, and read rights on the article type and article channel that controls the visibility for guest users.

## Syntax

The root endpoint for all Knowledge support APIs that can be open to guest users.

### Available since release

38.0

### Method

GET

### Formats

JSON, XML

**Authentication**

OAuth accesstoken

**Endpoint**

&lt;prefix&gt;/support

**HTTP headers****Accept:** Optional. Can be either `application/json` or `application/xml`.**Input**

None

**Output**

The following resources are related to Knowledge.

```
{
  "dataCategoryGroups" : "/services/data/vxx.0/support/dataCategoryGroups",
  "knowledgeArticles" : "/services/data/vxx.0/support/knowledgeArticles"
  :
}
```

Where `vxx.0` is the API version you requested.**IN THIS SECTION:**[Data Category Groups](#)

Get data category groups that are visible to the current user.

[Data Category Detail](#)

Get data category details and the child categories by a given category.

[Articles List](#)

Get a page of online articles for the given language and category through either search or query.

[Articles Details](#)

Get all article fields, accessible to the user.

## Data Category Groups

Get data category groups that are visible to the current user.

### Syntax

**Available since release**

38.0

**Method**

GET

**Formats**

JSON, XML

**Authentication**

OAuth accesstoken

**Endpoint**

[prefix]/support/dataCategoryGroups

**HTTP headers**

**Accept:** Optional. Can be either `application/json` or `application/xml`.

**Accept-language:** Optional. Language to translate the categories. Any ISO-639 language abbreviation, and an ISO-3166 country code subtag in the HTTP Accept-Language header. Only one language accepted. If no language specified, the non-translated labels are returned.

**Input:**

string `sObjectName`: Required. `KnowledgeArticleVersion` only.

boolean `topCategoriesOnly`: Optional. Defaults to true

- True returns only the top level categories.
- False returns the entire tree.

 **Note:** All the input parameters are case-sensitive.


**Output:**

A list of the active data category groups that are visible to the current user in the site context. Returns id, name, label, and their top level categories or the entire data category group tree that are visible to the current user. The labels must be translated to the given language if they are available.

- **Data Category Group List**

This payload lists the active root Data Category Groups that can be used in other requests to return the data categories and articles related to it.

```
{
  "categoryGroups": [ Data Category Group, ...],
}
```

 **Note:** Returns only the active groups that are associated to the given entity (by `sObjectName`). Only `KnowledgeArticleVersion` is supported.

- **Data Category Group**

This represents an individual data category group, and its root category.


```
{
  "name": String, // the unique name of the category group
  "label": String, // returns the translated version if it is available
  "objectUsage" : String, // currently only "KnowledgeArticleVersion" is available.

  "topCategories": [ Data Category Summary, ...]
}
```

- **Data Category Summary**

This provides a summary of data category information. The Summary and Detail responses share common properties, with the goal of providing only as much information as is necessary from associated resources.

```
{
  "name": String, // the unique name of the category
  "label": String, // returns the translated version if it is available
  "url": URL, // the url points to the data category detail API
  "childCategories": [ Data Category Summary, ...] // null if topCategoriesOnly is true
}
```

 **Note:** The URL property is a pre-calculated path to the unique resource representing this data category, in this case it is a [Data Category Detail](#) API.

## Example

### Input

```
/services/data/v38.0/support/dataCategoryGroups?sObjectName=KnowledgeArticleVersion
```

### Output

```
{
  "categoryGroups" : [ {
    "label" : "Doc",
    "name" : "Doc",
    "objectUsage" : "KnowledgeArticleVersion",
    "topCategories" : [ {
      "childCategories" : null,
      "label" : "All",
      "name" : "All",
      "url" :
"/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion"
    } ]
  }, {
    "label" : "Manual",
    "name" : "Manual",
    "objectUsage" : "KnowledgeArticleVersion",
    "topCategories" : [ {
      "childCategories" : null,
      "label" : "All",
      "name" : "All",
      "url" :
"/services/data/v38.0/support/dataCategoryGroups/Manual/dataCategories/All?sObjectName=KnowledgeArticleVersion"
    } ]
  } ]
}
```

## Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. Use the language code format used in [Which Languages Does Salesforce Support?](#)

Only the user's visible data categories are returned. A user might be able to see several sub trees in the category group, therefore, the top categories that are visible to the user in each group are returned.

## Data Category Detail

Get data category details and the child categories by a given category.

## Syntax

### Available since release

38.0

### Method

GET

### Formats

JSON, XML

### Authentication

OAuth accesstoken

### Endpoint

[prefix]/support/dataCategoryGroups/[group]/dataCategories/[category]

### HTTP headers

**Accept:** Optional. Can be either `application/json` or `application/xml`.

**Accept-language:** Optional. Language to translate the categories. Any ISO-639 language abbreviation, and an ISO-3166 country code subtag in the HTTP Accept-Language header. Only one language accepted. If no language specified, the non-translated labels are returned.

### Input:

string sObjectName: Required. KnowledgeArticleVersion only.

### Output:

Details of the category and a list of child categories (name, label, etc.).

- **Data Category Detail**

Used for situations where the hierarchical representation of data categories is important. The child property contains a list of child data categories.

```
{
  "name": String, // the unique name of the category
  "label": String, // returns the translated version if it is available
  "url": URL,
  "childCategories": [ Data Category Summary, ...],
}
```



**Note:** If the category isn't visible to the current user the return is empty.

## Example

### Input

```
/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion
```

### Output

```
{
  "childCategories" : [ {
    "childCategories" : null,
    "label" : "Help",
    "name" : "Help",
    "url" :
  }
```

```

"/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/Help?sObjectName=KnowledgeArticleVersion"
  }, {
    "childCategories" : null,
    "label" : "QA",
    "name" : "QA",
    "url" :
"/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/QA?sObjectName=KnowledgeArticleVersion"
  } ],
  "label" : "All",
  "name" : "All",
  "url" :
"/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion"
}

```

## Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. Use the language code format used in [Which Languages Does Salesforce Support?](#).

## Articles List

Get a page of online articles for the given language and category through either search or query.

## Syntax

### Available since release

38.0

### Method

GET

### Formats

JSON, XML

### Authentication

OAuth accesstoken

### Endpoint

[prefix]/support/knowledgeArticles

### HTTP headers

**Accept:** Optional. Can be either `application/json` or `application/xml`.

**Accept-language:** Required. The article must be an active language in the user's organization

- If the language code isn't valid, an error message is returned: "The language code is not valid or not supported by Knowledge."
- If the language code is valid, but not supported by Knowledge, then an error message is returned: "Invalid language code. Check that the language is included in your Knowledge language settings."

### Input:

string `q`: Optional, Performs an SOSL search. If the query string is null, empty, or not given, an SOQL query runs.

string `channel`: Optional, defaults to user's context. For information on channel values, see [Valid channel values](#).


- **App:** Visible in the internal Salesforce Knowledge application
- **Pkb:** Visible in the public knowledge base
- **Csp:** Visible in the Customer Portal
- **Prm:** Visible in the Partner Portal

string `categories` in map json format { "group1": "category1", "group2": "category2", ... } )

Optional, defaults to None. Category group must be unique in each group:category pair, otherwise you get `ARGUMENT_OBJECT_PARSE_ERROR`.

string `queryMethod` values are: `AT`, `BELOW`, `ABOVE`, `ABOVE_OR_BELOW`. Only valid when categories are specified, defaults to `ABOVE_OR_BELOW`.

string `sort`: Optional, a sortable field name `LastPublishedDate`, `CreatedDate`, `Title`, `ViewScore`. Defaults to `LastPublishedDate` for query and relevance for search.

 **Note:** When sorting on `ViewScore` it is only available for query, not search, and no pagination is supported. You only get one page of results.

string `order`: Optional, either `ASC` or `DESC`, defaults to `DESC`. Valid only when `sort` is valid.

integer `pageSize`: Optional, defaults to 20. Valid range 1 to 100.

integer `pageNumber` : Optional, defaults to 1.

### Output:

A page of online articles in the given language and category visible to the current user.

- **Article Page**

A page of articles. The individual entries are article summaries so the size can be kept at a minimum.

```
{
  "articles": [ Article Summary, ... ], // list of articles
  "currentPageUrl": URL, // the article list API with current page number
  "nextPageUrl": URL, // the article list API with next page number,
  // which can be null if there are no more articles.
  "pageNumber": Int // the current page number, starting at 1.
}
```

 **Note:** The API supports paging. Each page of responses includes a URL to its page, as well as the URL to the next page of articles.

- **Article Summary**

A summary of an article used in a list of article responses. It shares similar properties to the Article Detail representation, as one is a superset of the other.

```
{
  "id": Id, // articleId
  "articleNumber": String,
  "title": String,
  "summary": String,
  "url": URL, // to the Article Detail API
  "viewCount": Int, // view count in the interested channel
  "viewScore": double (in xxx.xxxx precision), // view score in the interested
  channel.
  "upVoteCount": int, // up vote count in the interested channel.
  "downVoteCount": int, // down vote count in the interested channel.
}
```

```

    "lastPublishedDate": Date // last publish date in ISO8601 format
    "categoryGroups": [ Data Category Group, ... ]
  }

```

The "url" property always points to the [Article Details](#) resource endpoint. For information on valid channel values, see the [channel parameter description](#)

- **Data Category Group**

An individual data category group, its root category, and a list of selected data categories in the group.

```

{
  "groupName": String, // the unique name of the category group
  "groupLabel": String, // returns the translated version
  "selectedCategories": [ Data Category Summary, ... ]
}

```

- **Data Category Summary**

Provides a summary of data category information. The Summary and Detail responses share common properties.

```

{
  "categoryName": String, // the unique name of the category
  "categoryLabel": String, // returns the translated version, per the API
  language specified
  "url": String // returns the url for the DataCategory REST API.
}

```



**Note:** The outputs of Data Category Group and Data Category Summary in Article List API are different from the Data Category Groups API.

## Example

### Input

```

/services/data/v38.0/support/knowledgeArticles?sort=ViewScore&channel=Pkb&pageSize=3
HTTP Headers:
  Content-Type: application/json; charset=UTF-8
  Accept: application/json
  Accept-Language: en-US

```

### Output

```

{
  "articles" : [ {
    "articleNumber" : "000001002",
    "categoryGroups" : [ ],
    "downVoteCount" : 0,
    "id" : "kA0xx000000000BCAQ",
    "lastPublishedDate" : "2015-02-25T02:07:18Z",
    "summary" : "With this online Chinese input tool, you can type Chinese characters
    through your web browser without installing any Chinese input software in your system.
    The Chinese online input tool uses the popular Pin Yin input method. It is a fast and
    convenient tool to input Chinese on English OS environments.",
    "title" : "Long text test",
    "upVoteCount" : 0,
    "url" : "/services/data/v38.0/support/knowledgeArticles/kA0xx000000000BCAQ",
  }
]
}

```



```

"viewCount" : 4,
"viewScore" : 100.0
}, {
"articleNumber" : "000001004",
"categoryGroups" : [ ],
"downVoteCount" : 0,
"id" : "kA0xx000000000LCAQ",
"lastPublishedDate" : "2016-06-21T21:11:02Z",
"summary" : "The number of characters required for complete coverage of all these
languages' needs cannot fit in the 256-character code space of 8-bit character encodings,
requiring at least a 16-bit fixed width encoding or multi-byte variable-length encodings.
\r\n\r\nAlthough CJK encodings have common character sets, the encodings often used to
represent them have been developed separately by different East Asian governments and
software companies, and are mutually incompatible. Unicode has attempted, with some
controversy, to unify the character sets in a process known as Han unification.\r\n\r\nCJK
character encodings should consist minimally of Han characters p",
"title" : "Test Images",
"upVoteCount" : 0,
"url" : "/services/data/v38.0/support/knowledgeArticles/kA0xx00000000LCAQ",
"viewCount" : 0,
"viewScore" : 0.0
}, {
"articleNumber" : "000001012",
"categoryGroups" : [ ],
"downVoteCount" : 0,
"id" : "kA0xx0000000006GCAQ",
"lastPublishedDate" : "2016-06-21T21:10:48Z",
"summary" : null,
"title" : "Test Draft 2",
"upVoteCount" : 0,
"url" : "/services/data/v38.0/support/knowledgeArticles/kA0xx000000006GCAQ",
"viewCount" : 0,
"viewScore" : 0.0
} ],
"currentPageUrl" :
"/services/data/v38.0/support/knowledgeArticles?channel=Pkb&pageSize=3&sort=ViewScore",
"nextPageUrl" : null,
"pageNumber" : 1
}

```

## Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. The Custom File Field is not supported because it returns a link to a binary stream. Use the language code format used in [Which Languages Does Salesforce Support?](#).

## Valid `channel` Values

- When using the options string `channel`, where the matching articles are visible, the following values are valid.
  - `App`—Visible in the internal Salesforce Knowledge application
  - `Pkb`—Visible in the public knowledge base
  - `Csp`—Visible in the Customer Portal

- `Prm`—Visible in the Partner Portal
- If `channel` isn't specified, the default value is determined by the type of user.
  - `Pkb` for a guest user
  - `Csp` for a Customer Portal user
  - `Prm` for a Partner Portal user
  - `App` for any other type of user
- If `channel` is specified, the specified value may be used to retrieve articles.
  - For guest, Customer Portal, and Partner Portal users, if the specified channel is other than the channel accessible to the user, an error is returned.
  - For all users other than guest, Customer Portal, and Partner Portal users, the specified channel value is used.

## Articles Details

Get all article fields, accessible to the user.

### Syntax

#### Available since release

38.0

#### Method

GET

#### Formats

JSON, XML

#### Authentication

OAuth accesstoken

#### Endpoint

[prefix]/support/knowledgeArticles/{articleId}

#### HTTP headers

**Accept:** Optional. Can be either `application/json` or `application/xml`.

**Accept-language:** Required. The article must be an active language in the user's organization

- If the language code isn't valid, an error message is returned: "The language code is not valid or not supported by Knowledge."
- If the language code is valid, but not supported by Knowledge, then an error message is returned: "Invalid language code. Check that the language is included in your Knowledge language settings."

#### Input:

string `channel`: Optional, defaults to user's context. For information on channel values, see [Valid channel Values](#).

- **App:** Visible in the internal Salesforce Knowledge application
- **Pkb:** Visible in the public knowledge base
- **Csp:** Visible in the Customer Portal
- **Prm:** Visible in the Partner Portal

boolean `updateViewstat`: Optional, defaults to true. If true, API updates the view count in the given channel as well as the total view count.

**Output:**

The detailed fields of the article, if the article is online and visible to the current user.

