



PUBLIC
2021-11-08

SAP Web IDE Full-Stack

Content

- 1 SAP Web IDE Full-Stack. 5**
- 2 What is SAP Web IDE. 6**
 - 2.1 Features and Benefits. 8
 - 2.2 User-Centric Customizability and Session Persistence. 10
 - 2.3 Collaborative Development. 10
 - 2.4 Assisted Development. 11
- 3 What's New for SAP Web IDE. 12**
 - 3.1 2018 SAP Web IDE (Archive). 24
 - 3.2 2017 SAP Web IDE (Archive). 44
 - 3.3 Introducing SAP Web IDE Full-Stack. 51
- 4 Getting Started. 53**
 - 4.1 Open SAP Web IDE. 55
 - 4.2 Connect to Cloud Foundry Services. 56
 - Troubleshooting. 56
 - 4.3 Connect to ABAP Systems. 57
 - Requirements for Connecting to ABAP Systems. 60
 - 4.4 Neo and Cloud Foundry Regions. 61
 - 4.5 Migrate your Projects. 63
 - Differences Between SAP Web IDE Versions. 63
 - Enable SAP Web IDE Full-Stack. 64
 - Migration Procedure. 68
- 5 Setting User Preferences. 70**
- 6 SAP Web IDE Basics. 72**
 - 6.1 Navigating SAP Web IDE. 72
 - 6.2 Working in the Workspace. 74
 - 6.3 Status Bar. 77
 - 6.4 Workspace Manager. 78
 - 6.5 Search Options. 79
 - Find and Replace in an Open File. 80
 - Search in the Workspace. 81
 - Find References. 84
 - 6.6 Resizing Panes. 85
 - 6.7 Keyboard Shortcuts. 85

	Customize Keyboard Shortcuts.	88
7	Developing.	90
7.1	Developing Web Applications.	91
	Creating Projects.	92
	Importing Projects.	104
	Customizing Your Project.	109
	Modifying the Application Descriptor Configuration.	121
	Developing Applications.	129
	Developing UI Libraries.	252
	Developing Application Tests.	253
	Layout Editor.	256
	Annotation Modeler.	300
	Storyboard.	327
	Extending an SAP Fiori Application with SAPUI5 Visual Editor.	333
	Using Source Control (Git).	363
	Running Applications in Development Mode.	391
	Building Applications.	405
	Deploying Applications.	414
	Extending SAPUI5 Applications.	423
7.2	Developing Multitarget Applications.	438
	Inside an MTA Descriptor.	440
	Setting Up Application Projects.	449
	Using the HTML5 Application Repository in a Multitarget Application.	453
	Developing SAP HANA Database (HDB) Modules.	457
	Developing Node.js Modules.	467
	Developing Java Modules.	478
	Developing HTML5 Modules.	490
	Developing SAP S/4HANA Service Extensions.	503
	Developing Business Applications.	504
	Adding a Service Dependency to your MTA Project.	505
	Packaging and Deploying Applications to Production Systems.	506
8	SAP Web IDE Extensions.	508
8.1	Enable SAP Web IDE Extensions.	510
9	Extending SAP Web IDE.	512
10	SAP Web IDE Personal Edition.	513
10.1	Installation and Setup.	514
	Install SAP Web IDE Personal Edition.	514
	Start SAP Web IDE Personal Edition.	518
	Upgrade SAP Web IDE Personal Edition.	520

	Uninstall SAP Web IDE Personal Edition.	521
10.2	Import the Git Server Certificate into the JVM.	521
10.3	Configure the Orion Application Server.	522
10.4	Connect Remote Systems in SAP Web IDE Personal Edition.	523
10.5	Connect to an External Git Repository.	526
10.6	Run Applications with Multiple SAPUI5 Versions.	527
11	Security.	528
11.1	Architectural Overview.	528
11.2	User Authentication and Authorization.	530
	Authentication.	531
	Assign Users Permission for SAP Web IDE.	532
	Browser Security.	536
	Transport Security.	536
11.3	Data Protection and Privacy.	537
	Export Workspaces.	538
	Delete User Data.	539
	Manage User Service Instances.	540
	Data Protection and Privacy for Previous Versions.	541
11.4	Connection to External Systems.	546
11.5	Principal Propagation.	548
11.6	Secure Programming Guide.	549
11.7	Auditing and Logging Information.	549
12	Troubleshooting.	551
12.1	Archive Import Troubleshooting.	551
12.2	Deployment to SAPUI5 ABAP Repository Troubleshooting.	552
12.3	Deployment to SAP BTP Troubleshooting.	554
12.4	Git Troubleshooting.	555
12.5	Grunt Build Troubleshooting.	558
12.6	Running Applications Troubleshooting.	559
12.7	Remote Connectivity Troubleshooting.	562
13	Known Issues.	563
14	Third-Party Restrictions.	570

1 SAP Web IDE Full-Stack



[What's New for SAP Web IDE \[page 12\]](#)

Learn about SAP Web IDE new features.



[What is SAP Web IDE \[page 6\]](#)

Understand how to use SAP Web IDE and learn about its key features and benefits for developers.



[Getting Started \[page 53\]](#)

Learn how to set up and start SAP Web IDE.



[Developing \[page 90\]](#)

See how SAP Web IDE supports key stages of the development life cycle.



[SAP Web IDE Extensions \[page 508\]](#)

Understand how to use SAP Web IDE extensions.



[Extend SAP Web IDE \[page 512\]](#)

Extend SAP Web IDE functionality by developing custom plugins, templates, and external commands.



[Security \[page 528\]](#)

Understand the security landscape.



[SAP Web IDE Personal Edition \[page 513\]](#)

Install a local instance of SAP Web IDE on your desktop.

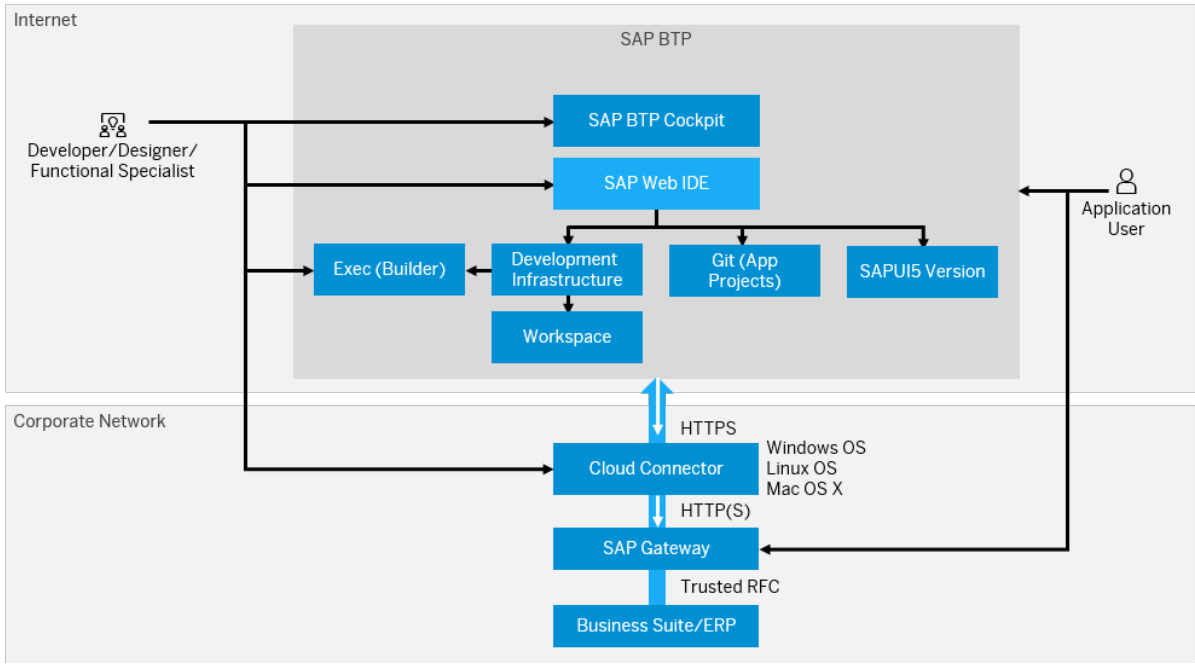
2 What is SAP Web IDE

SAP Web IDE is a browser-based IDE consisting of integrated parts that interact with each other and with an SAP system.