- **Article Detail**

Full detail of an article, with complete metadata and layout-driven fields used for display of an article. It includes all the same properties as an [Article Summary](#) representation.

```
{
  "id": Id, // articleId,
  "articleNumber": String,
  "title": String,
  "summary": String,
  "url": URL,
  "versionNumber": Int,
  "createdDate": Date, // in ISO8601 format
  "createdBy": User Summary on page 125,
  "lastModifiedDate": Date, // in ISO8601 format
  "lastModifiedBy": User Summary on page 125,
  "lastPublishedDate": Date, // in ISO8601 format
  "layoutItems": [ Article Field, ... ], // standard and custom fields visible
  to the user, sorted based on the layouts of the article type.
  "categories": [ Data Category Groups, ... ],
  "appUpVoteCount": Int,
  "cspUpVoteCount": Int,
  "prmUpVoteCount": Int,
  "pkbUpVoteCount": Int,
  "appDownVoteCount": Int,
  "cspDownVoteCount": Int,
  "prmDownVoteCount": Int,
  "pkbDownVoteCount": Int,
  "allViewCount": Int,
  "appViewCount": Int,
  "cspViewCount": Int,
  "prmViewCount": Int,
  "pkbViewCount": Int,
  "allViewScore": Double,
  "appViewScore": Double,
  "cspViewScore": Double,
  "prmViewScore": Double,
  "pkbViewScore": Double
}
```

- **User Summary**

```
{
  "id": String
  "isActive": boolean // true/false
  "userName": String // login name
  "firstName": String
  "lastName": String
  "email": String
  "url": String // to the chatter user detail url:
  /services/data/xx.x/chatter/users/{userId}, for guest user, it will return null.
}
```

- **Article Field**

An individual field of article information, which is listed in an [Article Detail](#) in the order required by the administrator's layout.

```
{
  "type": Enum,          // see the Notes
  "name": String,       // field name
  "label": String,      // label
  "value": String,
}
```

## Example

### Input

```
/services/data/v38.0/support/knowledgeArticles/kA0xx00000000LCAQ
HTTP Headers:
Content-Type: application/json; charset=UTF-8
Accept: application/json
Accept-Language: en-US
```

### Output

```
{
  "allViewCount" : 17,
  "allViewScore" : 100.0,
  "appDownVoteCount" : 0,
  "appUpVoteCount" : 0,
  "appViewCount" : 17,
  "appViewScore" : 100.0,
  "articleNumber" : "000001004",
  "categoryGroups" : [ ],
  "createdBy" : {
    "email" : "user@company.com",
    "firstName" : "Test",
    "id" : "005xx000001SvoMAAS",
    "isActive" : true,
    "lastName" : "User",
    "url" : "/services/data/v38.0/chatter/users/005xx000001SvoMAAS",
    "userName" : "admin@salesforce.org"
  },
  "createdDate" : "2016-06-21T21:10:54Z",
  "cspDownVoteCount" : 0,
  "cspUpVoteCount" : 0,
  "cspViewCount" : 0,
  "cspViewScore" : 0.0,
  "id" : "kA0xx00000000LCAQ",
  "lastModifiedBy" : {
    "email" : "user@company.com",
    "firstName" : "Test",
    "id" : "005xx000001SvoMAAS",
    "isActive" : true,
    "lastName" : "User",
    "url" : "/services/data/v38.0/chatter/users/005xx000001SvoMAAS",
    "userName" : "admin@salesforce.org"
  }
}
```

```

    },
    "lastModifiedDate" : "2016-06-21T21:11:02Z",
    "lastPublishedDate" : "2016-06-21T21:11:02Z",
    "layoutItems" : [ {
      "label" : "Out of Date",
      "name" : "IsOutOfDate",
      "type" : "CHECKBOX",
      "value" : "false"
    }, {
      "label" : "sample",
      "name" : "sample",
      "type" : "PICK_LIST",
      "value" : null
    }, {
      "label" : "Language",
      "name" : "Language",
      "type" : "PICK_LIST",
      "value" : "en_US"
    }, {
      "label" : "MyNumber",
      "name" : "MyNumber",
      "type" : "NUMBER",
      "value" : null
    }, {
      "label" : "My File",
      "name" : "My_File",
      "type" : "FILE",
      "value" : null
    } ],
    "pkbDownVoteCount" : 0,
    "pkbUpVoteCount" : 0,
    "pkbViewCount" : 0,
    "pkbViewScore" : 0.0,
    "prmDownVoteCount" : 0,
    "prmUpVoteCount" : 0,
    "prmViewCount" : 0,
    "prmViewScore" : 0.0,
    "summary" : "The number of characters required for complete coverage of all these languages' needs cannot fit in the 256-character code space of 8-bit character encodings, requiring at least a 16-bit fixed width encoding or multi-byte variable-length encodings.\r\n\r\nAlthough CJK encodings have common character sets, the encodings often used to represent them have been developed separately by different East Asian governments and software companies, and are mutually incompatible. Unicode has attempted, with some controversy, to unify the character sets in a process known as Han unification.\r\n\r\n\r\nCJK character encodings should consist minimally of Han characters p",
    "title" : "Test Images",
    "url" : "/services/data/v38.0/support/knowledgeArticles/kA0xx000000000LCAQ",
    "versionNumber" : 7
  }

```

## Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. The Custom File Field is not supported because it returns a link to a binary stream. Use the language code format used in [Which Languages Does Salesforce Support?](#)

A lookup custom field is visible to guest users depending on the lookup entity type. For example, User is visible, but Case and Account are not visible. Following standard fields are not visible to a guest user, even if they are in the layout:

- archivedBy
- isLatestVersion
- translationCompletedDate
- translationImportedDate
- translationExportedDate
- versionNumber
- visibleInInternalApp
- visibleInPKB
- visibleToCustomer
- visibleToPartner

## Valid `channel` Values

- When using the options string `channel`, where the matching articles are visible, the following values are valid.
  - `App`—Visible in the internal Salesforce Knowledge application
  - `Pkb`—Visible in the public knowledge base
  - `Csp`—Visible in the Customer Portal
  - `Ptm`—Visible in the Partner Portal
- If `channel` isn't specified, the default value is determined by the type of user.
  - `Pkb` for a guest user
  - `Csp` for a Customer Portal user
  - `Ptm` for a Partner Portal user
  - `App` for any other type of user
- If `channel` is specified, the specified value may be used to retrieve articles.
  - For guest, Customer Portal, and Partner Portal users, if the specified channel is other than the channel accessible to the user, an error is returned.
  - For all users other than guest, Customer Portal, and Partner Portal users, the specified channel value is used.

# CHAPTER 4 Salesforce Knowledge Metadata API

## In this chapter ...

- [ArticleType](#)
- [KnowledgeSettings](#)
- [DataCategoryGroup](#)
- [SearchSettings](#)
- [SearchLayouts](#)
- [SynonymDictionary](#)
- [ExternalDataSource](#)

You can manage articles, page layouts, and data categories with the Metadata API.

Use Metadata API to retrieve, deploy, create, update or delete customization information, such as custom object definitions and page layouts, for your organization. This API is intended for managing customizations and for building tools that can manage the metadata model, not the data itself.

For generic Metadata API information see the [Metadata API Developer Guide](#).

## ArticleType

---

Represents the metadata associated with an article type.

All articles in Salesforce Knowledge are assigned to an *article type*. An article's type determines the type of content it contains, its appearance, and which users can access it. For example, a simple FAQ article type might have two custom fields, `Question` and `Answer`, where article managers enter data when creating or updating FAQ articles. A more complex article type may require dozens of fields organized into several sections. Using layouts and templates, administrators can structure the article type in the most effective way for its particular content. User access to article types is controlled by permissions. For each article type, an administrator can grant "Create," "Read," "Edit," or "Delete" permissions to users. For example, the article manager might want to allow internal users to read, create, and edit FAQ article types, but let partner users only read FAQs. See "Knowledge Article Types" in the Salesforce online help and "Articles" in the [SOAP API Developer's Guide](#).

## Declarative Metadata File Suffix and Directory Location

An ArticleType is defined as a custom object and is stored in the `objects` folder. ArticleTypes have a suffix `__kav` (instead of `__c` for custom objects). ArticleType field names have a suffix of `__c` like other custom objects, and must be dot-qualified with the name of the article type to which they belong. This is shown in the following sample `package.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
  <Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>articlefilemetadata</fullName>
    <apiAccessLevel>Unrestricted</apiAccessLevel>

    <types>
      <members>newarticle__kav.description__c</members>
      <name>CustomField</name>
    </types>
    <types>
      <members>newarticle__kav</members>
      <name>CustomObject</name>
    </types>

  </Package>
```

## Version

ArticleTypes are available in API version 19.0 and later.

## Fields

Field Name	Field Type	Description
articleTypeChannelDisplay	articleTypeChannelDisplay	Represents the article-type templates used to display an article in the various channels. See "Article Type Templates" in the Salesforce online help.



Field Name	Field Type	Description
deploymentStatus	DeploymentStatus (enumeration of type string)	A string which represents the deployment status of a custom object or field. Valid values are: <ul style="list-style-type: none"> <li>InDevelopment</li> <li>Deployed</li> </ul>
description	string	A description of the article type. Maximum of 1000 characters.
fields	CustomField[]	Represents one or more fields in the article type.
gender	Gender	Indicates the gender of the noun that represents the object. This is used for languages where words need different treatment depending on their gender.
label	string	Label that represents the object throughout the Salesforce user interface.
pluralLabel	string	Plural version of the label value.
startsWith	StartsWith (enumeration of type string)	Indicates whether the noun starts with a vowel, consonant, or is a special character. This is used for languages where words need different treatment depending on the first character. Valid values are listed in StartsWith.

## articleTypeChannelDisplay

Determines the article-type templates that are used to display an article in its channels. Unless otherwise noted, all fields are createable, filterable, and nillable.

Field Name	Field Type	Description
articleTypeTemplates	<a href="#">articleTypeTemplates</a>	Indicates which article-type template applies in the specified channel.

## articleTypeTemplates

Sets the article-type template for a specific channel. If not specified, the default article-type template applies.

Field Name	Field Type	Description
channel	string	Specifies the channel where the article-type template applies: <ul style="list-style-type: none"> <li>AllChannels: all the available channels.</li> <li>App: the Articles tab in Salesforce Knowledge.</li> <li>Pkb: the public knowledge base.</li> <li>Csp: the Customer Portal.</li> <li>Prm: the partner portal.</li> </ul>
page	string	Represents the name of the custom Visualforce page used as a custom article-type template. Use this field when you select Page in the template field.

Field Name	Field Type	Description
template	string	<p>Indicates the article-type template used for the specified channel:</p> <ul style="list-style-type: none"> <li>• <b>Page</b>: custom Visualforce page. When specifying this value, you must also set the <code>page</code> field with the Visualforce page name.</li> <li>• <b>Tab</b>: display the sections you defined in the layout as tabs.</li> <li>• <b>Toc</b>: display the sections you defined in the layout as table of content.</li> </ul>

## Declarative Metadata Sample Definitions

A sample article type definition follows:

```
<?xml version="1.0" encoding="UTF-8"?>
  <CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <articleTypeChannelDisplay>
      <articleTypeTemplates>
        <channel>App</channel>
        <template>Tab</template>
      </articleTypeTemplates>
      <articleTypeTemplates>
        <channel>Prm</channel>
        <template>Tab</template>
      </articleTypeTemplates>
      <articleTypeTemplates>
        <channel>Csp</channel>
        <template>Tab</template>
      </articleTypeTemplates>
      <articleTypeTemplates>
        <channel>Pkb</channel>
        <template>Toc</template>
      </articleTypeTemplates>
    </articleTypeChannelDisplay>
    <deploymentStatus>Deployed</deploymentStatus>
    <description>Article type with custom fields</description>
    <fields>
      <fullName>description__c</fullName>
      <label>Description</label>
      <length>48</length>
      <type>Text</type>
    </fields>
    <label>newarticle</label>
    <pluralLabel>newarticles</pluralLabel>
  </CustomObject>
```

### IN THIS SECTION:

#### [ArticleType Layout](#)

Represents the metadata associated with an article type page layout. Article type layouts determine which fields users can view and edit when entering data for an article, they also determine which sections appear when users view articles.

[ChannelLayout](#)

Represents the metadata associated with an communication channel layout. Communication channel layouts allow admins to share article contents inline into communication channels (eg. email publisher, portal publisher, social publisher etc.). Admins can create a list of fields of an article type that they want to share for each communication channels and customize its order.

[ArticleType CustomField](#)

Represents the metadata associated with an article type custom field. Use this metadata type to create, update, or delete article type custom field definitions.

## ArticleType Layout

Represents the metadata associated with an article type page layout. Article type layouts determine which fields users can view and edit when entering data for an article, they also determine which sections appear when users view articles.

The format of the article, for example whether layout sections display as subtabs or as a single page with links, is defined by the [article-type template](#). Each article type has only one layout, but you can choose a different template for each of the article type's four channels. For more information, see "Knowledge Article Types" in the Salesforce online help and "Articles" in the [SOAP API Developer's Guide](#)

## File Suffix and Directory Location

ArticleType layouts are stored in the `layouts` directory of the corresponding package directory. The prefix must match with the article type API name. The extension is `.layout`.

## Version

ArticleType layouts are available in API version 19.0 and later.

## Fields

Field Name	Field Type	Description
<code>layoutSections</code>	<a href="#">LayoutSection[]</a>	The main sections of the layout containing the article fields. The order here determines the layout order.

## LayoutSection

LayoutSection represents a section of an ArticleType layout.

Field Name	Field Type	Description
<code>customLabel</code>	boolean	Indicates if this section's label is custom or standard (built-in). Custom labels can be any text, but must be translated. Standard labels have a predefined set of valid values, for example 'System Information', which are automatically translated.
<code>label</code>	string	The label; either standard or custom, based on the <code>customLabel</code> flag.
<code>layoutColumns</code>	<a href="#">LayoutColumn[]</a>	The columns of the layout, depending on the style. Salesforce Knowledge only supports one column in article type layouts.

Field Name	Field Type	Description
style	LayoutSectionStyle (enumeration of type string)	The style of the layout. Salesforce Knowledge only supports the value OneColumn which displays a one column page.

## LayoutColumn

LayoutColumn represents the items in a column within a layout section.

Field Name	Field Type	Description
layoutItems	LayoutItem[]	The individual items within a column (ordered from top to bottom).

## LayoutItem

LayoutItem represents the valid values that define a layout item.

Field Name	Field Type	Description
field	string	The field name reference, for example MyField__c.

## Declarative Metadata Sample Definition

The following is the definition of an ArticleType page layout:

```
<?xml version="1.0" encoding="UTF-8"?>
  <Layout xmlns="http://soap.sforce.com/2006/04/metadata">
    <layoutSections>
      <customLabel>true</customLabel>
      <label>Description</label>
      <layoutColumns>
        <layoutItems>
          <field>description__c</field>
        </layoutItems>
        <layoutItems>
          <field>dateTime__c</field>
        </layoutItems>
      </layoutColumns>
      <style>OneColumn</style>
    </layoutSections>
    <layoutSections>
      <label>Data Sheet</label>
      <layoutColumns>
        <layoutItems>
          <field>file__c</field>
        </layoutItems>
      </layoutColumns>
      <style>OneColumn</style>
    </layoutSections>
  </Layout>
```

## ChannelLayout

Represents the metadata associated with an communication channel layout. Communication channel layouts allow admins to share article contents inline into communication channels (eg. email publisher, portal publisher, social publisher etc.). Admins can create a list of fields of an article type that they want to share for each communication channels and customize its order.

### File Suffix and Directory Location

Channel layout components have the suffix `.channelLayout` and are stored in the `channelLayouts` folder of the corresponding package directory. The prefix must match with the article type API name.

### Version

Channel layout components are available in API version 32.0 and later.

### Fields

Field Name	Field Type	Description
<code>label</code>	string	The label
<code>layoutItems</code>	layoutItem	The article fields contained in the layout. The order here determines the field order.
<code>enabledChannels</code>	enabledChannel	The channels where this layout applies.

### layoutItem

Field Name	Field Type	Description
<code>field</code>	string	Name of the field. The format is <code>&lt;ArticleType name&gt;.&lt;Field name&gt;</code>
<code>enabledChannel</code>	enum	Enum name of the communication channel (eg. email, portal etc)

### Declarative Metadata Sample Definition

The following is an example of a Channel Layout component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ChannelLayout xmlns="http://soap.sforce.com/2006/04/metadata">
  <label>layoutaaaa</label>
  <layoutItems>
    <field>ArticleTypeA.FieldA</field>
  </layoutItems>
  <layoutItems>
    <field>ArticleTypeA.FieldC</field>
  </layoutItems>
  <enabledChannels>email</enabledChannels>
  <enabledChannels>portal</enabledChannels>
</ChannelLayout>
```

The following is an example `package.xml` that references the previous definition.

## ArticleType CustomField

Represents the metadata associated with an article type custom field. Use this metadata type to create, update, or delete article type custom field definitions.

This type extends the Metadata metadata type and inherits its `fullName` field.

Always specify the full name whenever you create or update a custom field. For example, a custom field on a custom object:

```
MyArticleType__kav.MyCustomField__c
```

## Declarative Metadata File Suffix and Directory Location

Custom fields are defined as part of the article type. ArticleType field names have a suffix of `__c` like other custom objects, and must be dot-qualified with the name of the article type to which they belong. See [ArticleType](#) for more information.

## Retrieving Custom Fields on Custom or Standard Objects

When you retrieve a custom or standard object, you return everything associated with the object. However, you can also retrieve only the custom fields for an object by explicitly naming the object and fields in `package.xml`. The following definition in `package.xml` retrieves the files `objects/MyCustomObject__c.object`, `objects/Account.object__c.object`, and `objects/MyArticleType__kav.object`, each containing one custom field definition.


```
<types>
  <members>MyCustomObject__c.MyCustomField__c</members>
  <members>Account.MyCustomAccountField__c</members>
  <members>MyArticleType__kav.MyOtherCustomField__c</members>
  <name>CustomField</name>
</types>
```

## Version

ArticleTypes custom fields are available in API version 19.0 and later.

## Fields for ArticleType

Unless otherwise noted, all fields are createable, filterable, and nillable.

 **Note:** If you create a knowledge validation rule, the errors always display at the top of the page, even if you add it beside the field. Therefore, write the errors descriptively so authors know how to satisfy the validation rule. For example, identify which field is causing the error. The Salesforce Classic user interface does not support field level error messages for articles.

Field Name	Field Type	Description
<code>defaultValue</code>	string	If specified, represents the default value of the field.

Field Name	Field Type	Description
<code>deleteConstraint</code>	DeleteConstraint (enumeration of type string)	Provides deletion options for lookup relationships. Valid values are:  <b>SetNull</b> This is the default. If the lookup record is deleted, the lookup field is cleared.  <b>Restrict</b> Prevents the record from being deleted if it's in a lookup relationship.  <b>Cascade</b> Deletes the lookup record as well as associated lookup fields.  For more information on lookup relationships, see "Object Relationships" in the Salesforce Help.
<code>description</code>	string	Description of the field.
<code>formula</code>	string	If specified, represents a formula on the field.
<code>formulaTreatBlankAs</code>	TreatBlanksAs (enumeration of type string)	Indicates how to treat blanks in a formula. Valid values are <code>BlankAsBlank</code> and <code>BlankAsZero</code> .
<code>fullName</code>	string	Inherited from Metadata, this field is not defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See <code>create()</code> to see an example of this field specified for a call.  This value cannot be <code>null</code> .
<code>inlineHelpText</code>	string	Represents the content of field-level help. For more information, see "Define Field-Level Help" in the Salesforce Help.
<code>label</code>	string	Label for the field. You cannot update the label for standard fields in Article Type such as Title, UrlName, Summary, etc.
<code>length</code>	int	Length of the field.
<code>picklist</code>	Picklist (Including Dependent Picklist)	<b>(Deprecated.</b> Use this field in API version 37.0 and earlier only. In later versions, use <code>valueSet</code> instead.) If specified, the field is a picklist, and this field enumerates the picklist values and labels.
<code>referenceTo</code>	string	If specified, indicates a reference this field has to another object.
<code>relationshipLabel</code>	string	Label for the relationship.
<code>relationshipName</code>	string	If specified, indicates the value for one-to-many relationships. For example, in the object <code>MyObject</code> that had a relationship to <code>YourObject</code> , the relationship name might be <code>YourObjects</code> .

Field Name	Field Type	Description
required	boolean	Indicates whether the field requires a value on creation ( <code>true</code> ) or not ( <code>false</code> ).
type	FieldType	Required. Indicates the field type for the field. Valid values are: <ul style="list-style-type: none"> <li>• Checkbox available in version 30.0 and later</li> <li>• Currency</li> <li>• ArticleCurrency</li> <li>• Date</li> <li>• DateTime</li> <li>• Email</li> <li>• File</li> <li>• Formula</li> <li>• Html</li> <li>• Lookup</li> <li>• Number</li> <li>• Percent</li> <li>• Phone</li> <li>• Picklist</li> <li>• DependentPicklist</li> <li>• MultiselectPicklist</li> <li>• Text</li> <li>• TextArea</li> <li>• LongTextArea</li> <li>• URL</li> </ul>
visibleLines	int	Indicates the number of lines displayed for the field.

## Declarative Metadata Sample Definition

```
<?xml version="1.0" encoding="UTF-8"?>
  <CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    ....
    <fields>
      <fullName>Comments__c</fullName>
      <description>add your comments about this object here</description>
      <label>Comments</label>
      <length>32000</length>
      <type>LongTextArea</type>
      <visibleLines>30</visibleLines>
    </fields>
    ....
  </CustomObject>
```



# KnowledgeSettings

---

Represents the metadata used to manage settings for Salesforce Knowledge.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the “Settings” name. See Settings for more details.

## File Suffix and Directory Location

KnowledgeSettings values are stored in a single file named `Knowledge.settings` in the `settings` directory. The `.settings` files are different from other named components because there is only one settings file for each settings component.

## Version

KnowledgeSettings is available in API version 27.0 and later.

## Fields

Field Name	Field Type	Description
<code>answers</code>	<a href="#">KnowledgeAnswerSettings</a>	Represents the metadata used to manage settings for Salesforce Knowledge and Answers.
<code>cases</code>	<a href="#">KnowledgeCaseSettings</a>	Represents the metadata used to manage settings for Salesforce Knowledge and Cases.
<code>defaultLanguage</code>	string	Required. The default language for Salesforce Knowledge. Use the abbreviation for the language, for example, <code>en_US</code> for United States English.
<code>languages</code>	<a href="#">KnowledgeLanguageSettings</a>	A list of languages enabled for Salesforce Knowledge.
<code>enableChatterQuestionKBDeflection</code>	boolean	Indicates whether tracking for case deflection via Chatter is enabled ( <code>true</code> ) or not ( <code>false</code> ).
<code>enableCreateEditOnArticlesTab</code>	boolean	Indicates whether users can create and edit articles on the articles tab ( <code>true</code> ) or not ( <code>false</code> ).
<code>enableExternalMediaContent</code>	boolean	Indicates whether connecting to external media is enabled ( <code>true</code> ) or not ( <code>false</code> ).
<code>enableKnowledge</code>	boolean	Indicates whether Salesforce Knowledge is enabled ( <code>true</code> ) or not ( <code>false</code> ).
<code>showArticleSummariesCustomerPortal</code>	boolean	Indicates whether article summaries appear in the Customer Portal ( <code>true</code> ) or not ( <code>false</code> ).

Field Name	Field Type	Description
showArticleSummariesInternalApp	boolean	Indicates whether article summaries appear in the internal knowledge base ( <code>true</code> ) or not ( <code>false</code> ).
showArticleSummariesPartnerPortal	boolean	Indicates whether article summaries appear in the partner portal ( <code>true</code> ) or not ( <code>false</code> ).
showValidationStatusField	boolean	Indicates whether validation status appears on articles ( <code>true</code> ) or not ( <code>false</code> ).
suggestedArticles	<a href="#">KnowledgeSuggestedArticlesSettings</a>	Represents the metadata used to manage settings for the case fields used to suggest articles for cases. Available in API version 37.0 and later.

## KnowledgeAnswerSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Answers.

Field Name	Field Type	Description
assignTo	string	Specifies the username an article is assigned to from Answers.
defaultArticleType	string	The default article type for articles created from Answers. Uses the API name of the article type.
enableArticleCreation	boolean	Indicates whether users can create articles from Answers ( <code>true</code> ) or not ( <code>false</code> ).

## KnowledgeCaseField

Represents the name of the case field used to suggest articles for the case. Available in API version 37.0 and later.

Field Name	Field Type	Description
name	string	Specifies the name of the case field used to suggest articles for the case.

## KnowledgeCaseFieldsSettings

Represents a list of the case fields used to suggest articles for the case. Available in API version 37.0 and later.

Field Name	Field Type	Description
field	<a href="#">KnowledgeCaseField[]</a>	Specifies the names of the case fields used to suggest articles for the case.

## KnowledgeWorkOrderField

Represents the name of the work order field used to suggest articles for the work order. Available in API version 39.0 and later.

Field Name	Field Type	Description
name	string	Specifies the name of the work order field used to suggest articles for the work order.

## KnowledgeWorkOrderFieldsSettings

Represents a list of the work order fields used to suggest articles for the work order. Available in API version 39.0 and later.

Field Name	Field Type	Description
field	<a href="#">KnowledgeWorkOrderField[]</a>	Specifies the names of the work order fields used to suggest articles for the work order.

## KnowledgeWorkOrderLineItemField

Represents the name of the work order line item field used to suggest articles for the work order line item. Available in API version 39.0 and later.

Field Name	Field Type	Description
name	string	Specifies the name of the work order line item field used to suggest articles for the work order line item.

## KnowledgeWorkOrderLineItemFieldsSettings

Represents a list of the work order line item fields used to suggest articles for the work order line item. Available in API version 39.0 and later.

Field Name	Field Type	Description
field	<a href="#">KnowledgeWorkOrderLineItemField[]</a>	Specifies the names of the work order line item fields used to suggest articles for the work order line item.

## KnowledgeSuggestedArticlesSettings

Represents the metadata used to manage settings for the articles suggested for cases, work orders, and work order line items.

Field Name	Field Type	Description
caseFields	<a href="#">KnowledgeCaseFieldsSettings</a>	Represents a list of the case fields used to suggest articles for the case.

Field Name	Field Type	Description
useSuggestedArticlesForCase	boolean	Indicates whether case content is used to suggest articles for cases ( <code>true</code> ) or not ( <code>false</code> ).
workOrderFields	<a href="#">KnowledgeWorkOrderFieldsSettings</a>	Represents a list of the work order fields used to suggest articles for the work order.
workOrderLineItemFields	<a href="#">KnowledgeWorkOrderLineItemFieldsSettings</a>	Represents a list of the work order line item fields used to suggest articles for the work order line item.

## KnowledgeCaseSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Cases.

Field Name	Field Type	Description
articlePDFCreationProfile	string	The profile used to create a PDF of an article from Cases.
articlePublicSharingSites	<a href="#">KnowledgeSitesSettings</a>	Represents the metadata used to manage settings for Salesforce Knowledge and Sites.
articlePublicSharingCommunities	<a href="#">KnowledgeSitesSettings</a>	Represents the metadata used to manage settings for Salesforce Knowledge and Communities.
articlePublicSharingSitesChatterAnswers	<a href="#">KnowledgeSitesSettings</a>	Represents the metadata used to manage settings for Salesforce Knowledge and Sites with Chatter Answers.
assignTo	string	Specifies the username an article is assigned to from Cases.
customizationClass	string	Specifies the Apex class used for customization.
defaultContributionArticleType	string	The default article type for articles created from Cases.
editor	KnowledgeCaseEditor (enumeration of type string)	Indicates the rich text editor type. Valid values are: <ul style="list-style-type: none"> <li>• <code>simple</code></li> <li>• <code>standard</code></li> </ul>
enableArticleCreation	boolean	Indicates whether users can create articles from Cases ( <code>true</code> ) or not ( <code>false</code> ). Controls whether other fields on KnowledgeCaseSettings can be set.
enableArticlePublicSharingSites	boolean	Indicates whether articles can be shared via a public site (URL) from Cases ( <code>true</code> ) or not ( <code>false</code> ).
useProfileForPDFCreation	boolean	Indicates whether a profile is used to create a PDF of an article from Cases ( <code>true</code> ) or not ( <code>false</code> ).

## KnowledgeSitesSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Sites.

Field Name	Field Type	Description
site	string[]	Specifies the site used for Salesforce Knowledge and Sites.

## KnowledgeLanguageSettings

A list of languages enabled for Salesforce Knowledge. KnowledgeLanguageSettings is available in API version 28.0 and later.

Field Name	Field Type	Description
language	<a href="#">KnowledgeLanguage</a>	Represents the metadata used to manage settings for the languages enabled for Salesforce Knowledge.

## KnowledgeLanguage

Represents the metadata used to manage settings for the languages enabled for Salesforce Knowledge. KnowledgeLanguage is available in API version 28.0 and later.

Field Name	Field Type	Description
active	boolean	Indicates whether the language is enabled ( <code>true</code> ) or not ( <code>false</code> ).
defaultAssignee	string	The default assignee for articles in the language.
defaultAssigneeType	KnowledgeLanguageLookupValueType (enumeration of type string)	Indicates the default assignee type. Valid values are: <ul style="list-style-type: none"> <li>• <code>user</code></li> <li>• <code>queue</code></li> </ul>
defaultReviewer	string	The default reviewer for articles in the language.
defaultReviewerType	KnowledgeLanguageLookupValueType (enumeration of type string)	Indicates the default reviewer type. Valid values are: <ul style="list-style-type: none"> <li>• <code>user</code></li> <li>• <code>queue</code></li> </ul>
name	string	The code for the language name, for example: English is <code>en</code> . See “What languages does Salesforce support?” in the Salesforce online help for a list of supported languages and their codes.