SAP Web IDE Full-Stack streamlines the end-to-end application lifecycle – easily develop, test, build, deploy, and extend role-based, consumer-grade apps for business users. Create applications rapidly and deliver an outstanding user experience. Developers can extend or build SAP Fiori apps, create new SaaS solutions, extend S/4HANA cloud services, develop hybrid mobile applications and build IoT apps for SAP Leonardo, using the UI development toolkit for HTML5 (SAPUI5) for desktop and mobile devices, SAP HANA toolset, and Java programming language and technologies. Since SAP Web IDE Full-Stack runs on SAP Business Technology Platform (BTP), it needs no installation and allows you to integrate with other services that run on the platform – such as SAP Fiori Cloud apps, Git integration, mobile services, IoT services, and more.

Architecture

The following diagram provides high level typical architecture for SAP Web IDE Full-Stack.



Component	Description
SAP BTP	SAP BTP enables customers and partners to rapidly build, deploy, and manage cloud-based enterprise applications that complement and extend your SAP or non-SAP solutions, either on-premise or on-demand.

Component	Description
SAP BTP cockpit	Central point for managing all activities associated with your SAP BTP account and for accessing key information about your applications. For more information, see SAP BTP cockpit .
SAP Web IDE application	Integrated development environment used to create or extend SAP UI5 or SAP Fiori applications.
Git	Revision control and source code management system.
SAPUI5	User interface technology that is used to build and adapt client applications. For more information, see UI development toolkit for HTML5 - Demo Kit
Cloud Connector	Allows SAP Web IDE and SAP BTP to connect to an on-premise system securely and with minimal configuration effort. For more information, see Cloud Connector .
SAP Gateway	Provides a simple way to connect SAP Web IDE to an external SAP system with access to OData functionality.

i Note

When working in SAP Web IDE, the following operations may be processed by our partner Infrastructure-as-a-Service (IaaS) providers:

- Code completion
- Code validation

These operations may involve transfer and process of data in different regions.

The list of operations is subject to change without prior notice.

Who is it for?

SAP Web IDE is a flexible tool for developers who want to dive right into the code editor without having to spend time configuring and administering the development environment.

The tool is aimed at developers who need a modern and secure environment to create new or extend existing SAP Fiori, SAPUI5, or hybrid applications. Developers are provided with a comprehensive set of tools, including strong code editors with templates, wizards, beautifier capabilities, code completion, code snippets, code validation, code checking, WYSIWYG, and many more features.

i Note

SAP Web IDE does not support touch capabilities.

2.1 Features and Benefits

SAP Web IDE offers many key features that can be used by developers.

The following table describes the main features of SAP Web IDE.

Key Feature	Use
Zero Installation and Maintenance	Available as a cloud service on SAP BTP; SAP Web IDE only requires a browser.
High Productivity Tools	SAP Web IDE includes tools that allow you to create applications easily such as: templates, wizards, sample applications, and mock data.
Business Application Development	The application builders, editors and templates in SAP Web IDE help to create business applications following SAP's best practices using the SAP BTP Application Programming Model.
Code Authoring Toolset	<p>SAP Web IDE supports free-style coding by providing robust editors with capabilities of code completion (XML, property files, JavaScript, Java, Node.js, CDS, and SAPUI5), code validation, code templates, and many more. Name and resource content-based search across all the user's files and workspaces provide additional efficiency.</p> <p>The Problems view provides information in the console about problems in the projects in your workspace.</p> <p>Our code editors also beautify file formatting for HDB, JavaScript, JSON, XML, and CSS files using the context menu.</p>
Java and Node.js Debuggers	Debug your Java and Node.js code.
Testing and Debugging	<p>Launch the application in the browser at different resolutions and in different languages, either with real or mock data. You can view the application in your mobile using QR codes.</p> <p>Mock data can be automatically generated or manually created to run the application independently for front-end and back-end development decoupling, testing, and demoing purposes.</p>
Multi-device Support	The responsive patterns and controls of SAPUI5 and the adaptive design of apps allow SAP Web IDE to run on desktops, tablets, and smartphones. The apps accommodate the resolution, image size, and scripting on-the-fly, as users switch between devices, allowing them to work how and where they want.
Application Templates	Build SAP Fiori and full-stack applications applying SAP best practice guidelines using wizards and templates.

Key Feature	Use
Visual UI Editors	<ul style="list-style-type: none"> • Layout Editor: The layout editor makes developing SAPUI5 applications faster and simpler. It has the following areas: <ul style="list-style-type: none"> ◦ Palette: Contains the SAPUI5 controls. Expand or collapse the sections by clicking the arrows, and then you can drag and drop the controls to the canvas. ◦ Canvas: Displays the content of the XML view in a way that closely corresponds to how it will appear in your finished application. ◦ Properties/Data: Shows the properties' values and data binding of a selected control. ◦ Outline pane: Displays the outline of your XML view. • SAPUI5 Visual Editor: The SAPUI5 Visual Editor allows developers to change, adapt, and extend the user interface of SAPUI5 applications. • Storyboard: The storyboard provides a visual representation of the application's UI including its pages (views), navigations, and the services and entities that it uses. • CDS graphical editor: SAP Web IDE offers data modelers the graphical modeling capabilities that they require to model artifacts. These artifacts define the data persistence objects using the Core Data Services (CDS).
Extensibility Capabilities	<ul style="list-style-type: none"> • Extend SAP Fiori applications through a visual extensibility editor. SAP Fiori, cloud edition leverages SAP Web IDE to extend and customize SAP Fiori apps. • Develop SAP S/4HANA Service extensions: SAP Web IDE enables you to create S/4 HANA extensions. You implement an application with an OData service that extends an existing S/4HANA service and exposes additional data from a different source, such as an SAP HANA database. You can do this using the SAP BTP Business Application Development Tools.
Extensions SAP Web IDE	<p>Extensions are the building blocks of SAP Web IDE. They are used to group functionality into small units.</p> <p>Extensions can expose services to provide public APIs. Extension code may include any SAP Web IDE component: a new command, template, editor, pane, or any other contribution to the SAP Web IDE application.</p> <p>You can enable extensions in SAP Web IDE to extend the existing functionality.</p> <p>Likewise, you can create your own extensions in SAP Web IDE to enrich the tool.</p>

2.2 User-Centric Customizability and Session Persistence

SAP Web IDE shares a common workspace for code; however, all individual user settings, including any configuration changes or window resizing, are automatically saved when you exit. Your personal preferences persist, despite the workspace being shared.

Key customizations include:

- Workspace View** If you have changed the size of the workspace, the console, or various panes by dragging the splitter, the splitter's new position is saved automatically and your views are restored when you reopen SAP Web IDE.
- Reset View** You can reset to the default settings by choosing **View > Reset to Default**.
- Custom Code Readability** Code beautification support allows you to apply readability standards to the files that you work in. Define custom beautification settings and apply them to files as needed.
- Inline Error Validation** You can configure custom code validation checks as required so issues can be displayed and corrected inline.
- Cache Behavior** Caching can be controlled by implementing custom properties.

2.3 Collaborative Development

SAP Web IDE is an online development space where an application's development team can work together across time zones and geographic regions to develop project deliverables, using a shared repository.

2.3.1 Git Client Plugin

Git is the revision control and source code management (SCM) system used in SAP Web IDE.

Related Information

[Using Source Control \(Git\) \[page 363\]](#)

2.3.2 Application Testing

Use SAP Web IDE to run applications that are under development, allowing you to instantaneously evaluate progress, test functionality, and preview localizations.

You can run applications using a simulator within SAP Web IDE. Choose from predefined simulators to preview the application in SAP Web IDE: desktop, tablet, phone, or custom.

2.4 Assisted Development

A collection of dynamic interactive features, code completion, and API reference support, which facilitate development by expediting coding and testing.

You can also start your development based on predefined templates.

2.4.1 Code Completion

Use code completion to assist in writing JavaScript (for SAPUI5 namespaces) or XML code. Inline code completion popups provide contextually-aware API reference support, and make hints available to the editor directly where they are needed.

The editor completes words, code fragments, or entire SAPUI5 objects (including methods, properties, and events) based on the current context, and the context of other similar words within the same component. The selected code fragment is then entered into the cursor location in syntactically correct format.

To generate an appropriate list of suggestions, SAP Web IDE parses and analyzes the context by reviewing:

- The parent node
- The current node
- Any prefixes

SAP Web IDE then determines which suggestion is most appropriate: whether to suggest a namespace, a control, an attribute, event, value, or in the case of XML, only syntax. You can also select entire code snippets or custom objects. All suggestions appear as a filtered and sorted list, with an icon that indicates the type of code to be injected, allowing you to quickly find and select the correct option.