## Declarative Metadata Sample Definition

This is a sample Knowledge settings file.


```
<?xml version="1.0" encoding="UTF-8"?>
<KnowledgeSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <answers>
    <enableArticleCreation>false</enableArticleCreation>
  </answers>
  <cases>
    <articlePDFCreationProfile>partner portal knowledge
profile</articlePDFCreationProfile>
    <articlePublicSharingSites>
      <site>KnowledgeSite</site>
      <site>PKB2Site</site>
      <site>ChatterAnswersSite</site>
    </articlePublicSharingSites>
    <articlePublicSharingSitesChatterAnswers>
      <site>ChatterAnswersSite</site>
    </articlePublicSharingSitesChatterAnswers>
    <assignTo>testall@kb.org</assignTo>
    <defaultContributionArticleType>Support</defaultContributionArticleType>
    <editor>simple</editor>
    <enableArticleCreation>true</enableArticleCreation>
    <enableArticlePublicSharingSites>true</enableArticlePublicSharingSites>
    <useProfileForPDFCreation>true</useProfileForPDFCreation>
  </cases>
  <defaultLanguage>ja</defaultLanguage>
  <enableCreateEditOnArticlesTab>true</enableCreateEditOnArticlesTab>
  <enableExternalMediaContent>true</enableExternalMediaContent>
  <enableKnowledge>true</enableKnowledge>
  <showArticleSummariesCustomerPortal>true</showArticleSummariesCustomerPortal>
  <showArticleSummariesInternalApp>true</showArticleSummariesInternalApp>
  <showArticleSummariesPartnerPortal>true</showArticleSummariesPartnerPortal>
  <showValidationStatusField>true</showValidationStatusField>
  <suggestedArticles>
    <caseFields>
      <field>
        <name>Subject</name>
      </field>
      <field>
        <name>SuppliedEmail</name>
      </field>
    </caseFields>
    <useSuggestedArticlesForCase>true</useSuggestedArticlesForCase>
  </suggestedArticles>
</KnowledgeSettings>
```

## DataCategoryGroup

---

Represents a data category group.

This type extends the Metadata metadata type and inherits its `fullName` field.

 **Warning:** Using Metadata API to deploy category changes from one organization to another permanently removes categories and record categorizations that are not specified in your XML file. Salesforce recommends that you manually create data categories and record associations in an organization from Setup by entering *Data Categories* in the *Quick Find* box, then selecting **Data Categories** rather than deploying changes from a sandbox to a production organization. For more information, see [Usage](#).

Data category groups are provided to:

- Classify and filter data.
- Share data among users.

Every data category group contains items or data categories that can be organized hierarchically.

The example below shows the *Geography* data category group and its data categories.

```

Geography
  Worldwide
    North America
      United States of America
      Canada
      Mexico
    Europe
    Asia
  
```

 **Note:** See "Data Categories in Salesforce.com" in the Salesforce online help for more information on data category groups, data categories, parent and sub categories.

## File Suffix and Directory Location

The file suffix is `.datacategorygroup`. There is one file for each data category group stored in the `datacategorygroups` folder in the corresponding package directory.

## Version

Data category groups are available in API version 18.0 and later.

## Fields



This metadata type contains the following fields:

Field Name	Field Type	Description
<code>active</code>	boolean	Required. The status of the category group. Indicates whether this category group is active, ( <code>true</code> ), or not active ( <code>false</code> ).
<code>dataCategory</code>	<a href="#">DataCategory</a>	Required. The top-level category within the data category group.
<code>description</code>	string	The description of the data category group.
<code>fullName</code>	string	Required. The unique name of the data category group. When creating a data category group, the <code>fullName</code> field and the file name (without its suffix) must match. The <code>fullName</code> can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not

Field Name	Field Type	Description
		include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
label	string	Required. Label that represents the object in Salesforce.
objectUsage	<a href="#">ObjectUsage</a>	The objects that are associated with the data category group.

## DataCategory

Represents an item (or data category) in the data category group. A data category can recursively contain a list of other data categories.


Field Name	Field Type	Description
dataCategory	<a href="#">DataCategory[]</a>	A recursive list of sub data categories. For example, a list of countries within a continent. You can create up to 100 categories in a data category group and have up to 5 levels in a data category group hierarchy.
label	string	Required. Label for the data category throughout the Salesforce user interface.
name	string	Required. The developer name of the data category used as a unique identifier for API access. The name can only contain characters, letters, and the underscore ( <code>_</code> ) character, must start with a letter, and cannot end with an underscore or contain two consecutive underscore characters. <ul style="list-style-type: none"> <li> <b>Important:</b> The value for this field is defined once and cannot be changed later.</li> <li> <b>Warning:</b> If you deploy a category group that already exists in an organization, any category that is not defined in the XML file is permanently removed from your organization. For more information see <a href="#">Usage</a>.</li> </ul>

## ObjectUsage

Represents the objects that can be associated with the data category group. This association allows the object to be classified and filtered using the data categories.

Field Name	Field Type	Description
object	string[]	A list of the object names that can be associated with the data category group. Valid values are: <ul style="list-style-type: none"> <li>• <code>KnowledgeArticleVersion</code>—to associate articles. See "Modify Default Category Group Assignments for Articles" in the Salesforce online help for more information on data category groups association to articles.</li> </ul>



Field Name	Field Type	Description
		<ul style="list-style-type: none"> <li>• <b>Question</b>—to associate questions. You can associate the <b>Question</b> object with at most one category group. See "Assigning Data Categories to Answers" in the Salesforce online help for more information on data category groups association to questions.</li> </ul> <p> <b>Warning:</b> If you deploy a category group that already exists in an organization, any object association that is not defined in the XML file is permanently removed from your organization. Ensure that your XML file specifies all the records associated with your category group in the organization. For more information see <a href="#">Usage</a>.</p>

## Declarative Metadata Sample Definition

This sample is the definition of the **Geography** data category group and its data categories:

```
<?xml version="1.0" encoding="UTF-8"?>
<DataCategoryGroup xmlns="http://soap.sforce.com/2006/04/metadata">
  <label>Geography</label>
  <description>Geography structure of service center locations</description>
  <fullName>geo</fullName>

  <dataCategory> <name>WW</name> <label>Worldwide</label>
    <dataCategory> <name>AMER</name> <label>North America</label>
      <dataCategory>
        <name>USA</name>
        <label>United States of America</label>
      </dataCategory>
      <dataCategory>
        <name>CAN</name>
        <label>Canada</label>
      </dataCategory>
      <dataCategory>
        <name>MEX</name>
        <label>Mexico</label>
      </dataCategory>
    </dataCategory>
  <dataCategory> <name>EMEA</name> <label>Europe, Middle East, Africa</label>
    <dataCategory>
      <name>FR</name>
      <label>France</label>
    </dataCategory>
    <dataCategory>
      <name>SP</name>
      <label>Spain</label>
    </dataCategory>
    <dataCategory>
      <name>UK</name>
      <label>United-Kingdom</label>
    </dataCategory>
  </dataCategory>
```

```

    </dataCategory>
  <dataCategory>
    <name>APAC</name>
    <label>Asia</label>
  </dataCategory>
</dataCategory>


<objectUsage>
  <object>KnowledgeArticleVersion </object>
</objectUsage>
</DataCategoryGroup>


```

## Usage

When you deploy a category group XML file, Metadata API checks whether the category group exists in the target organization. If the category group does not exist, it is created. If the category group already exists, then Metadata API:

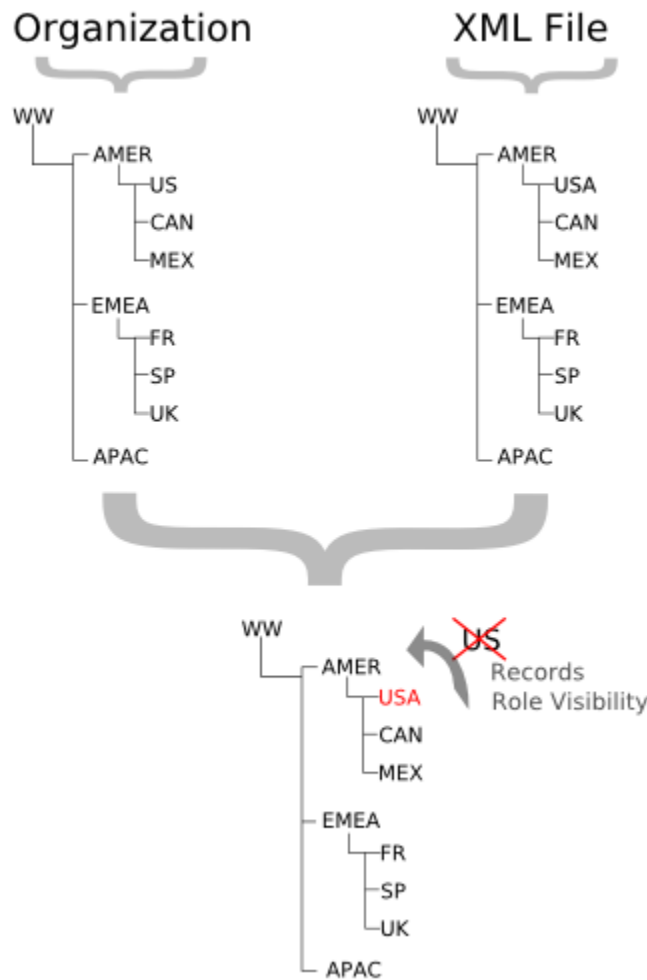
- Adds any new category or object defined in the XML file.
- Deletes any category that is not defined in the XML file. Records associated with the deleted categories are re-associated with the parent category.
- Deletes any object association that is not defined in the XML file.
- Moves any category if its hierarchical position differs from the position specified in the XML file.

 **Note:** When a category moves to a new parent category, users that have no visibility on the new parent category lose their visibility to the repositioned category.

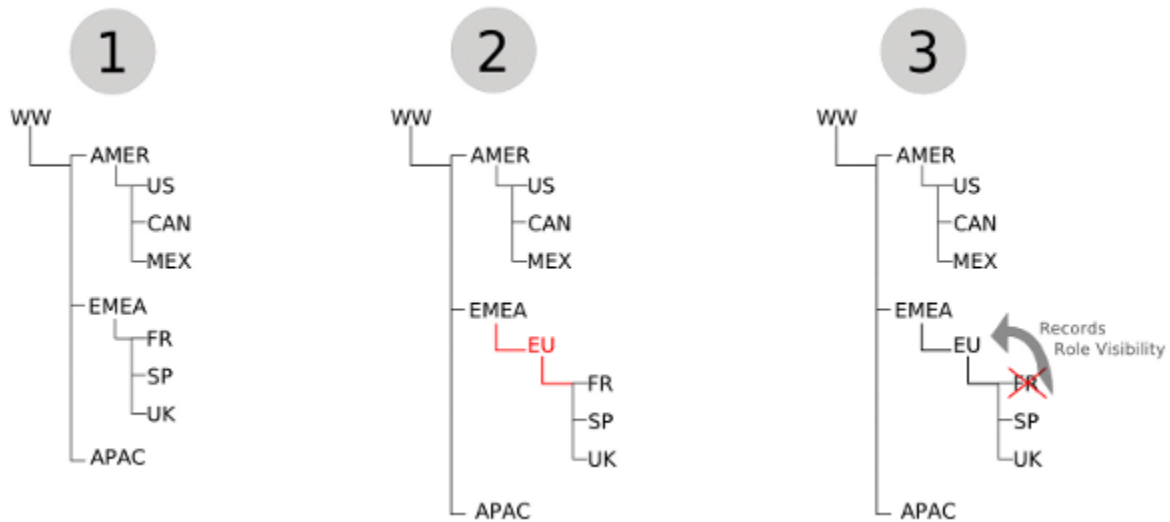
 **Note:** For more information about category deletion, category repositioning and its impact on record categorization and visibility see "Delete a Data Category" and "Modify and Arrange Data Categories" in the Salesforce online help.

Using Metadata API to deploy category changes from one organization to another permanently removes categories and record categorizations that are not specified in your XML file. Salesforce recommends that you manually create data categories and record associations in an organization from Setup by entering *Data Categories* in the *Quick Find* box, then selecting **Data Categories** rather than deploying changes from a sandbox to a production organization.

The following example illustrates what happens if you deploy an XML representation of a *Geography* data category group hierarchy to an organization that already has this data category group defined. Note that the organization contains a *US* category, while the XML file includes a *USA* category in the same hierarchical position. The Metadata API deployment process deletes the *US* category from the organization and moves associations for any records from *US* to the parent *AMER* category. It also adds the *USA* category under *AMER*. Note that all records that were previously categorized with *US* are now associated with the *AMER* category.



The next example illustrates what can happen when you delete or move a category in a data category group and deploy its XML representation from a sandbox to a production organization that already has this data category group defined. Hierarchy 1 shows the initial data category group in the sandbox organization. In hierarchy 2, we add an `EU` category under `EMEA` and move `FR`, `SP` and `UK` below `EU`. In hierarchy 3, we delete `FR` and associate its records with its new parent, `EU`. Finally, we deploy the changes from the sandbox to the production organization.



Metadata API has no concept of the order of the changes made to the sandbox organization. It just deploys the changes from one organization to another. During the deployment, it first notices the deletion of the `FR` category and removes it from the production organization. Consequently, it moves associations for any records from `FR` to its parent on the production organization, `EMEA`. Metadata API then adds the `EU` category and moves `SP` and `UK` below it. Although the category group hierarchy looks the same in both organizations, record categorization in production is different from the sandbox organization. The records that were originally associated with `FR` in hierarchy 1 are associated with `EU` in the sandbox organization, but are associated with `EMEA` in the production organization.

## SearchSettings

Represents an org's search settings.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the "Settings" name. See Settings for more details.

## File Suffix and Directory Location

SearchSettings values are stored in a single file named `Search.settings` in the `settings` folder. The `.settings` files are different from other named components because there is only one settings file for each settings component.

## Version

SearchSettings is available in API version 37.0 and later.

## Fields

Field Name	Field Type	Description
<code>documentContentSearchEnabled</code>	boolean	Indicates if a full-text document search is performed.
<code>optimizeSearchForCJKEnabled</code>	boolean	Indicates whether the search is optimized for the Japanese, Chinese, and Korean languages. This setting affects sidebar

Field Name	Field Type	Description
		search and the account search for <b>Find Duplicates</b> on a lead record in sidebar search and global search. Enable this option if users are searching mostly in Japanese, Chinese, or Korean, and if the text in searchable fields is mostly in those languages.
recentlyViewedUsersForBlankLookupEnabled	boolean	Indicates whether the list of records that are returned from a user autocomplete lookup and from a blank user lookup is taken from the user's recently viewed user records. Otherwise this setting is <code>false</code> if the lookup shows a list of recently accessed user records from across your org ( <code>false</code> ). Only applies to User object blank lookup searches.
searchSettingsByObject	<a href="#">SearchSettingsByObject</a>	Represents a list of search settings for each object.
sidebarAutoCompleteEnabled	boolean	Indicates if autocomplete is enabled for sidebar search. Autocomplete is when users start typing search terms and sidebar search displays a matching list of recently viewed records.
sidebarDropDownListEnabled	boolean	Indicates if a drop-down list appears in the sidebar search section. From this list, users can select to search within tags, within a specific object, or across all objects.
sidebarLimitToItemsIOwnCheckboxEnabled	boolean	Indicates if the <b>Limit to Items I Own</b> checkbox appears. The checkbox allows your users to include only records for which they are the record owner when entering search queries in the sidebar.
singleSearchResultShortcutEnabled	boolean	Indicates if a shortcut is enabled. With the shortcut, users skip the search results page and go directly to the record's detail page when their search returns only a single item. This setting doesn't apply to tags, case comments (in advanced search), and global search.
spellCorrectKnowledgeSearchEnabled	boolean	Indicates if spell check is enabled for Knowledge search.

## SearchSettingsByObject

Includes `ObjectSearchSetting` field type, which is a list of search settings for each object.

Field Name	Field Type	Description
searchSettingsByObject	<a href="#">ObjectSearchSetting</a>	Contains a list of search settings for each object.

## ObjectSearchSetting

A list of search settings for each object.

Field Name	Field Type	Description
enhancedLookupEnabled	boolean	Indicates if enhanced lookups is enabled for the object.
lookupAutoCompleteEnabled	boolean	Indicates if autocomplete is enabled for lookup search. Autocomplete is when users edit the lookup field inline by choosing an autosuggestion.
name	string	The entity name of the object being configured.
resultsPerPageCount	int	The number of search results per page.

## Declarative Metadata Sample Definition

The following is an example of the `Search.settings` file.

```
<?xml version="1.0" encoding="UTF-8"?>
  <SearchSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <documentContentSearchEnabled>true</documentContentSearchEnabled>
    <optimizeSearchForCJKEnabled>true</optimizeSearchForCJKEnabled>
    <recentlyViewedUsersForBlankLookupEnabled>true</recentlyViewedUsersForBlankLookupEnabled>

    <searchSettingsByObject>
      <searchSettingsByObject>
        <enhancedLookupEnabled>false</enhancedLookupEnabled>
        <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
        <name>Account</name>
        <resultsPerPageCount>25</resultsPerPageCount>
      </searchSettingsByObject>
      <searchSettingsByObject>
        <enhancedLookupEnabled>false</enhancedLookupEnabled>
        <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
        <name>Activity</name>
        <resultsPerPageCount>25</resultsPerPageCount>
      </searchSettingsByObject>
      <searchSettingsByObject>
        <enhancedLookupEnabled>false</enhancedLookupEnabled>
        <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
        <name>Asset</name>
        <resultsPerPageCount>25</resultsPerPageCount>
      </searchSettingsByObject>
    </searchSettingsByObject>
    <sidebarAutoCompleteEnabled>true</sidebarAutoCompleteEnabled>
    <sidebarDropDownListEnabled>true</sidebarDropDownListEnabled>
    <sidebarLimitToItemsIOwnCheckboxEnabled>true</sidebarLimitToItemsIOwnCheckboxEnabled>

    <singleSearchResultShortcutEnabled>true</singleSearchResultShortcutEnabled>
    <spellCorrectKnowledgeSearchEnabled>true</spellCorrectKnowledgeSearchEnabled>
  </SearchSettings>
```

## Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Account settings metadata for an organization.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>Search</members>
    <name>Settings</name>
  </types>
  <version>37.0</version>
</Package>
```

## SearchLayouts

Represents the metadata associated with the Search Layouts for an object. You can customize which fields to display for users in search results, search filter fields, lookup dialogs, and recent record lists on tab home pages.

For more information, see "Customize Search Layouts" and "Customize Search Layouts for Custom Objects" in the Salesforce online help.

## Version

Search layouts for custom objects are available in API version 14.0 and later. The ability to modify search layouts for standard objects (except events and tasks) is available in API version 27.0 and later.

## Fields

Field	Field Type	Description
customTabListAdditionalFields	string[]	The list of fields displayed in the Recent <i>Object Name</i> list view on a tab associated with the object. The <i>name</i> field is mandatory and is always displayed as the first column header, so it is not included in this list; all additional fields are included. The field name relative to the object name, for example <i>MyCustomField__c</i> , is specified for each custom field.
excludedStandardButtons	string[]	The list of standard buttons excluded from the search layout.
listViewButtons	string[]	The list of buttons available in list views for an object.  This field is equivalent to the Buttons Displayed value in the <i>Object Name List View</i> in the Search Layouts related list on the object detail page in the Salesforce user interface. For more information, see "Standard and Enhanced Lookups in Salesforce Classic" in the Salesforce online help.
lookupDialogsAdditionalFields	string[]	The list of fields displayed in a lookup dialog for the object. The <i>name</i> field is mandatory and is always displayed as the

Field	Field Type	Description
		<p>first column header, so it is not included in this list; all additional fields are included. The field name relative to the object name, for example <i>MyCustomField__c</i>, is specified for each custom field.</p> <p>Salesforce objects often include one or more <i>lookup fields</i> that allow users to associate two records together in a relationship. For example, a contact record includes an <code>Account</code> lookup field that represents the relationship between the contact and the organization with which the contact is associated. A lookup search dialog helps you search for the record associated with the one being edited. Lookup filter fields allow you to filter your lookup search by a customized list of fields in the object.</p> <p>This field is equivalent to the <code>Lookup Dialogs</code> in the Search Layouts related list on the object detail page in the application user interface. For more information, see “Standard and Enhanced Lookups in Salesforce Classic” in the Salesforce online help.</p>
<code>lookupFilterFields</code>	<code>string[]</code>	<p>The list of fields that can be used to filter enhanced lookups for an object. Enhanced lookups are optionally enabled by your administrator. The field name relative to the object name, for example <i>MyCustomField__c</i>, is specified for each custom field.</p> <p>This field is equivalent to the <code>Lookup Filter Fields</code> in the Search Layouts related list on the object detail page in the application user interface. For more information, see “Standard and Enhanced Lookups in Salesforce Classic” in the Salesforce online help.</p>
<code>lookupPhoneDialogsAdditionalFields</code>	<code>string[]</code>	<p>The list of phone-related fields displayed in a lookup dialog for the object. The <code>name</code> field is mandatory and is always displayed as the first column header, so it is not included in this list; all additional fields are included. The field name relative to the object name, for example <i>MyCustomField__c</i>, is specified for each custom field.</p> <p>This list enables integration of the fields with a SoftPhone dial pad. For more information, see “About CTI 1.0 and 2.0 SoftPhones” in the Salesforce online help.</p> <p>This field is equivalent to the <code>Lookup Phone Dialogs</code> in the Search Layouts related list on the object detail page in the application user interface.</p>
<code>searchFilterFields</code>	<code>string[]</code>	<p>The list of fields that can be used to filter a search for the object. The field name relative to the object name, for</p>



Field	Field Type	Description
		<p>example <i>MyCustomField__c</i>, is specified for each custom field.</p> <p>This field is equivalent to the <code>Search Filter Fields</code> in the Search Layouts related list on the object detail page in the application user interface.</p>
<code>searchResultsAdditionalFields</code>	<code>string[]</code>	<p>The list of fields displayed in a search result for the object. The <code>name</code> field is mandatory and is always displayed as the first column header, so it is not included in this list; all additional fields are included. The field name relative to the object name, for example <i>MyCustomField__c</i>, is specified for each custom field.</p> <p>This field is equivalent to the <code>Search Results</code> in the Search Layouts related list on the object detail page in the application user interface.</p>
<code>searchResultsCustomButtons</code>	<code>string[]</code>	<p>The list of custom buttons available in a search result for the object. The actions associated with the buttons can be applied to any of the records returned in the search result.</p>

## Declarative Metadata Sample Definition

A sample definition of search layouts in an object is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
  <CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    . . .
    <searchLayouts>
      <listViewButtons>New</listViewButtons>
      <listViewButtons>Accept</listViewButtons>
      <listViewButtons>ChangeOwner</listViewButtons>
      <lookupDialogsAdditionalFields>firstQuote__c</lookupDialogsAdditionalFields>
      <lookupDialogsAdditionalFields>finalQuote__c</lookupDialogsAdditionalFields>
      <searchResultsAdditionalFields>CREATEDBY_USER</searchResultsAdditionalFields>
    </searchLayouts>
    . . .
  </CustomObject>
```

## SynonymDictionary

Represents a set of synonym groups, which are groups of words or phrases that are treated as equivalent in users' searches. You can define synonym groups to optimize search results for acronyms, variations of product names, and other terminology unique to your organization.

Synonyms are available in Service Cloud features such as Salesforce Knowledge. This type extends the Metadata metadata type and inherits its `fullName` field.

## File Suffix and Directory Location

SynonymDictionary components have the suffix `.synonymDictionary` and are stored in the `synonymDictionaries` folder.

## Version

SynonymDictionary components are available in API version 29.0 and later.

## Special Access Rules

Synonyms must be enabled in your organization. Only users with the “Manage Synonyms” permission can access this object.

## Fields

Field Name	Field Type	Description
<code>groups</code>	<a href="#">SynonymGroup</a>	The synonym groups defined in this dictionary.
<code>isProtected</code>	boolean	Indicates whether this component is protected ( <code>true</code> ) or not ( <code>false</code> ). Protected components cannot be linked to or referenced by components created in the installing organization.
<code>label</code>	string	Required. Specifies the display name of the synonym dictionary.

## SynonymGroup

Represents a group of synonymous words or phrases.

Field Name	Field Type	Description
<code>languages</code>	Language	Required. Specifies the languages the synonym group applies to. If synonyms are specific to a single language, specify only that language. If the synonyms apply to multiple languages, specify multiple languages for one synonym group.
<code>terms</code>	string	Required. A word or phrase synonymous with other terms in the group. Maximum of 50 characters. Minimum of two <code>terms</code> per group.  Synonym groups are symmetric, which means that if oranges and apples are defined in a synonym group, a search for <i>oranges</i> will return a match for <i>apples</i> , and vice versa for a search for <i>apples</i> .

## Declarative Metadata Sample Definition

The following is an example of a `SynonymDictionary` component:

```
<?xml version="1.0" encoding="UTF-8"?>
<SynonymDictionary xmlns="http://soap.sforce.com/2006/04/metadata">
  <groups>
    <languages>en_US</languages>
    <terms>Salesforce</terms>
    <terms>salesforce.com</terms>
    <terms>The Customer Company</terms>
    <terms>SFDC</terms>
  </groups>
  <groups>
    <languages>fr</languages>
    <terms>renault</terms>
    <terms>clio</terms>
  </groups>
  <label>Sample Dictionary</label>
</SynonymDictionary>
```

The following is an example `package.xml` that references the `SynonymDictionary` component.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>Sample Dictionary</members>
    <name>SynonymDictionary</name>
  </types>
  <version>39.0</version>
</Package>
```

## Usage

If you have existing synonym groups defined before API version 29.0, your existing groups are associated with a default dictionary called `_Default`.

If you have a set of synonyms that require frequent updates, we recommend assigning the synonym group or groups to a dedicated dictionary with a small number of groups. Each time you deploy an existing dictionary, all of its synonym groups are overwritten. We don't support deploying updates to only a single synonym group within a dictionary.

## ExternalDataSource

---

Represents the metadata associated with an external data source. Create external data sources to manage connection details for integration with data and content that are stored outside your Salesforce org.

This type extends the `Metadata` type and inherits its `fullName` field.

## File Suffix and Directory Location

`ExternalDataSource` components are stored in the `dataSources` directory of the corresponding package directory. `ExternalDataSource` components have the suffix `.dataSource`, and the prefix is the name of the external data source.

## Version

ExternalDataSource components are available in API version 28.0 and later.

## Fields

Field Name	Field Type	Description
<code>authProvider</code>	string	The authentication provider that is represented by the AuthProvider component.
<code>certificate</code>	string	If you specify a certificate, your Salesforce org supplies it when establishing each two-way SSL connection with the external system. The certificate is used for digital signatures, which verify that requests are coming from your Salesforce org.
<code>customConfiguration</code>	string	<p>A string of configuration parameters that are specific to the external data source's type.</p> <ul style="list-style-type: none"> <li><a href="#">customConfiguration for Salesforce Connect—Cross-Org Adapter</a></li> <li><a href="#">customConfiguration for Salesforce Connect—OData 2.0 or 4.0 Adapter</a></li> <li><a href="#">customConfiguration for Salesforce Connect—Custom Adapter</a></li> </ul>
<code>endpoint</code>	string	<p>The URL of the external system, or if that URL is defined in a named credential, the named credential URL. Corresponds to <code>URL</code> in the user interface.</p> <p>A named credential URL contains the scheme <code>callout:</code>, the name of the named credential, and an optional path. For example: <code>callout:My_Named_Credential/some_path</code>.</p> <p>You can append a query string to a named credential URL. Use a question mark (?) as the separator between the named credential URL and the query string. For example: <code>callout:My_Named_Credential/some_path?format=json</code>.</p>
<code>isWritable</code>	boolean	<p>Lets the Force.com platform and users in this org create, update, and delete records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read only. Corresponds to <code>Writable External Objects</code> in the user interface.</p> <p>Available in API version 35.0 and later. However, with the cross-org adapter for Salesforce Connect, you can set this field to <code>true</code> only in API version 39.0 and later.</p>
<code>label</code>	string	<p>A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views.</p> <p>Examples include Acme Team Marketing Site, or Acme SharePoint.</p>

Field Name	Field Type	Description
<code>oauthRefreshToken</code>	string	The OAuth refresh token. Used to obtain a new access token for an end user when a token expires.
<code>oauthScope</code>	string	Specifies the scope of permissions to request for the access token. Corresponds to <code>Scope</code> in the user interface.
<code>oauthToken</code>	string	The access token issued by the external system.
<code>password</code>	string	The password to be used by your org to access the external system. Ensure that the credentials you use have adequate privileges to access the external system, perform searches, return data, and return information about the external system's metadata.
<code>principalType</code>	ExternalPrincipalType (enumeration of type string)	Determines whether you're using one set or multiple sets of credentials to access the external system. Corresponds to <code>Identity Type</code> in the user interface. The valid values are: <ul style="list-style-type: none"> <li>• <code>Anonymous</code></li> <li>• <code>PerUser</code></li> <li>• <code>NamedUser</code></li> </ul>
<code>protocol</code>	AuthenticationProtocol (enumeration of type string)	The authentication protocol that's required to access the external system. The valid values are: <ul style="list-style-type: none"> <li>• <code>NoAuthentication</code></li> <li>• <code>OAuth</code></li> <li>• <code>Password</code></li> </ul> For cloud-based Files Connect external systems, select <b>OAuth 2.0</b> . For on-premises systems, select <b>Password Authentication</b> . For Simple URL data sources, select <b>No Authentication</b> .
<code>repository</code>	string	Used for SharePoint Online. If metadata is not accessible, use this field to create tables and default table fields.
<code>type</code>	ExternalDataSourceType (enumeration of type string)	For Salesforce Connect, specifies the adapter that connects to the external system. The valid values are: <ul style="list-style-type: none"> <li>• <code>OData</code>—OData 2.0 adapter</li> <li>• <code>OData4</code>—OData 4.0 adapter</li> <li>• <code>SfdcOrg</code>—cross-org adapter</li> <li>• <code>ApexClassId</code>—<code>DataSource.Provider</code> class that defines the custom adapter created via the Apex Connector Framework</li> </ul> For Files Connect, specifies the data source type. The valid values are: <ul style="list-style-type: none"> <li>• <code>ContentHubSharepoint</code>—SharePoint 2010 or 2013</li> <li>• <code>ContentHubSharepointOffice365</code>—SharePoint Online</li> <li>• <code>ContentHubSharepointOneDrive</code>—OneDrive for Business</li> <li>• <code>ContentHubGDrive</code>—Google Drive</li> </ul>

Field Name	Field Type	Description
		<p>If Chatter is enabled, you can also specify <code>SimpleURL</code> to access data hosted on a web server that doesn't require authentication.</p> <ul style="list-style-type: none"> <li><code>outgoingemail</code>—A data source used for sending an email through a quick action.</li> </ul> <p>The <code>Identity</code> and <code>Wrapper</code> types are reserved for future use.</p>
<code>username</code>	<code>string</code>	The username to be used by your org to access the external system. Ensure that the credentials you use have adequate privileges to access the external system, perform searches, return data, and return information about the external system's metadata.
<code>version</code>	<code>string</code>	Reserved for future use.

## customConfiguration for Salesforce Connect—Cross-Org Adapter

The following sample JSON-encoded configuration string defines parameters that apply when the external data source's `type` is set to `SfdcOrg`.