2.4.2 Integrated and Dynamic API Reference Support

API reference information is embedded in SAP Web IDE, providing contextually-aware reference information on demand when you are writing code.

API reference information is available in a tooltip when you hover over one of the options displayed when using code completion.

3 What's New for SAP Web IDE

Core Components, 2020


Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Cloud Foundry	Role Editor	When creating a new role, you have the option of selecting Create config file . The option is off by default. Selecting this option creates a role config with a filename that corresponds to the new role name. You can choose to keep this name, enter a different name, or choose an existing role config file by using the Browse button. You can check for the role config file association in the Schema references tab.	Recommended	Changed	2020-10-29
SAP Web IDE	Extension Suite - Development Efficiency	Cloud Foundry	Role Editor	The new Role Editor allows you to create new roles and edit existing roles. You can grant or revoke roles or object, schema, analytic and system privileges from new or existing roles. To access the Role Editor, right-click any <code>.hdbrole</code> file and click Open Role Editor . See Developing Database Artifacts .	Recommended	New	2020-06-28
SAP Web IDE	Extension Suite - Development Efficiency	Cloud Foundry	HDB Modules	HDB Modules now support SAP HANA Cloud.	Info only	New	2020-03-27
SAP Web IDE	Extension Suite - Development Efficiency	Cloud Foundry	HDB Modules	You can now test the connection to a database in the SAP HANA Service Connection tool. See Connect to an External HDB Service .	Info only	New	2020-03-27

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Cloud Foundry	Build Tools	The default HTML5 module build now uses UI5 build tools instead of the Grunt build.	Recommended	Change	2020-03-15
SAP Web IDE	Extension Suite - Development Efficiency	Cloud Foundry	HDB Modules	You can now configure a connection to an external HDB service outside of your current schema. See Connect to an External HDB Service .	Info only	New	2019-11-21
SAP Web IDE	Extension Suite - Development Efficiency	Cloud Foundry	MTA Build	The current Build option for MTA projects (that uses the Multitarget Application Archive Builder) is deprecated. As of May 31, 2020, this build option will no longer be available. Make sure you switch to the new Cloud MTA Build Tool by that date. See this blog and Packaging and Deploying Applications to Production Systems .	Required	Announcement	2019-11-21

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Annotation modeler	<p>Annotation modeler provides the following new features:</p> <ul style="list-style-type: none"> You can now identify and work with experimental annotations and annotation properties from the vocabularies. You can also identify deprecated annotations and annotation properties. You can now directly navigate to a referenced annotation by using an icon next to the value assigned in the annotation path. See Edit Annotations You can now define an annotation property of complex type as a path that references an annotation of the same type. Complex annotation properties that can be defined as a path to an annotation of the same type (such as SelectionVariant and DataPoint properties in UI.KPI annotation or SelectionVariant and PresentationVariant properties in SelectionPresentationVariant) now have a combo box to define properties inline (blank) or as a path. 	Info only	New	2019-10-15
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	Deploying HTML5 Modules	After making changes to HTML5 modules in a multitarget application, you can now deploy all HTML5 modules to SAP BTP to see the changes quickly. See Deploy HTML5 Modules .	Info only	New	2019-09-12
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Annotation modeler	Annotation modeler now also supports applying annotations to the following targets: Entity Container, Functions, Actions, Action Imports. See Architecture .	Info only	New	2019-08-15

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	Running Modules	When running a Java, Node.js or HTML5 module in Cloud Foundry, you can override the module's settings in the <code>mta.yaml</code> file by configuring the required settings in an extension file that will be used when running the module. See Run a Java Module , Run Node.js Modules , and Run an HTML5 Module in the Cloud Foundry Environment .	Info only	New	2019-08-01
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	UI based on Node.js Module	You can now develop UI based on a Node.js module that exposes an OData V4 service. See Create UI Based on Your OData Service .	Info only	New	2019-07-08
SAP Web IDE	Extension Suite - Development Efficiency	Neo	i18n Validation	Your i18n files are now validated in your code editor according to the SAP Fiori best practices. See Validation of i18n Files .	Recommended	New	2019-06-06
SAP Web IDE	Extension Suite - Development Efficiency	Neo	UI5 Build	The UI5 build now replaces the Grunt build for generating new SAP Fiori projects deployed to the Neo or ABAP environment. See Run a UI5 Build .	Recommended	New	2019-06-06

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Productive Accounts	All the features that were only available for trial accounts in the last release, are now available in productive accounts as well.	Info only	New	2019-04-28
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Git	If you have been assigned the <code>DiScpGitAdministrator</code> role, you can now delete local and remote Git branches directly from SAP Web IDE.	Info only	New	2019-04-11
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	SAPUI5 Libraries	You can now develop SAPUI5 libraries when creating a project for the Cloud Foundry environment or when adding an HTML5 module to your MTA project. See Develop an SAPUI5 Library .	Info only	Changed	2019-04-11
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Builders	(For Trial only) You no longer need to install the builder application to build your project. See Select a Cloud Foundry Space .	Recommended	New	2019-04-02

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of	
SAP Web IDE	Extension Suite - Development Efficiency	Neo	MTA Deployment	(For Trial only) You can use MTA extensions for deploying MTA archives to the SAP Cloud Foundry environment. See Packaging and Deploying Applications to Production Systems .		New	2019-04-02	
SAP Web IDE	Extension Suite - Development Efficiency	Neo	MTA Archive Location	When you build an MTA project, the resulting MTA archive file (<ID>_<version>.mtar file) is generated within the project's <i>mta_archives</i> folder. See Packaging and Deploying Applications to Production Systems .		Changed	2019-04-02	
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Quick Fix	If your project uses JavaScript or JAVA LSP validators, you are now provided with a list of possible solutions for errors and warnings found in the selected line of code.		New	2019-04-02	
								
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Rename Project	You can now rename a project from the context menu in the Workspace.		New	2019-03-28	

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	Creating a Service Instance	You can now add a service dependency to your MTA project and specify its configuration parameters using the SAP BTP Service wizard. See Adding a Service Dependency to your MTA Project .		New	2019-03-28
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	New Project Wizard	When creating a project in the Cloud Foundry environment, module creation is now part of the main flow in the wizard.		Changed	2019-03-28
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Generating Stable IDs	SAP Web IDE can now generate stable IDs automatically when <code>flexEnabled</code> is activated. See Validation of Stable IDs .		New	2019-03-14
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	Grunt Build	The <code>grunt-sapui5-bestpractice-build</code> plugin has been updated to version 1.3.65. See Run a Grunt Build .		Changed	2019-03-14

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	Batch Processing	You can now enable or disable batch processing for OData services from the <i>Descriptor Editor</i> . See Batch Control for OData Services .		Changed	2019-03-14
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	HTML5 Modules	You can now view OData backend annotations when creating an HTML5 module from the ABAP catalog within an SAP BTP Service. See Create UI Based on an SAP BTP Service .		Changed	2019-03-14
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Resource Manager	In the Resource Manager, you can now update a service instance from the runtime environment based on the design time configuration. See Resource Manager .		New	2019-02-28
SAP Web IDE	Extension Suite - Development Efficiency	Cloud Foundry	Code Editors	Code editors for HANA-related artifacts now highlight keywords.		New	2019-02-28

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Service Provisioning	SAP Web IDE now automatically creates the required service instance for the <code>com.sap.xs.uaa</code> resource type. See MTA Service Provisioning .		New	2019-02-28
SAP Web IDE	Extension Suite - Development Efficiency	Neo	SAPUI5 Visual Editor	You can now use an adaptation project and the SAPUI5 Visual Editor for Overview Page applications. See Create an Adaptation Project and Extending SAP Fiori Elements-Based Applications .		New	2019-02-28
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Search Capabilities	You can now perform case-sensitive search in the Search pane. See Search in the Workspace .		New	2019-02-28
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Compare Editor	You can now compare any two files from your workspace in read-only mode. Select any 2 files in your workspace, right-click, and select Compare .		New	2019-02-28