```
{ "apiVersion": "32.0", "environment": "CUSTOM",
  "searchEnabled": "true", "timeout": "120" }
```

The parameters correspond to these fields in the user interface:

- `apiVersion`—API Version
- `environment`—Connect to
- `searchEnabled`—Enable Search
- `timeout`—Connection Timeout

## customConfiguration for Salesforce Connect—OData 2.0 or 4.0 Adapter

The following JSON-encoded configuration string defines parameters that apply when the external data source's `type` is set to `OData` or `OData4`.

```
{ "inlineCountEnabled": "true", "csrfTokenName": "X-CSRF-Token",
  "requestCompression": "false", "pagination": "CLIENT",
  "noIdMapping": "false", "format": "ATOM",
  "searchFunc": "", "compatibility": "DEFAULT",
  "csrfTokenEnabled": "true", "timeout": "120",
  "searchEnabled": "true" }
```

The parameters correspond to these fields in the user interface.

- `compatibility`—Special Compatibility
- `csrfTokenEnabled`—CSRF Protection
- `csrfTokenName`—Anti-CSRF Token Name
- `format`—Format
- `inlineCountEnabled`—Request Row Counts
- `noIdMapping`—High Data Volume

- pagination—Server Driven Pagination
- requestCompression—Compress Requests
- searchEnabled—Enable Search
- searchFunc—Custom Query Option for Salesforce Search
- timeout—Connection Timeout

## customConfiguration for Salesforce Connect—Custom Adapter

The following sample JSON-encoded configuration string defines the parameter that applies when the external data source's `type` is set to the ID of a `DataSource.Provider` class.

```
{"noIdMapping":"false"}
```

The `noIdMapping` parameter corresponds to the `High Data Volume` field in the user interface.

## Declarative Metadata Sample Definition

The following is the definition of an external data source for Salesforce Connect—OData 2.0 or 4.0 adapter.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataSource xmlns="http://soap.sforce.com/2006/04/metadata">
  <authProvider>FacebookAuth</authProvider>
  <customConfiguration>{"compatibility":"DEFAULT",
    "noIdMapping":"false", "inlineCountEnabled":"true",
    "searchEnabled":"true", "format":"ATOM",
    "requestCompression":"false", "pagination":"SERVER",
    "timeout":"120"}</customConfiguration>
  <endpoint>http://myappname.herokuapp.com/DataHub.svc</endpoint>
  <label>DataHub</label>
  <principalType>NamedUser</principalType>
  <protocol>Oauth</protocol>
  <type>OData</type>
</ExternalDataSource>
```

# CHAPTER 5 Salesforce Knowledge with SOQL and SOSL

## In this chapter ...

- [Update an Article's Keyword Tracking with SOQL](#)
- [Update an Article Viewstat with SOQL](#)
- [WITH filteringExpression](#)
- [Using Relationship Queries with Data Category Selection Objects](#)
- [About SOSL](#)

Use the Salesforce Object Query Language (SOQL) to search your organization's Knowledge data for specific information. SOQL is similar to the SELECT statement in the widely used Structured Query Language (SQL) but is designed specifically for Salesforce. Use the Salesforce Object Search Language (SOSL) to construct text-based search queries against the search index.

Knowledge Object	Limits
KnowledgeArticleVersion	<ul style="list-style-type: none"><li>• Always filter on a single value of <code>PublishStatus</code> unless the query filters on one or more primary key IDs. To support security, only users with the "Manage Articles" permission see articles whose <code>PublishStatus</code> value is <code>Draft</code>.</li><li>• Archived article versions are stored in the <code>articletype_kav</code> object. To query archived article versions, specify the article <code>Id</code> and set <code>IsLatestVersion='0'</code>.</li><li>• Always filter on a single value of <code>Language</code>. However, in SOQL, you can filter on more than one <code>Language</code> if there is a filter on <code>Id</code> or <code>KnowledgeArticleId</code>.</li></ul>

For generic SOQL and SOSL information see the [Force.com SOQL and SOSL Reference](#).



## Update an Article's Keyword Tracking with SOQL

Track keywords that are used in Salesforce Knowledge article searches with the `UPDATE TRACKING` optional clause on a SOQL query. `UPDATE TRACKING` is an optional clause that can be added to a `SELECT` statement of a SOQL query to report on article searches and views. Developers can use `UPDATE TRACKING` to track the keywords that are used in Salesforce Knowledge article searches.



**Example:** You can use this syntax to track a keyword that are used in Salesforce Knowledge article search:

```
SELECT Title FROM FAQ__kav
WHERE Keyword='Apex' and
Language = 'en_US' and
KnowledgeArticleVersion = 'ka230000000PCiy'
UPDATE TRACKING
```

## Update an Article Viewstat with SOQL

Determine how many hits a Salesforce Knowledge article has had by using the `UPDATE VIEWSTAT` optional clause on a SOQL query. You can get a view count for every article that you have access to online.

The `UPDATE VIEWSTAT` clause is used in a `SELECT` statement to report on Salesforce Knowledge article searches and views. It allows developers to update an article's view statistics.

You can use this syntax to increase the view count for every article you have access to online:

```
SELECT Title FROM FAQ__kav
WHERE PublishStatus='online' and
Language = 'en_US' and
KnowledgeArticleVersion = 'ka230000000PCiy'
UPDATE VIEWSTAT
```

## WITH *filteringExpression*

You can filter records based on field values, for example, to filter according to category or to query and retrieve changes that are tracked in a user's profile feed by using `WITH filteringExpression`. This optional clause can be added to a `SELECT` statement of a SOQL query.

Unlike the `WHERE` clause which only supports fields from the object specified in the `FROM` clause, `WITH` allows you to filter by other related criteria. For example, you can use the `WITH` clause to filter articles based on their classification in one or more data category groups. The `WITH` clause can only be used in the following cases:

- To filter records based on their categorization. See `WITH DATA CATEGORY filteringExpression`.
- To query and retrieve record changes tracked in a user profile feed. See `UserProfileFeed` in the [Object Reference for Salesforce and Force.com](#).

If `WITH` is specified, the query returns only records that match the filter and are visible to the user. If unspecified, the query returns only the matching records that are visible to the user.

The filtering expression in the statements below is highlighted in bold. The syntax is explained in the following sections.

- `SELECT Title FROM KnowledgeArticleVersion WHERE PublishStatus='online' WITH DATA CATEGORY Geography__c ABOVE usa__c`

- `SELECT Id FROM UserProfileFeed WITH UserId='005D0000001AamR' ORDER BY CreatedDate DESC, Id DESC LIMIT 20`

IN THIS SECTION:


[WITH DATA CATEGORY filteringExpression](#)

You can search for Salesforce Knowledge articles and questions by their data category in a SOQL query. `WITH DATA CATEGORY` is an optional clause in a `SELECT` statement that's used to filter records that are associated with one or more data categories and are visible to users.

## WITH DATA CATEGORY *filteringExpression*

You can search for Salesforce Knowledge articles and questions by their data category in a SOQL query. `WITH DATA CATEGORY` is an optional clause in a `SELECT` statement that's used to filter records that are associated with one or more data categories and are visible to users.

If `WITH DATA CATEGORY` is specified, the `query()` returns only matching records that are associated with the specified data categories and are visible to the user. If unspecified, the `query()` only returns the matching records that are visible to the user.

 **Important:** `CategoryData` is an object and `DATA CATEGORY` is syntax in a SOQL `WITH` clause. `WITH DATA CATEGORY` is valid syntax, but `WITH CategoryData` is not supported.

A SOQL statement using a `WITH DATA CATEGORY` clause must also include a `FROM ObjectTypeName` clause where `ObjectTypeName` equals:

- `KnowledgeArticleVersion` to query all article types
- an article type `API Name` to query a specific article type
- `Question` to query questions

When `ObjectTypeName` equals to `KnowledgeArticleVersion` or any article type `API Name` in the `FROM` clause, a `WHERE` clause must be specified with one of the following parameters:

- `PublishStatus` to query articles depending on their status in the publishing cycle:
  - `WHERE PublishStatus='online'` for published articles
  - `WHERE PublishStatus='archived'` for archived articles
  - `WHERE PublishStatus='draft'` for draft articles
- `Id` to query an article based on its id

For information on article types or questions, see “Knowledge Article Types” and “Finding and Viewing Questions” in the Salesforce Help.

 **Note:** The `WITH DATA CATEGORY` clause does not support bind variables.

## *filteringExpression*

The *filteringExpression* in the `WITH DATA CATEGORY` clause uses the following syntax:

```
dataCategorySelection [AND [dataCategorySelection2] [...]]
```

The examples in this section are based on the following data category group:

```
Geography__c
  ww__c
```

```
northAmerica__c
  usa__c
  canada__c
  mexico__c
europe__c
  france__c
  uk__c
asia__c
```

The category filtering in the statements below is highlighted in bold. The syntax is explained in the following sections.

- **SELECT** Title FROM KnowledgeArticleVersion WHERE PublishStatus='online' **WITH DATA CATEGORY Geography\_\_c ABOVE usa\_\_c**
- **SELECT** Title FROM Question WHERE LastReplyDate > 2005-10-08T01:02:03Z **WITH DATA CATEGORY Geography\_\_c AT (usa\_\_c, uk\_\_c)**
- **SELECT** UrlName FROM KnowledgeArticleVersion WHERE PublishStatus='draft' **WITH DATA CATEGORY Geography\_\_c AT usa\_\_c AND Product\_\_c ABOVE\_OR\_BELOW mobile\_phones\_\_c**

You can only use the AND logical operator. The following syntax is incorrect as OR is not supported:

```
WITH DATA CATEGORY Geography__c ABOVE usa__c OR Product__c AT mobile_phones__c
```

#### IN THIS SECTION:

##### [dataCategorySelection](#)

The syntax of the data category selection in a **WITH DATA CATEGORY** clause in a SOQL query includes a category group name to use as a filter, the filter selector, and the name of the category to use for filtering.

##### [Filtering Selectors](#)

When specifying filters for a **WITH CATEGORY** clause of a SOQL query, you can use **AT** to select the specified category, **ABOVE** to select the category and all its parent categories, **BELOW** to select the category and all its subcategories, and **ABOVE\_OR\_BELOW** to select the category, its parent categories, and its subcategories.

##### [Example WITH DATA CATEGORY Clauses](#)

Here are examples of **WITH DATA CATEGORY** clauses in a **SELECT** statement in a SOQL query.

### *dataCategorySelection*

The syntax of the data category selection in a **WITH DATA CATEGORY** clause in a SOQL query includes a category group name to use as a filter, the filter selector, and the name of the category to use for filtering.

The *dataCategorySelection* uses the following syntax:

```
dataCategoryGroupName filteringSelector dataCategoryName
```

#### Syntax

#### Description

*dataCategoryGroupName* The name of the data category group to use as a filter. `Geography__c` is the data category group in the following example.

You cannot use the same data category group more than once in a query. As an example, the following command is incorrect: `WITH DATA CATEGORY Geography__c ABOVE usa__c AND Geography__c BELOW europe__c`

Syntax	Description
<i>filteringSelector</i>	The selector used to filter the data in the specified data category. See <a href="#">Filtering Selectors</a> for a list of valid selectors.
<i>dataCategoryName</i>	<p>The name of the data category for filtering. You must have visibility on the category you specify. For more information on category visibility, see "Data Category Visibility" in the Salesforce Help.</p> <p>You can use parentheses to apply the filtering operator to more than one data category. Each data category must be separated by a comma.</p> <p>Example: WITH DATA CATEGORY Geography__c AT (usa__c,france__c,uk__c)</p> <p>You can't use the AND operator instead of parentheses to list multiple data categories. The following syntax does not work WITH DATA CATEGORY Geography__c AT usa__c AND france__c</p>

## Filtering Selectors

When specifying filters for a WITH CATEGORY clause of a SOQL query, you can use AT to select the specified category, ABOVE to select the category and all its parent categories, BELOW to select the category and all its subcategories, and ABOVE\_OR\_BELOW to select the category, its parent categories, and its subcategories.

The following table lists the *filteringSelector* values that are used in the *dataCategorySelection* syntax.

The examples in this section are based on the following data category group:


```

Geography__c
  ww__c
    northAmerica__c
      usa__c
      canada__c
      mexico__c
    europe__c
      france__c
      uk__c
  asia__c

```

Selector	Description
AT	<p>Select the specified data category.</p> <p>For example, the following syntax selects <code>asia__c</code>.</p> <pre>WITH DATA CATEGORY Geography__c AT asia__c</pre>
ABOVE	<p>Select the specified data category and all its parent categories.</p> <p>For example, the following syntax selects <code>usa__c</code>, <code>northAmerica__c</code>, and <code>ww__c</code>.</p> <pre>WITH DATA CATEGORY Geography__c ABOVE usa__c</pre>

Selector	Description
BELOW	<p>Select the specified data category and all its subcategories.</p> <p>For example the following selects <code>northAmerica__c</code>, <code>usa__c</code>, <code>canada__c</code>, and <code>mexico__c</code>.</p> <pre>WITH DATA CATEGORY Geography__c BELOW northAmerica__c</pre>
ABOVE_OR_BELOW	<p>Select the specified data category and:</p> <ul style="list-style-type: none"> <li>all its parent categories</li> <li>all its subcategories</li> </ul> <p>For example the following selects <code>ww__c</code>, <code>europa__c</code>, <code>france__c</code> and <code>uk__c</code>.</p> <pre>WITH DATA CATEGORY Geography__c ABOVE_OR_BELOW europa__c</pre>

 **Note:** For more information on data category groups, data categories, parent and subcategories, see "Data Categories in Salesforce.com" in the Salesforce Help.

## Example WITH DATA CATEGORY Clauses

Here are examples of `WITH DATA CATEGORY` clauses in a `SELECT` statement in a SOQL query.