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Consuming OData services	You can now create an HTML5 module in your multitarget application that displays data from an SAP BTP OData service. The feature is available for multitarget applications that use the HTML5 Application Repository. See Create a Module Based on an SAP BTP OData Service .		New	2019-01-31
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Git Permissions	You can now assign a user to the <code>DiscpGitAdministrator</code> role. This provides the user with <code>manageGit</code> permissions. In SAP Web IDE, the user will now be able to push commits made by other users. See Assign Users Permission for SAP Web IDE .		New	2019-01-31
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Code completion snippets	The Language Service Protocol used in SAP Web IDE now supports code completion snippets. This way, you will be able to insert predefined snippets into your code according to the language best practice guidelines.		New	2019-01-31
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Code editors	The SAP Web IDE code editors now support beautification of SAP HANA database artifacts. Currently, only the <code>.json</code> format is supported. In the future, the <code>.xml</code> format will be supported as well.		New	2019-01-31

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Debugging	You can now debug and rerun Spring boot applications using SAP Web IDE. See Debug Java Modules .		New	2019-01-31
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Terminology Update	In the context of SAP Web IDE feature and plugin development and enablement, the term SAP Web IDE <i>feature</i> is now being called SAP Web IDE <i>extension</i> . You can create SAP Web IDE extensions and run them. Delivered SAP Web IDE extensions can still be enabled from the Preferences perspective on the Extensions page. See Concept and Extension Structure in the SAP Web IDE SDK		Changed	2019-01-31
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Compare Editor	You can now manually resolve conflicts when the automated merge cannot make a determination using the 3-Way Merge comparison editor. See Compare and Merge Code .		New	2019-01-17
SAP Web IDE	Extension Suite - Development Efficiency	Neo	SAPUI5 Visual Editor	To create tickets related to issues with the SAPUI5 Visual Editor and Adaptation Project, use the component names CA-WDE-VE and CA-WDE-ADP respectively.		New	2019-01-17

Technical Component	Capability	Environment	Title	Description	Action	Type	Available as of
SAP Web IDE	Extension Suite - Development Efficiency	Neo	SA-PUI5 Visual Editor	A debugger is available with the controller extension. See Controller Extensions .		New	2019-01-17
SAP Web IDE	Extension Suite - Development Efficiency	Neo Cloud Foundry	Run on Cloud Foundry environment	<p>When setting up a new run configuration for an HTML5 module, you can now select the target environment on which you want to preview your module.</p> <ul style="list-style-type: none"> For modules that are part of a multi-target application with HTML5 Application Repository – the default target environment is Cloud Foundry. For modules that are part of a multi-target application without HTML5 Application Repository – the default target environment is Neo. For existing run configurations – the default target environment remains Neo. <p>See Run an HTML5 Module in Cloud Foundry Environment.</p>		New	2019-01-17
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Grunt Best Practice Plugin Update	The <code>grunt-sapui5-bestpractice-build</code> plugin has been updated to version 1.3.64. See Run a Grunt Build .		Changed	2019-01-17
SAP Web IDE	Extension Suite - Development Efficiency	Neo	Automated Tests	The sample automated test content generated in a new SAPUI5 project is now based on SAPUI5 version 1.60 or above.		New	2019-01-03

3.1 2018 SAP Web IDE (Archive)

20 December 2018 – SAP Web IDE

New

You can now sync the metadata of your applications source file and the respective annotation files with the backend version.

Open the descriptor editor from your manifest.json file, go to the *Data Sources* tab, and click *Sync Metadata*.

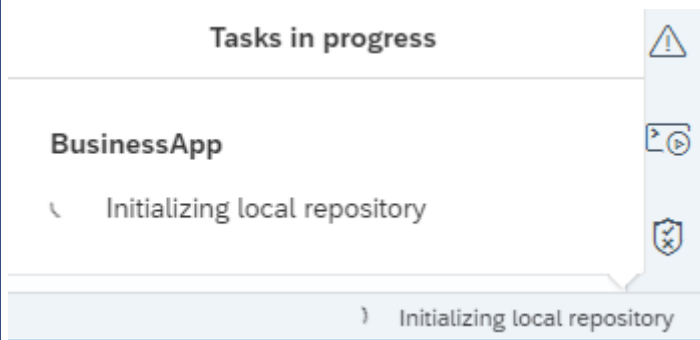
For more information, see [Data Sources Tab Options \[page 123\]](#).

6 December 2018 – SAP Web IDE

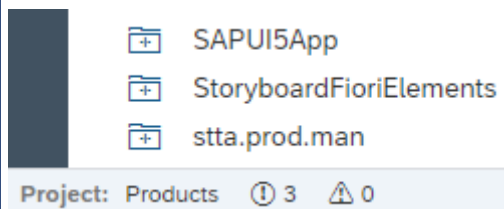
New

We have added a status bar to the IDE so that you can see which processes are running.

You can see the status of each process per project and cancel any process from the popup.



The selected project is shown beneath the workspace, and if there are any problems or warnings, they are indicated in the status bar as well. Double-clicking these icons will open the Problems View.



Changed

The project settings and user preferences pages have been updated to simplify their use.

New

When you generate a project based on the [SAPUI5 Application](#) template, a test structure is automatically generated. This structure includes OPA and unit tests using best practices that can be further enhanced.

22 November 2018 – SAP Web IDE

Changed

When you create controller extensions for list report and object page applications using the SAPUI5 Visual Editor, new methods are available for further reuse. You can use these methods to override the base methods existing in the applications.

8 November 2018 - SAP Web IDE

Changed

You can now reach new data sources by creating a destination from within SAP Web IDE instead of creating it from the cockpit.

For more information, see [Create New Data Source \[page 99\]](#).

25 October 2018 - SAP Web IDE

New

A new capability was added to the Multi-Target application, to enable developers to store their static resources centrally on the HTML5 Application Repository.

For more information, see [Using the HTML5 Application Repository in a Multitarget Application \[page 453\]](#).

Enhancement

Single Sign-On for Deployment to SAP BTP

- SAP Web IDE users who are not defined as subaccount members can now deploy SAP Fiori applications to their subaccount.
- Users will no longer be prompted for credentials when deploying SAP Fiori applications from SAP Web IDE to their subaccount.

For more information, see [Deploy Applications to SAP BTP \[page 418\]](#).

11 October 2018 – SAP Web IDE

New

You can now create new services and data models from the Project Explorer.

For more information, see [Working in the Workspace \[page 74\]](#).

New

You can now move or copy your projects from one workspace to another including your project settings and Git history.

For more information, see [Workspace Manager \[page 78\]](#).

Changed

The Layout Editor now supports more SAPUI5 controls.

For the list of supported controls, see [SAPUI5 Controls Supported in the Layout Editor \[page 275\]](#).

Changed

Now you can reuse a CDS model with a local dependency and simultaneously develop both the reused CDS model and the business application that is using the model in SAP Web IDE.

27 September 2018 – SAP Web IDE

Changed

The UI Adaptation editor is now called the **SAPUI5 Visual Editor**.

The UI Adaptation editor allowed you make changes to the SAPUI5 applications during application runtime.

With enhanced capabilities, the **SAPUI5 Visual Editor** enables you to create an adaptation project for SAP Fiori Elements-based applications which you can adapt and extend during design time.

13 September 2018 - SAP Web IDE

New

Terminology Update

In the context of SAP Web IDE feature and plugin development and enablement, the term SAP Web IDE "feature" is now being called SAP Web IDE "extension". You can create SAP Web IDE extensions and run them. Delivered SAP Web IDE extensions can still be enabled from the [Preferences](#) perspective on the [Features \(Extensions\)](#) page.

For more information, see [Concept](#) in the *SAP Web IDE SDK*.

Changed

You can automatically generate the custom handlers for the entities defined in your service model using SAP Web IDE.

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

Changed

The annotation modeler can now use metadata containing multiple schemas.

30 August 2018 - SAP Web IDE

New

You can now create SAP HANA Database Application projects from the dedicated project template. This template creates a multi-target application (MTA) project with an SAP HANA Database (HDB) module.

To use this template, make sure you have enabled the [SAP HANA Database Development Tools](#) feature.

New

You can now create database artifacts of different supported [New Database Artifact](#) dialog for HDB modules.

New

Database artifacts that haven't yet been built are marked by the [\[Inactive\]](#) sign displayed next to the artifact's name in the project.

New

The new Resource Manager tool enables you to view information about all the resources (services) required by the modules in a multi-target application project, as well as delete the bound service instances.

To open the tool, choose the  (Resource Manager) icon at the bottom of the right sidebar.

Changed

The *Project Explorer* has new capabilities.