Type of Search	Example(s)
Select the title from all questions classified with the <code>mobile_phones__c</code> data category in the <code>Product__c</code> data category group	<pre>SELECT Title FROM Question WHERE LastReplyDate &lt; 2005-10-08T01:02:03Z WITH DATA CATEGORY Product__c AT mobile_phones__c</pre>
Select the title and summary from all published Knowledge articles classified: <ul style="list-style-type: none"> <li>above or below <code>europa__c</code> in the <code>Geography__c</code> data category group</li> <li>below <code>allProducts__c</code> in the <code>Product__c</code> data category group</li> </ul>	<pre>SELECT Title, Summary FROM KnowledgeArticleVersion WHERE PublishStatus='Online' AND Language = 'en_US' WITH DATA CATEGORY Geography__c ABOVE_OR_BELOW europa__c AND Product__c BELOW All__c</pre>
Select the ID and title from draft articles of type "Offer__kav" classified : <ul style="list-style-type: none"> <li>with the <code>france__c</code> or <code>usa__c</code> data category in the <code>Geography__c</code> data category group</li> <li>above the <code>dsl__c</code> data category in the <code>Product__c</code> data category group</li> </ul>	<pre>SELECT Id, Title FROM Offer__kav WHERE PublishStatus='Draft' AND Language = 'en_US' WITH DATA CATEGORY Geography__c AT (france__c,usa__c) AND Product__c ABOVE dsl__c</pre>

## Using Relationship Queries with Data Category Selection Objects

Data categories are used to classify records. In SOQL, you can use the `Article__DataCategorySelection` or `QuestionDataCategorySelection` objects. You can also build a relationship query with the `DataCategorySelections` relationship name in a `FROM` clause.

Imagine an `Offer` article type. The following query returns the ID of any categorization associated with an offer and the ID of the categorized article.

```
SELECT Id, ParentId
FROM Offer__DataCategorySelection
```

The following example uses the `DataCategorySelections` relationship name to build a relationship query that returns the ID of published offers and the ID of all the categorizations associated to these offers.

```
SELECT Id, Title
(
  SELECT Id
  FROM DataCategorySelections
)
FROM Offer__kav WHERE publishStatus='online';
```

## About SOSL

Use the Salesforce Object Search Language (SOSL) to construct text-based search queries against the search index.

You can search text, email, and phone fields for multiple objects, including custom objects, that you have access to in a single query in the following environments.

- SOAP or REST calls
- Apex statements
- Visualforce controllers and getter methods
- Schema Explorer of the Eclipse Toolkit

 **Note:** If your org has relationship queries enabled, SOSL supports SOQL relationship queries.

## Compare SOSL and SOQL

Use SOQL when:

- You want to search against the org's database. Results from a database search include matches for the exact search string.
- You know in which objects or fields the data resides.
- You want to:
  - Retrieve data from a single object or from multiple objects that are related to one another
  - Count the number of records that meet specified criteria
  - Sort results as part of the query
  - Retrieve data from number, date, or checkbox fields

Use SOSL when:

- You don't know in which object or field the data resides, and you want to find it in the most efficient and fastest way possible.

- You want to:
  - Retrieve data for a specific term that you know exists within a field. Because SOSL can tokenize multiple terms within a field and build a search index off this, SOSL searches are faster and can return more relevant results. Depending on the search object, SOSL searches also take advantage of the advanced features of the search index, such as out-of-order matching, synonyms, lemmatization, and spell check.
  - Retrieve multiple objects and fields efficiently, and the objects might or might not be related to one another.
  - Retrieve data for a particular division in an org using the divisions feature, and you want to find it in the most efficient way possible.

## Define Efficient SOSL Text Searches

If your searches are too general, they are slow and return too many results. Use the following clauses to write more targeted and useful searches.

- `IN`: Limits the types of fields to search, including email, name, or phone.
- `LIMIT`: Specifies the maximum number of rows to return.
- `OFFSET`: Displays the search results on multiple pages.
- `RETURNING`: Limits the objects and fields to return.
- `WITH DATA CATEGORY`: Specifies the data categories to return.
- `WITH DivisionFilter`: Specifies the division field to return.
- `WITH NETWORK`: Specifies the community ID to return.
- `WITH PricebookId`: Specifies the price book ID to return.

### IN THIS SECTION:

#### [Update an Article's Keyword Tracking with SOSL](#)

Track keywords that are used in Salesforce Knowledge article searches with the `UPDATE TRACKING` optional clause on a SOSL query. You can use the language attribute to search by locale.

#### [Update an Article's Viewstat with SOSL](#)

Determine how many hits a Salesforce Knowledge article has had by using the `UPDATE VIEWSTAT` optional clause on a SOSL query. You can use the language attribute to search by locale.

#### [WITH DATA CATEGORY DataCategorySpec](#)

`WITH DATA CATEGORY` is an optional clause that can be added to a SOSL query to filter all search results that are associated with one or more data categories and are visible to users. This clause is used in searches of Salesforce Knowledge articles and questions.

#### [WITH SNIPPET](#)

`WITH SNIPPET` is an optional clause that can be added to a SOSL query for article, case, feed, and idea searches. On the search results page, excerpts below article titles show terms matching the search query highlighted within the context of surrounding text. Snippets make it easier for users to identify the content they're looking for.

## Update an Article's Keyword Tracking with SOSL

Track keywords that are used in Salesforce Knowledge article searches with the `UPDATE TRACKING` optional clause on a SOSL query. You can use the language attribute to search by locale.

The `UPDATE TRACKING` clause is used to report on Salesforce Knowledge article searches and views. It allows developers to track the keywords used in Salesforce Knowledge article searches. Also, the language attribute can be used to search by a specific language (locale). However, only one language can be specified in a single query. Make a separate query for each language that you want. Use the Java format, which uses the underscore (for example, `fr_FR`, `jp_JP`, and so on), to supply locales. Search the Web for “java locale codes” to get a list of supported locales.

You can use this syntax to track a keyword used in Salesforce Knowledge article search:

```
FIND {Keyword}
RETURNING KnowledgeArticleVersion (Title WHERE PublishStatus="Online" and language="en_US")
UPDATE TRACKING
```

## Update an Article's Viewstat with SOSL

Determine how many hits a Salesforce Knowledge article has had by using the `UPDATE VIEWSTAT` optional clause on a SOSL query. You can use the language attribute to search by locale.

The optional `UPDATE VIEWSTAT` clause is used to report on Salesforce Knowledge article searches and views. It allows developers to update an article's view statistics. Also, the language attribute can be used to search by a specific language (locale). However, only one language can be specified in a single query. Make a separate query for each language that you want. Use the Java format, which uses the underscore (for example, `fr_FR`, `jp_JP`, and so on), to supply locales. Search the Web for “java locale codes” to get a list of supported locales.

You can use this syntax to increase the view count for every article you have access to online in US English:

```
FIND {Title}
RETURNING FAQ__kav (Title WHERE PublishStatus="Online" and
language="en_US" and
KnowledgeArticleVersion = 'ka230000000PCiy')
UPDATE VIEWSTAT
```

## WITH DATA CATEGORY *DataCategorySpec*

`WITH DATA CATEGORY` is an optional clause that can be added to a SOSL query to filter all search results that are associated with one or more data categories and are visible to users. This clause is used in searches of Salesforce Knowledge articles and questions.

The `WITH DATA CATEGORY` clause can be used in API version 18.0 or later.

### Syntax

The `WITH DATA CATEGORY` syntax is:

```
WITH DATA CATEGORY DataCategorySpec [logicalOperator DataCategorySpec2 ... ]
```

Where *DataCategorySpec* consists of a *groupName*, *operator*, and *category*.

Name	Description
<b>groupName</b>	The name of the data category group to filter. For information on category groups, see “Create and Modify Category Groups” in the Salesforce Help.
<b>Operator</b>	Use one of the following operators: <ul style="list-style-type: none"> <li>• <code>AT</code>—Queries the specified data category.</li> </ul>



Name	Description
	<ul style="list-style-type: none"> <li>• ABOVE—Queries the specified data category and all of its parent categories.</li> <li>• BELOW—Queries the specified data category and all of its subcategories.</li> <li>• ABOVE_OR_BELOW—Queries the specified data category, all of its parent categories, and all of its subcategories.</li> </ul>
<b>category</b>	The name of the category to filter. To include multiple data categories, enclose them in parentheses, separated by commas. For information on categories, see “Add Data Categories to Category Groups” in the Salesforce Help.

You can add multiple data category specifiers by using the logical operator `AND`. Other operators, such as `OR` and `AND NOT`, are not supported.

A SOSL statement using the `WITH DATA CATEGORY` clause must also include a `RETURNING ObjectName` clause, with a `WHERE` clause that filters on the `PublishStatus` field.

In the `RETURNING` clause, specify one of the following for *ObjectName*:

- To search a specific article type, use the article type name with the suffix `__kav`
- To search all article types, use [KnowledgeArticleVersion](#)
- To search questions, use [Question](#)

For information on article types, see “Knowledge Article Types” in the Salesforce Help.

The `WHERE` clause must use one of the following publish statuses:

- `WHERE PublishStatus='online'` for published articles
- `WHERE PublishStatus='archived'` for archived articles
- `WHERE PublishStatus='draft'` for draft articles

## Examples

Search Type	Example
Search all published (online)Salesforce Knowledge articles with a category from one category group.	
Search online FAQ articles with categories from two category groups.	
Search archived FAQ articles from one category group.	

**Search Type****Example**

Search all draft Salesforce Knowledge articles from one category group.

```
FIND {tourism} RETURNING KnowledgeArticleVersion
  (Id, Title WHERE PublishStatus='draft')
  WITH DATA CATEGORY Geography__c BELOW Europe__c
```

For information on the `WITH DATA CATEGORY` clause, see the [WITH DATA CATEGORY filteringExpression](#).



**Tip:** You can also search for articles by ID, without using the `WITH DATA CATEGORY` clause. For more information, see [Example WHERE Clauses](#).

## WITH SNIPPET

`WITH SNIPPET` is an optional clause that can be added to a SOSL query for article, case, feed, and idea searches. On the search results page, excerpts below article titles show terms matching the search query highlighted within the context of surrounding text. Snippets make it easier for users to identify the content they're looking for.



**Note:** To generate search results with highlighted matches but not snippets, use `WITH HIGHLIGHT`.

Search snippets and highlights are generated from the following field types.

- Email
- Text
- Text Area
- Text Area (Long)
- Text Area (Rich)

Search snippets and highlights are *not* generated from the following field types.

- Checkbox
- Currency
- Date
- Date/Time
- File
- Formula
- Lookup Relationship
- Number
- Percent
- Phone
- Picklist
- Picklist (Multi-Select)
- URL


 **Example:** The following SOSL statement returns snippets for articles that match the search term *San Francisco*.

```
FIND {San Francisco} IN ALL FIELDS RETURNING KnowledgeArticleVersion(id, title WHERE
PublishStatus = 'Online' AND Language = 'en_US') WITH
SNIPPET (target_length=120)
```

The matching terms are highlighted with <mark> tags within the context of the snippet results. Stemmed forms of the term and any synonyms defined are also highlighted.

 **Example:**

```
[ {
  "attributes" : {
    "type" : "KnowledgeArticleVersion",
    "url" : "/services/data/v32.0/subjects/KnowledgeArticleVersion/kaKD0000000001MAA"
  },
  "Id" : "kaKD0000000001MAA",
  "Title" : "San Francisco",
  "Summary" : "City and County of San Francisco",
  "snippet.text" : "<mark>San</mark> <mark>Francisco</mark>, officially the City and
County of <mark>San</mark> <mark>Francisco</mark> is the... City and County of
<mark>San</mark> <mark>Fran</mark>"
  "highlight.Title" : "<mark>San</mark> <mark>Francisco</mark>"
}, {
  "attributes" : {
    "type" : "KnowledgeArticleVersion",
    "url" : "/services/data/v32.0/subjects/KnowledgeArticleVersion/kaBD0000000007DMAQ"
  },
  "Id" : "kaBD0000000007DMAQ",
  "Title" : "San Francisco Bay Area",
  "Summary" : "Nine county metropolitan area",
  "snippet.text" : "The <mark>SF</mark> Bay Area, commonly known as the Bay Area, is
a populated region that"
  "highlight.Title" : "<mark>San</mark> <mark>Francisco</mark> Bay Area"
}, {
  "attributes" : {
    "type" : "KnowledgeArticleVersion",
    "url" : "/services/data/v32.0/subjects/KnowledgeArticleVersion/ka3D0000000042OIAQ"
  },
  "Id" : "ka3D0000000042OIAQ",
  "Title" : "California",
  "Summary" : "State of California",
  "snippet.text" : "(Greater Los Angeles area and <mark>San</mark>
<mark>Francisco</mark> Bay Area, respectively), and eight of the nation's 50 most"
} ]
```

 **Note:** In this example, “SF” (as a synonym defined for “San Francisco”) and “San Fran” (as a stemmed form of “San Francisco”) are also highlighted in the results as matching terms.

## Usage

For SOSL statements using the `WITH SNIPPET` clause, we recommend using a `RETURNING ObjectName` clause, with a `WHERE` clause that filters on the `PublishStatus` field.

In the `RETURNING` clause, specify one of the following for `ObjectName`:

- To search a specific article type, use the article type name with the suffix `__kav`.
- To search all article types, use [KnowledgeArticleVersion](#).
- To search case, case comment, feed, feed comment, idea, and idea comment types, use `Case`, `CaseComment`, `FeedItem`, `FeedComment`, `Idea`, and `IdeaComment`. For example:

```
FIND {San Francisco} IN ALL FIELDS RETURNING FeedItem, FeedComment WITH SNIPPET
(target_length=120)
```

Other objects that are included in searches that contain `WITH SNIPPET` don't return snippets.


Snippets aren't displayed for search terms that contain a wildcard, when the search doesn't return any articles, or if the user doesn't have access to the field that contains the snippet. Even if you add the `WITH SNIPPET` clause, searches that don't return snippets don't return snippets.

Snippets are only displayed when 20 or fewer results are returned on a page.

 **Tip:** Use the `LIMIT` or `OFFSET` clause to return only 20 results at a time.

## Escaped HTML Tags

When matching terms within HTML tags are returned in a snippet, the HTML tags are escaped and the matching terms are highlighted in the results.

 **Example:** A search for `salesforce` returns an article with the text "For more information, visit `<a href='http://salesforce.com'>salesforce.com</a>`". The original hyperlink tags from the article are escaped (encoded) and "salesforce" is highlighted in the snippet result.


```
For more information, visit &lt;a
href='http://salesforce.com'&gt;salesforce.com&lt;/a&gt;
```

## Target Snippet Length

By default, each snippet displays up to approximately 300 characters, which is usually three lines of text in a standard browser window display. The number of characters displayed is the *target length*, within a statistically insignificant degree of variance.

Snippets consist of one or more *fragments* of text that contain the matching terms. If the returned snippet includes multiple text fragments (for example, for matches within multiple fields), the target length is the maximum total length of all the returned fragments.

To specify an alternate target length, add the optional `target_length` parameter to the `WITH SNIPPET` clause. You can specify a target length from 50 to 1,000 characters. When the `target_length` is set to an invalid number, such as `0` or a negative number, the length defaults to `300`.

 **Example:** A `target_length` parameter of 120 characters is useful for displaying a snippet of approximately three lines of text in a standard mobile interface.

```
FIND {San Francisco} IN ALL FIELDS RETURNING KnowledgeArticleVersion(id, title WHERE
PublishStatus = 'Online' AND Language = 'en_US') WITH
SNIPPET(target_length=120)
```

## Supported APIs

The `WITH SNIPPET` clause can be used in API version 32.0 or later. The `WITH SNIPPET` clause in SOSL is supported in SOAP API, REST API, and Apex.