- You can now open the *New SAPUI5 Application* wizard from a UI container.
- You can add an entity set to a service.

For more information, see [Working in the Workspace \[page 74\]](#).

Changed

You can now use regular expressions when using the Search pane.

For more information, see [Search in the Workspace \[page 81\]](#)

16 August 2018 - SAP Web IDE

New

You can now create, build, deploy, and reuse UI libraries. This enables you to share custom controls between SAPUI5 projects in SAP Web IDE.

For more information, see [Developing UI Libraries \[page 252\]](#).

New

The workspace context menus have been reorganized to bring clarity, order and smoother user interaction.

For more information, see [New context menus experience in SAP Web IDE](#) .

Enhancement

You can now use wildcards (*, !, /) when performing a search in your workspace.

For more information, see [Search in the Workspace \[page 81\]](#).

Change

The *SAP S/4 HANA Extension Tools* feature has been removed from SAP Web IDE. The functionality provided by this feature is now covered by the **SAP BTP Business Application Development Tools** feature.

For more information, see [SAP BTP Business Application Development Tools](#).

2 August 2018 - SAP Web IDE

Enhancement

You can now add [View](#) elements to your Fiori Elements and freestyle projects from the *Project Explorer*.

For more information, see [Working in the Workspace \[page 74\]](#).

Enhancement

SAP HANA Database Development Tools

With the SAP HANA database development tools, you can:

- Develop HDB modules and database artifacts, such as core data services (CDS), calculation views, database procedures and more.
- Use the SAP HANA Deployment Infrastructure (HDI).
- Access data with the SAP HANA database explorer.

To use these tools, enable the SAP HANA Database Development Tools feature. By default, the feature is not enabled, and its related menu options, dialogs, and wizards are not available.

For more information, see [Enable SAP Web IDE Extensions \[page 510\]](#).

Enhancement

Grunt Best Practice Plugin Update

The `grunt-sapui5-bestpractice-build` plugin has been updated to version 1.3.62.

For more information, see [Run a Grunt Build \[page 409\]](#).

25 July 2018 - SAP Web IDE

New

Authorization Migration

As an account administrator, you can easily migrate SAP Web IDE user role assignments to SAP Web IDE Full-Stack.

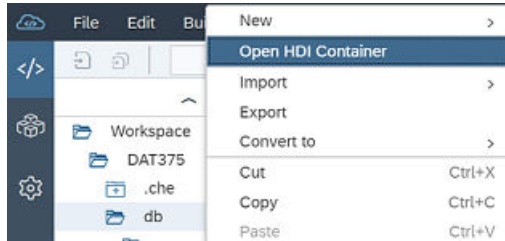
For more information, see [Authorization Migration \[page 535\]](#).

19 July 2018 - SAP Web IDE

Enhancement

Easy Access to SAP HANA Database (HDB) Module Artifacts in the Database Explorer

After building an HDB module, you can now easily access its artifacts in the database explorer by choosing *Open HDI Container* from the module context menu.



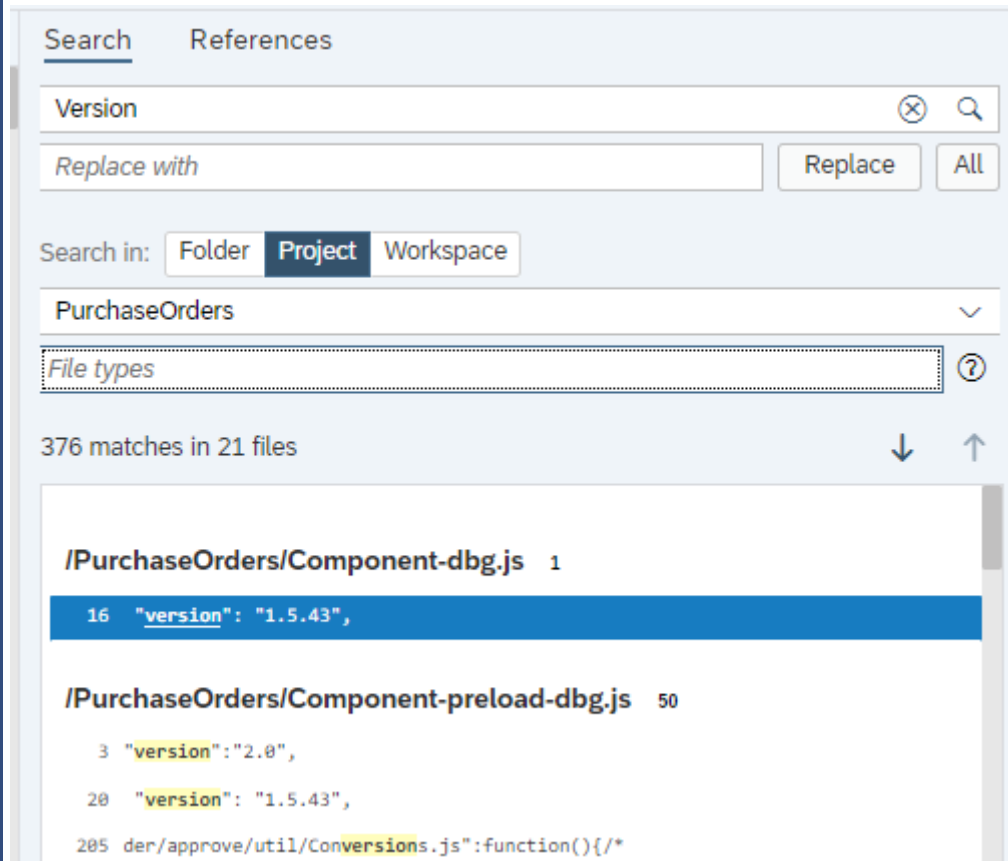
For more information, see [Developing Database Artifacts \[page 459\]](#).

Enhancement

Updated Search Pane

The Search pane has been redesigned to provide a more intuitive search experience.

You can now preview each result within the relevant file.



For more information, see [Search in the Workspace \[page 81\]](#).

Enhancement

Live Java Application Logs in the Run Console

All the log and error messages of a running Java application are now displayed in the Run console as soon as they are generated.

Enhancement

Generating Operation Hooks for Service Entities in SAP BTP Business Applications.

You can now generate entity operation hooks for selected service entities from the data model, and implement custom handlers for these hooks.

To do this, choose *Build CDS* from the project's context menu, and then **► New ► Entity Operation Hooks ►** from a Java module's context menu. A Java class with the generated entity operation hooks is created. You can implement the custom handlers for the hooks as required.

For more information, see [Generate Entity Operation Hooks](#) and [Register Custom Handlers to Hooks](#).

Enhancement

Single Sign-On for SAP Git Service Repositories

- The SAP Git service is now available also for SAP Web IDE users who are not defined as subaccount members.
 - Users will no longer be prompted for credentials when communicating from SAP Web IDE with SAP Git service in their subaccounts.
-

5 July 2018 - SAP Web IDE

Enhancement

Grunt Best Practice Plugin Update

The `grunt-sapui5-bestpractice-build` plugin has been updated to version 1.3.58 and now supports the following:

- npm 5.6.0

For more information, see [Run a Grunt Build \[page 409\]](#).

Enhancement

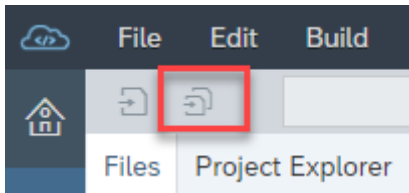
SAP Web IDE Feature (MTA Project) Optimization

Newly created projects based on the *SAP Web IDE Feature (MTA Project)* template now include a bundling and minification script that automatically optimizes feature performance after a project is built and delivered.

21 June 2018 - SAP Web IDE

New

You can now save all your open files with the click of one button.



Enhancement

You can now see a detailed migration report. The report provides information on which files and folders were added or updated in during migration. It also describes recommended next steps.

For more information, see the [Migration Procedure \[page 68\]](#).

11 June 2018 - SAP Web IDE

New

Java Hot Deployment

You can now redeploy your Java modules faster. Just run a Java module again without stopping it, and the Java hot deployment mechanism will redeploy the application on-the-fly.

Enhancement

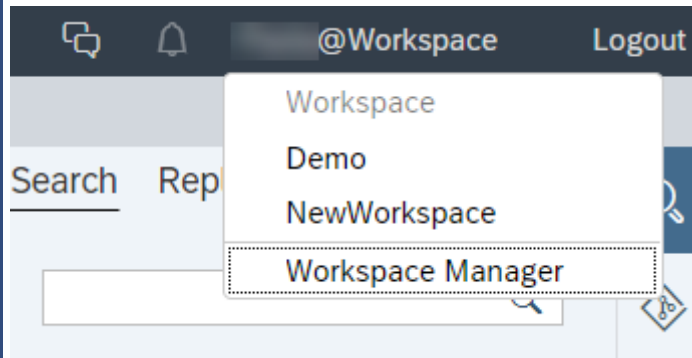
Grunt Best Practice Plugin Update

The `grunt-sapui5-bestpractice-build` plugin has been updated to version **1.3.56**.

For more information, see [Run a Grunt Build \[page 409\]](#).

Enhancement

You can now switch between workspaces from the Workspace menu.



For more information, see [Workspace Manager \[page 78\]](#).

31 May 2018 - SAP Web IDE

New

SAP BTP Business Application Development Tools

You can now develop business applications using the application programming model for SAP BTP.

The SAP BTP Business Application development tools contain a new project template and help you implement data models, services, and UIs to develop your own stand-alone business applications or extend other cloud solutions, such as SAP S/4 HANA.

To use this new feature, you need to enable the tile (shown below) in the [Preferences](#) perspective on the [Features](#) page.

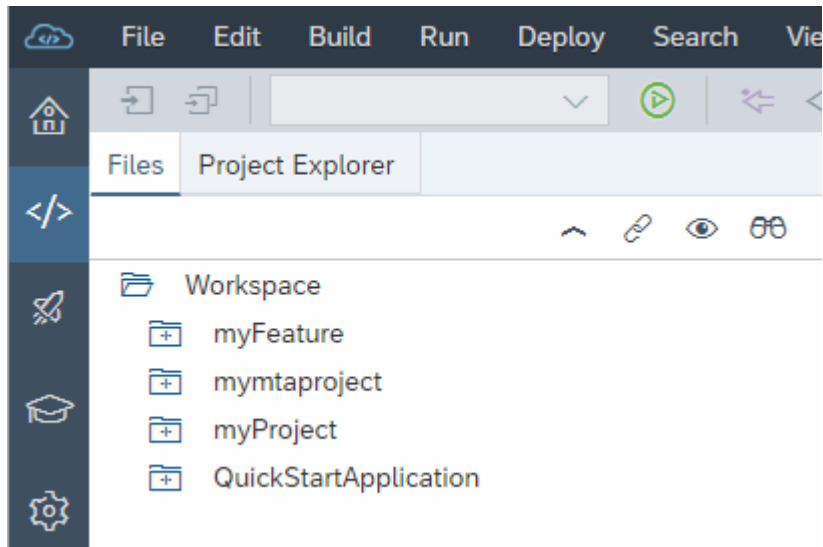
For more information, see [Business Applications](#) in the SAP BTP documentation.

New

The workspace now has two tabs:

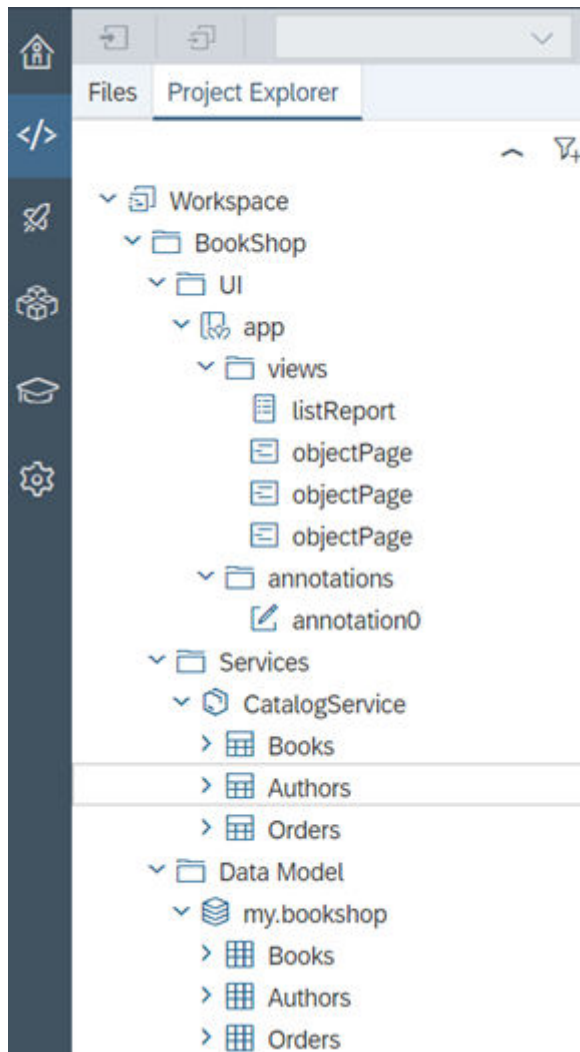
- *Files*

You can display all your project folders and files.



- *Project Explorer*

You can now use the *Project Explorer* tab to view the logical structure of your applications. You can see SAPUI5 views and fragments, CDS services and entities, and annotation files. When you double-click a node it opens the associated editor.



New

You can now add the best practice test structure (including sample tests) to your project.

For more information, see [Create Test Structures \[page 253\]](#).

Enhancement

The storyboard now displays the entity sets for freestyle views and SAP Fiori Elements List Report pages.

For more information, see [Storyboard \[page 327\]](#).

10 May 2018 - SAP Web IDE

New

You can now easily migrate your projects from SAP Web IDE to SAP Web IDE Full-Stack.

For more information, see [Migrate your Projects \[page 63\]](#).

Enhancement

You can now access a Git repository using personal access token authentication.

1. Make sure you have a personal access token for your Git server.
 2. In the *Authentication* dialog box, click *Token Authentication*, and supply your token.
-

Enhancement

You can selectively build database artifacts in SAP HANA Database (HDB) modules, thus optimizing your module development.

For more information, see [Selective Build of HDB Modules \[page 464\]](#).

Enhancement

The `grunt-sapui5-bestpractice-build` plugin has been updated to version 1.3.55 and now includes a task for generating the `manifest-bundle.zip` file, which contains the `manifest.json` and `i18n` files.

For more information, see [Run a Grunt Build \[page 409\]](#).

26 April 2018 - SAP Web IDE

Enhancement

You can now use the extension wizard to add the following extensions for SAP Fiori elements:

- List report
 - Filter
 - Action
 - Column
 - Object page
 - Action
 - Facet
 - Column
 - Header
 - Form
 - Analytical list page
 - Filter
 - Action
-

Change

Java module templates for multi-target applications *Web Application with OData V2 Support* and *Web Application with OData V4 Support* have been deprecated and replaced by the respective templates *OData V2 Service* and *OData V4 Service*. The new templates are based on the SAP BTP SDK for service development.

For more information, see [Create a Java Module \[page 480\]](#).

12 April 2018 - SAP Web IDE

New

Configure Navigation in Storyboard

Using the Storyboard perspective, you can now configure a navigation between views.

For more information, see [Configure a Navigation Between Views \[page 332\]](#).

Enhancement

Grunt Best Practice Plugin Update

The `grunt-sapui5-bestpractice-build` plugin has been updated to version **1.3.50** and now includes the following new tasks:

- Task for generation of the `CachebusterInfo.json` file. For more information, see [Application Cache Buster](#).
- Task for generation of the `changes-bundle.json` file.
- Task for building an HTML5 module inside full-stack projects. The build results are stored as part of the `.mtar` file.

For more information, see [Run a Grunt Build \[page 409\]](#).

29 March 2018 - SAP Web IDE

New

Multiple Workspaces

You can now have multiple workspaces, enabling you to group your projects in different workspaces. The Workspace Manager tool lets you do the following:

- Create and delete workspaces.
- Open another workspace in the current browser tab or in a new tab.

For more information, see [Workspace Manager \[page 78\]](#).

Enhancement

More Options in Template Customization Step

In the SAP Web IDE wizard for creating new projects, in the *Template Customization* step, you can now choose an additional OData sub-navigation target. You can also decide whether your app is to include smart variant management and / or flexible column layout.

For more information, see [Create List Report and Object Page Applications \[page 244\]](#).

Enhancement

Model Assignment Step in OData Service Wizard

When you add a new OData service to your app, you can assign a service to a default model or create a new model, letting you convert an application with mock data to a live application.