# CHAPTER 6 Salesforce Knowledge, Apex, and Visualforce

## In this chapter ...

- [PublishingService Class](#)

With the power of Apex and Visualforce, you can customize your knowledge base look and feel, for example with custom search pages, or build a robust article publishing cycle.

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Force.com platform server in conjunction with calls to the Force.com API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex code can be initiated by Web service requests and from triggers on objects.

Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Force.com platform. The Visualforce framework includes a tag-based markup language, similar to HTML, and a set of server-side “standard controllers” that make basic database operations, such as queries and saves, very simple to perform.

Due to the endless possibilities with Apex and Visualforce, only the Apex Publishing Service class is in this guide. For more Apex and Visualforce information see the [Apex Developer Guide](#) and the [Visualforce Developer Guide](#).

# PublishingService Class

---

Use the methods in the `KbManagement.PublishingService` class to manage the lifecycle of an article and its translations.

## Namespace

`KbManagement`

## Usage

Use the methods in the `KbManagement.PublishingService` class to manage the following parts of the lifecycle of an article and its translations:

- Publishing
- Updating
- Retrieving
- Deleting
- Submitting for translation
- Setting a translation to complete or incomplete status
- Archiving
- Assigning review tasks for draft articles or translations

 **Note:** Date values are based on GMT.

To use the methods in this class, you must enable Salesforce Knowledge. See [Salesforce Knowledge Implementation Guide](#) for more information on setting up Salesforce Knowledge.

### IN THIS SECTION:

[PublishingService Methods](#)

## PublishingService Methods

The following are methods for `PublishingService`. All methods are static.

### IN THIS SECTION:

[archiveOnlineArticle\(articleId, scheduledDate\)](#)

Archives an online version of an article. If the specified `scheduledDate` is null, the article is archived immediately. Otherwise, it archives the article on the scheduled date.

[assignDraftArticleTask\(articleId, assigneeId, instructions, dueDate, sendEmailNotification\)](#)

Assigns a review task related to a draft article.

[assignDraftTranslationTask\(articleVersionId, assigneeId, instructions, dueDate, sendEmailNotification\)](#)

Assigns a review task related to a draft translation.

[cancelScheduledArchivingOfArticle\(articleId\)](#)

Cancels the scheduled archiving of an online article.

[cancelScheduledPublicationOfArticle\(articleId\)](#)

Cancels the scheduled publication of a draft article.

[completeTranslation\(articleVersionId\)](#)

Puts a translation in a completed state that is ready to publish.

[deleteArchivedArticle\(articleId\)](#)

Deletes an archived article.

[deleteArchivedArticleVersion\(articleId, versionNumber\)](#)

Deletes a specific version of an archived article.

[deleteDraftArticle\(articleId\)](#)

Deletes a draft article.

[deleteDraftTranslation\(articleVersionId\)](#)

Deletes a draft translation.

[editArchivedArticle\(articleId\)](#)

Creates a draft article from the archived master version and returns the new draft master version ID of the article.

[editOnlineArticle\(articleId, unPublish\)](#)

Creates a draft article from the online version and returns the new draft master version ID of the article. Also, unpublishes the online article, if *unPublish* is set to *true*.

[editPublishedTranslation\(articleId, language, unPublish\)](#)

Creates a draft version of the online translation for a specific language and returns the new draft master version ID of the article. Also, unpublishes the article, if set to *true*.

[publishArticle\(articleId, flagAsNew\)](#)

Publishes an article. If *flagAsNew* is set to *true*, the article is published as a major version.

[restoreOldVersion\(articleId, versionNumber\)](#)

Creates a draft article from an existing online article based on the specified archived version of the article and returns the article version ID.

[scheduleForPublication\(articleId, scheduledDate\)](#)

Schedules the article for publication as a major version. If the specified date is null, the article is published immediately.

[setTranslationToIncomplete\(articleVersionId\)](#)

Sets a draft translation that is ready for publication back to “in progress” status.

[submitForTranslation\(articleId, language, assigneeId, dueDate\)](#)

Submits an article for translation to the specified language. Also assigns the specified user and due date to the submittal and returns new ID of the draft translation.

**archiveOnlineArticle(articleId, scheduledDate)**

Archives an online version of an article. If the specified *scheduledDate* is null, the article is archived immediately. Otherwise, it archives the article on the scheduled date.

**Signature**

```
public static Void archiveOnlineArticle(String articleId, Datetime scheduledDate)
```



## Parameters

*articleId*

Type: String

*scheduledDate*

Type: Datetime

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';
Datetime scheduledDate = Datetime.newInstanceGmt(2012, 12, 1, 13, 30, 0);
KbManagement.PublishingService.archiveOnlineArticle(articleId, scheduledDate);
```

## **assignDraftArticleTask(articleId, assigneeId, instructions, dueDate, sendEmailNotification)**

Assigns a review task related to a draft article.

## Signature

```
public static Void assignDraftArticleTask(String articleId, String assigneeId, String instructions, Datetime dueDate, Boolean sendEmailNotification)
```

## Parameters

*articleId*

Type: String

*assigneeId*

Type: String

*instructions*

Type: String

*dueDate*

Type: Datetime

*sendEmailNotification*

Type: Boolean

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';
String assigneeId = '';
String instructions = 'Please review this draft.';
Datetime dueDate = Datetime.newInstanceGmt(2012, 12, 1);
KbManagement.PublishingService.assignDraftArticleTask(articleId, assigneeId, instructions,
    dueDate, true);
```

## **assignDraftTranslationTask(articleVersionId, assigneeId, instructions, dueDate, sendEmailNotification)**

Assigns a review task related to a draft translation.

## Signature

```
public static Void assignDraftTranslationTask(String articleVersionId, String assigneeId,
String instructions, Datetime dueDate, Boolean sendEmailNotification)
```

## Parameters

*articleVersionId*

Type: String

*assigneeId*

Type: String

*instructions*

Type: String

*dueDate*

Type: Datetime

*sendEmailNotification*

Type: Boolean

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';
String assigneeId = 'Insert assignee ID';
String instructions = 'Please review this draft.';
Datetime dueDate = Datetime.newInstanceGmt(2012, 12, 1);
KbManagement.PublishingService.assignDraftTranslationTask(articleId, assigneeId,
instructions, dueDate, true);
```

## **cancelScheduledArchivingOfArticle(articleId)**

Cancels the scheduled archiving of an online article.

## Signature

```
public static Void cancelScheduledArchivingOfArticle(String articleId)
```

## Parameters

*articleId*  
Type: String

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';  
KbManagement.PublishingService.cancelScheduledArchivingOfArticle (articleId);
```

## **cancelScheduledPublicationOfArticle (articleId)**

Cancels the scheduled publication of a draft article.

## Signature

```
public static Void cancelScheduledPublicationOfArticle(String articleId)
```

## Parameters

*articleId*  
Type: String

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';  
KbManagement.PublishingService.cancelScheduledPublicationOfArticle (articleId);
```

## **completeTranslation (articleVersionId)**

Puts a translation in a completed state that is ready to publish.

## Signature

```
public static Void completeTranslation(String articleVersionId)
```

## Parameters

*articleVersionId*  
Type: String

## Return Value

Type: Void

## Example

```
String articleVersionId = 'Insert article ID';  
KbManagement.PublishingService.completeTranslation(articleVersionId);
```

## **deleteArchivedArticle(articleId)**

Deletes an archived article.

## Signature

```
public static Void deleteArchivedArticle(String articleId)
```

## Parameters

*articleId*  
Type: String

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';  
KbManagement.PublishingService.deleteArchivedArticle(articleId);
```

## **deleteArchivedArticleVersion(articleId, versionNumber)**

Deletes a specific version of an archived article.

## Signature

```
public static Void deleteArchivedArticleVersion(String articleId, Integer versionNumber)
```

## Parameters

*articleId*  
Type: String

*versionNumber*  
Type: Integer

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';  
Integer versionNumber = 1;  
KbManagement.PublishingService.deleteArchivedArticleVersion(articleId, versionNumber);
```

## **deleteDraftArticle (articleId)**

Deletes a draft article.

## Signature

```
public static Void deleteDraftArticle(String articleId)
```

## Parameters

*articleId*  
Type: String

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';  
KbManagement.PublishingService.deleteDraftArticle(articleId);
```

## **deleteDraftTranslation (articleVersionId)**

Deletes a draft translation.

## Signature

```
public static Void deleteDraftTranslation(String articleVersionId)
```

## Parameters

*articleVersionId*  
Type: String

## Return Value

Type: Void

## Example

```
String articleVersionId = 'Insert article ID';  
KbManagement.PublishingService.deleteDraftTranslation (articleVersionId);
```

## **editArchivedArticle (articleId)**

Creates a draft article from the archived master version and returns the new draft master version ID of the article.

## Signature

```
public static String editArchivedArticle(String articleId)
```

## Parameters

*articleId*  
Type: String

## Return Value

Type: String

## Example

```
String articleId = 'Insert article ID';  
String id = KbManagement.PublishingService.editArchivedArticle(articleId);
```

## **editOnlineArticle (articleId, unpublish)**

Creates a draft article from the online version and returns the new draft master version ID of the article. Also, unpublishes the online article, if *unpublish* is set to `true`.

## Signature

```
public static String editOnlineArticle(String articleId, Boolean unpublish)
```

## Parameters

*articleId*  
Type: String

*unpublish*  
Type: Boolean

## Return Value

Type: String

## Example

```
String articleId = 'Insert article ID';  
String id = KbManagement.PublishingService.editOnlineArticle (articleId, true);
```

## **editPublishedTranslation(articleId, language, unpublish)**

Creates a draft version of the online translation for a specific language and returns the new draft master version ID of the article. Also, unpublishes the article, if set to `true`.

## Signature

```
public static String editPublishedTranslation(String articleId, String language, Boolean  
unpublish)
```

## Parameters

*articleId*

Type: String

*language*

Type: String

*unpublish*

Type: Boolean

## Return Value

Type: String

## Example

```
String articleId = 'Insert article ID';  
String language = 'fr';  
String id = KbManagement.PublishingService.editPublishedTranslation(articleId, language,  
true);
```

## **publishArticle(articleId, flagAsNew)**

Publishes an article. If *flagAsNew* is set to `true`, the article is published as a major version.

## Signature

```
public static Void publishArticle(String articleId, Boolean flagAsNew)
```

## Parameters

*articleId*

Type: String

*flagAsNew*

Type: Boolean

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.publishArticle(articleId, true);
```

## **restoreOldVersion(articleId, versionNumber)**

Creates a draft article from an existing online article based on the specified archived version of the article and returns the article version ID.

## Signature

```
public static String restoreOldVersion(String articleId, Integer versionNumber)
```

## Parameters

*articleId*

Type: String

*versionNumber*

Type: Integer

## Return Value

Type: String

## Example

```
String articleId = 'Insert article ID';
String id = KbManagement.PublishingService.restoreOldVersion (articleId, 1);
```

## **scheduleForPublication(articleId, scheduledDate)**

Schedules the article for publication as a major version. If the specified date is null, the article is published immediately.

## Signature

```
public static Void scheduleForPublication(String articleId, Datetime scheduledDate)
```



## Parameters

*articleId*

Type: String

*scheduledDate*

Type: Datetime

## Return Value

Type: Void

## Example

```
String articleId = 'Insert article ID';
Datetime scheduledDate = Datetime.newInstanceGmt(2012, 12, 1, 13, 30, 0);
KbManagement.PublishingService.scheduleForPublication(articleId, scheduledDate);
```

## **setTranslationToIncomplete(articleVersionId)**

Sets a draft translation that is ready for publication back to “in progress” status.

## Signature

```
public static Void setTranslationToIncomplete(String articleVersionId)
```

## Parameters

*articleVersionId*

Type: String

## Return Value

Type: Void

## Example

```
String articleVersionId = 'Insert article ID';
KbManagement.PublishingService.setTranslationToIncomplete(articleVersionId);
```

## **submitForTranslation(articleId, language, assigneeId, dueDate)**

Submits an article for translation to the specified language. Also assigns the specified user and due date to the submittal and returns new ID of the draft translation.

## Signature

```
public static String submitForTranslation(String articleId, String language, String assigneeId, Datetime dueDate)
```

## Parameters

*articleId*  
Type: String

*language*  
Type: String

*assigneeId*  
Type: String

*dueDate*  
Type: Datetime

## Return Value

Type: String

## Example

```
String articleId = 'Insert article ID';  
String language = 'fr';  
String assigneeId = 'Insert assignee ID';  
Datetime dueDate = Datetime.newInstanceGmt(2012, 12,1);  
String id = KbManagement.PublishingService.submitForTranslation(articleId, language,  
assigneeId, dueDate);
```

# INDEX

## A

- Article
  - DataCategorySelection 12
- ArticleType component
  - Channel Layout 135
  - Layout 133, 144

## C

- Calls
  - describeDataCategoryGroups() 70, 73, 77–78, 81
  - describeKnowledge() 69
- Categories filtering (WITH DATA CATEGORY) 163–164
- Channel Layout (for article types) 135
- Classes
  - KbManagement.PublishingService 177
- Components
  - Article Type 136
  - ArticleType 130
  - Channel Layout (for article types) 135
  - ExternalDataSource 157
  - KnowledgeSettings 139
  - Layout (for article types) 133, 144
  - SearchLayouts 153
  - SearchSettings 150
  - SynonymDictionary 155

## D

- Data categories
  - filtering 166
- Data categories, filtering in SOSL 170
- Data Category Selection 165
- DataCategorySelection article 12
- describeDataCategoryGroups() call 70, 73, 77–78, 81
- describeKnowledge() call 69

## E

- ExternalDataSource component 157

## K

- KbManagement.PublishingService
  - class 177
- KnowledgeSettings component 139

## L

- Layout component (for article types) 133, 144

## N

- NewsFeed object 42

## O

- Objects
  - NewsFeed 42
  - ReportFeed 52
  - UserProfileFeed 62

## Q

- query() call
  - SOQL overview 162

## R

- Relationship queries
  - data categories 168
- ReportFeed object 52

## S

- Salesforce Knowledge
  - knowledgeManagement/settings 90
- Search
  - suggestedSearchQueries 113
  - suggestions 106
  - suggestTitleMatches 111
- SearchLayouts component 153
- SearchSettings components 150
- SELECT
  - UPDATE 163
  - WITH 163
  - WITH DATA CATEGORY 164–165
- SOQL
  - overview 162
- SOSL
  - about 168
  - WITH DATA CATEGORY clause 170
  - WITH SNIPPET clause 172
- Suggested Article Titles 111
- Suggested Queries 113
- Suggested Records 106
- SynonymDictionary component 155

## U

- UPDATE TRACKING 163, 169
- UPDATE VIEWSTAT 163, 170

## Index

UserProfileFeed object [62](#)

## W

WITH [163](#)

WITH DATA CATEGORY [164](#)

WITH DATA CATEGORY clause, SOSL [170](#)

WITH SNIPPET clause, SOSL [172](#)