15 March 2018 - SAP Web IDE

New

npm v5 for SAP Fiori and SAPUI5 Projects

SAP Web IDE Full-Stack now leverages npm v5 for building your SAP Fiori and SAPUI5 projects with Grunt technology.

The npm 5 major release delivers several improvements over previous versions of npm that make the build of your SAP Fiori or SAPUI5 project run faster.

Note

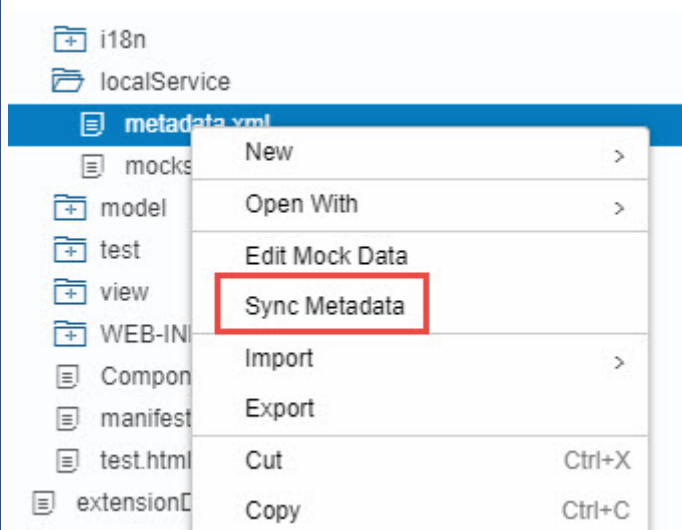
There is a change in Grunt build behavior: When projects are built, a new file named `package-lock.json` is added automatically to existing projects, providing a consistent and totally reproducible build of your project.

For more information, see [How npm v5 Affects Your Projects in SAP Web IDE Full-Stack](#) and [Run a Grunt Build \[page 409\]](#).

New

Sync Metadata

In SAP Fiori application projects, you can now sync the metadata of a service from the context menu of the `metadata.xml` file in your workspace.



For more information, see [Edit Mock Data \[page 403\]](#).

New

AutoFix Available in Development Perspective

In Java and JavaScript files, you can now make corrections from the *Problems* view.

For more information, see [Using the Problems View \[page 242\]](#).

Enhancement

Git Blame

You can now use the Git Blame feature to view the details of the last revision for each line in a code file.

For more information, see [Git Blame \[page 388\]](#).

Enhancement

Annotation Modeler

All targets annotated in the currently open annotation file are now displayed in the annotation structure by default. Previously, only the first target from the annotation file was displayed by default, and the rest of the targets had to be specified manually, using the Use the Git Blame feature to view the details of the last revision for each line in a code file. [Select Targets](#) dialog.

For more information, see [Select Targets \[page 312\]](#).

1 March 2018 - SAP Web IDE

New

Static Code Checks in Build

SAP Web IDE now performs static code checks by default when you generate a build for SAP Web IDE feature projects based on the recommended configuration.

For more information, see [How to Configure Static Code Checks for Feature Projects](#) and [Recommended Severity Levels for SAP Web IDE Features](#).

Enhancement

Use the Git Blame feature to view the details of the last Enhancement Integration with SAP Translation Hub

Destinations and trust management settings are now configured automatically.

For more information, see [Use SAP Translation Hub \[page 113\]](#).

15 February 2018 - SAP Web IDE

New

ESLint Plugin in SDK

You can now view static code check results for *SAP Web IDE Feature* projects in the *Problem* view based on recommended ESLint configurations.

For more information, see [Documentation](#) >> [Static Code Checks](#) in the [SAP Web IDE SDK](#).

Enhancement

Stash Changes Operation in the Git Pane

You can now stash (store away) your uncommitted changes, revert them from your working directory, and resume working on them at a later point.

For more information, see [Stash Changes \[page 379\]](#).

1 February 2018 - SAP Web IDE

New

Support Assistant Setting

You can now create a run configuration to include the Support Assistant tool, which checks whether your application is built according to the best practices for building SAPUI5 apps.

For more information, see [General Tab \[page 395\]](#) and [Support Assistant](#).

New

Template

The *SAP Fiori Worklist Application OData V4* template is now available; it works with OData V4.

For more information, see [Worklist Template](#).

New

HTML5 Modules for Multi-Target Projects

The following HTML modules are now available for multi-target projects:

- [SAP Fiori Master-Detail Application](#)
- [SAP Fiori Worklist Application](#)

i Note

These new modules work with SAPUI5 1.50 and later.

For more information, see [Worklist Template](#) and [Master-Detail Template](#).

4 January 2018 - SAP Web IDE

New

Debug Java Modules

It is now possible to debug Java modules using the Debugger pane, in which you can:

- View the call stack.
- Examine the variables.
- Step in and out of the functions.

For more information, see [Debug Java Modules \[page 485\]](#).

New

Add or Delete View in Storyboard

You can now add a new view to a freestyle project in the *Storyboard* perspective.

For more information, see [Add View to Freestyle Project \[page 330\]](#) or [Delete View \[page 333\]](#).

New

Create a Navigation Between Views

Using the layout editor, you can now create a navigation between views that is visible in the *Storyboard* perspective.

For more information, see [Create a Navigation Between Views \[page 265\]](#).

New

Export User Workspace

Administrators can now export the workspaces of a given user.

For more information, see [Export Workspaces \[page 538\]](#).

Archived Release Notes

- [2017 SAP Web IDE \(Archive\) \[page 44\]](#)

3.2 2017 SAP Web IDE (Archive)

The following document provides information about what was released in 2017.

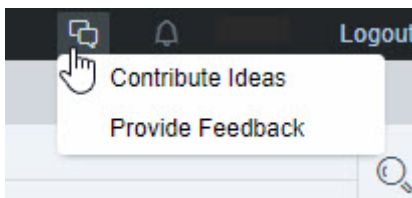
For more information about the latest release notes, see [What's New for SAP Web IDE \[page 12\]](#).

7 December 2017 - SAP Web IDE

New

Contribute Ideas

You can influence the development of SAP Web IDE by contributing your ideas which can be rated and voted on by other SAP Web IDE users. You can also track the ratings of your suggested feature on the Customer Influence website.



For more information, see [Customer Influence](#).

Enhancement

Provide Feedback

Now you can provide feedback for specific existing features, panes, and various development processes. You can either rate SAP Web IDE on these specific features or provide text feedback.

Enhancement

Tagging Commits in the Git History pane

You can mark important commits, such as release points, with custom tags.

For more information, see [Git Commands from the Git History Pane \[page 386\]](#).

Enhancement

ESLint v4.0.0 Support

The ESLint v4.0.0 linting utility is now supported for JavaScript validation in SAP Web IDE.

For more information, see <https://eslint.org/> .

23 November 2017 - SAP Web IDE

New

Beautify Option in Code Editor Settings

You can now set the code editor to automatically beautify the code of an active document when saving manually.

For more information, see [Configure the Code Editor \[page 133\]](#).

New

New Functionality in Java Editor

The following new functionality is now available in the Java editor:

- **Code Completion**
Code completion is now available in Java files.
- **Errors and Warnings in Problems View**
You can now see all error and warning messages in *Problems* view in Java files.
- **Beautify Code**
You can now beautify Java code from the context menu or the *Edit* menu.

Note

The first time you open a Java file, it may take longer for the code completion and problems to be displayed.

New

A new series of blogs describing how to develop full-stack applications in SAP Web IDE is available on the SAP Community .

[Introduction and On-boarding](#) 

[Creating a Database Module](#) 

[Creating a Java Module](#) 

[Creating a UI Module](#) 

New

There is a new blog describing how to deal with conflicts when merging Git changes.

See [Git Merge and Conflict Resolution](#) 

Enhancement

Annotation Modeler

You can now also use annotation modeler with mock data.

For more information, see [Configure Annotation Modeler to Use Mock Data \[page 109\]](#).

Enhancement

Cherry-Pick Operation in the Git History Pane

You can now cherry-pick changes from a selected commit to the current branch.

For more information, see [Git Commands from the Git History Pane \[page 386\]](#).

9 November 2017 - SAP Web IDE

New

SAP Enterprise App Modeler

SAP Enterprise App Modeler (also known as Mobile development kit for SAP Mobile Services) is a metadata-based application development platform which allows you to create, customize, deploy, and manage your enterprise apps in the cloud, without having to write code.

Use SAP Enterprise App Modeler Editor, to create new apps or customize existing apps. This feature provides app templates, additional wizards, drag and drop UI elements, and codeless building blocks that you can use to create your enterprise mobile apps without writing code.

For more information, see [Setting Up the Editor in SAP Web IDE](#).

New

Git History Pane

From the Git History pane, you can explore the history of committed changes that were made for repositories, folders, and files in a specific project. You can compare different versions of a selected file.

For more information, see [Git History \[page 385\]](#).

Enhancement

Git Pane

The Git pane visual design has been Enhancement and simplified, in particular the file staging functionality.

For more information, see [Stage Files \[page 376\]](#).

26 October 2017 - SAP Web IDE

Enhancement

Annotation Modeler

You can now also annotate function imports and function import parameters.

For more information, see [Architecture \[page 301\]](#).

14 September 2017 - SAP Web IDE

New

Additional Features

- [SAP S/4HANA Service Extension Tools](#) enables you to develop OData V4 services that extend existing SAP S/4HANA services by exposing additional data from different sources. The tools include the dedicated project and Java module templates.

For more information, see [Developing SAP S/4HANA Service Extensions \[page 503\]](#).

Enhancement

You can now use the following Java design-time features in the SAP Web IDE code editor:

- Code Assist
You can get code proposals for Java files inside a Java module.
- Code Validation
You can get code validation for Java files inside a Java module.

For more information, see [Developing Java Modules \[page 478\]](#).

31 August 2017 - SAP Web IDE

New

You can develop and debug Java modules for your multi-target applications (MTA) in Eclipse, while performing all the other development tasks in SAP Web IDE.

For more information, see [Use Eclipse to Develop Java Modules \[page 488\]](#).

New

You can now use the following tools for SAP HANA Database development:

- You can model calculation views (.hdbcalculationview).
- You can use the CDS graphical editor to develop CDS artifacts (.hdbcds).
-

For more information, see [Developing Database Artifacts \[page 459\]](#).

Enhancement

You can now display a tooltip with the descriptions for values of expression type You can use the analytic privilege editor to create analytic privileges (.hdbanalyticprivilege).EnumMember.

For more information, see [Edit Annotations \[page 310\]](#).

17 August 2017 - SAP Web IDE

New

You can now deploy multi-target applications (MTA) to production systems directly from SAP Web IDE.

For more information, see [Packaging and Deploying Applications to Production Systems \[page 506\]](#).

3 August 2017 - SAP Web IDE

New

The workflow editor feature has a new *You can use the analytic privilege editor to create analyticBuild and Extend Workflows using SAP Workflow service* video.

See the video in the [SAP Web IDE YouTube channel](#) .

Enhancement

You can now search through the annotation file in order to quickly locate the annotations you want to edit.

For more information, see [Search in Annotations \[page 318\]](#).

Enhancement

The ESLint tool of the basic JavaScript validator has now been updated to version 4.0.0.

Enhancement

In the development perspective, you can now arrange open files by dragging their tabs.

20 July 2017 - SAP Web IDE

Enhancement

The annotation modeler has a new *Getting started with the SAP Web IDE annotation modeler* video.

See the video in the [SAP Web IDE YouTube channel](#) .

6 July 2017 - SAP Web IDE

New

Compare in Git Status Table

You can now compare a file listed in the status table in the *Git Pane* with an earlier version of the file. If you staged the file, you can compare it with the last committed version. If you did not stage the file, you can compare the file to the staged version or the last committed version (if no staged version exists).

For more information, see [Compare and Merge Code \[page 377\]](#).

New

Workflow Editor Feature

You can now create and deploy workflows on SAP BTP using the Workflow editor feature in SAP Web IDE.

For more information, see [SAP Workflow Service](#).

22 June 2017 - SAP Web IDE

New

SAP HANA Database Explorer

The SAP HANA database explorer is now available for use with SAP Web IDE Full-Stack. The database explorer allows you to execute SQL statements and database procedures, query information about the database, and view information about database catalog objects.

For more information about the database explorer for the Cloud Foundry environment, see [About the SAP HANA Database Explorer and the SQL Analyzer](#).

Enhancement

Multitarget Application Development

The development of full-stack multitarget applications is now Enhancement by an ability to create the database (HDB) and business logic (Java) modules in addition to the previously available UI (HTML5) modules.

For more information, see [Developing SAP HANA Database \(HDB\) Modules \[page 457\]](#) and [Developing Java Modules \[page 478\]](#).

Enhancement

Using the Translation Hub

You can now translate your project's `i18n.properties` file using the Translation Hub service. This capability, which has previously been released as experimental, is now fully supported.

For more information, see [Use SAP Translation Hub \[page 113\]](#).

14 June 2017 - SAP Web IDE

New

SAP Enterprise App Modeler Overview Feature

The SAP Enterprise App Modeler is a metadata-based application development platform. It uses SAP Web IDE and SAP BTP to enable you to customize, deploy, and manage SAP Asset Manager in the cloud.

For more information, see [SAP Enterprise App Modeler](#).

25 May 2017 - SAP Web IDE

New

You can now create a destination for a Cloud Foundry service from within the wizard for creating an HTML5 module within a multi-target application. The wizard automatically fills in many of the destination fields, and saves you the need to go to the SAP BTP cockpit to create the destination.

The screenshot shows a wizard interface with a progress bar at the top containing five steps: 'Template Selection', 'Basic Information', 'Data Connection' (highlighted in dark blue), 'Template Customization', and 'Confirmation'. Below the progress bar, the title is 'New CRUD Master-Detail Module' and the sub-title is 'Data Connection'. A message reads: 'Choose a service from one of the sources listed below.' Below this, a red-bordered box contains the text: 'If no destination exists for your Cloud Foundry service, configure a new destination'. Underneath, there are two columns. The left column is titled 'Sources' and contains two items: 'Service Catalog' (highlighted in dark blue) and 'Workspace'. The right column is titled 'Choose a system, explore, and select a service. Optionally, you can drill d...' and contains a 'Select a system' input field and a 'Services' section with a 'Search Service' button.

For more information, see [.Create Destinations for Cloud Foundry Services \[page 497\]](#)

17 May 2017 - SAP Web IDE

Announcement

SAP is offering this new release of SAP Web IDE, based on the Eclipse Che foundation, side by side with the standard SAP Web IDE, which uses the older Eclipse Orion server-side foundation and is no longer being developed. When you navigate to the SAP Cloud Portal cockpit services, you will see two different tiles, SAP Web IDE and the new SAP Web IDE, multi-cloud version.

If you want to develop applications for SAP BTP, Cloud Foundry environments, or are interested in running Grunt builds, you should use the new release of SAP Web IDE, multi-cloud version.

See [SAP Web IDE Multi-Cloud Version](#)

3.3 Introducing SAP Web IDE Full-Stack

This release of SAP Web IDE Full-Stack introduces development support for SAP's multi-cloud solution, which includes support for developing applications that are deployed to SAP BTP environments based on Cloud Foundry technology.

To support SAP's full-stack development solution, this release includes changes to SAP Web IDE. In particular, SAP Web IDE now uses a powerful open-source server-side foundation called Eclipse Che, which will serve to introduce additional innovative development services as they are become available. One new feature is already

available in this release: the ability to create and run Grunt builds of your development project before final deployment.

If you want to develop applications for SAP BTP, Cloud Foundry environments, or are interested in running UI5 builds or Grunt builds, you should use the new release of SAP Web IDE Full-Stack.

For detailed onboarding instructions, see [Getting Started \[page 53\]](#).

4 Getting Started





Here's a checklist for setting up your system so you can develop applications using SAP Web IDE Full-Stack.

i Note

If you use the personal edition together with the full-stack development version, you can migrate the projects between SAP Web IDE versions as described in [Import Projects from the Previous Version of SAP Web IDE \[page 105\]](#). However, all changes to the project settings must be copied manually.

i Note

SAP Web IDE is a [service running in the Neo environment](#). To enable and configure the SAP Web IDE service, you must navigate to the corresponding subaccount as described in the [Navigate to Global Accounts and Subaccounts in the Cockpit](#) topic.

Step	Description	Links/Information
Sign up for an SAP BTP global account.	You require an SAP BTP global account to enable SAP Web IDE.	Getting a Global Account
Enable principal propagation (production only)	In order to support the SSO solution of SAP Web IDE, you need to configure your account to allow principal propagation.	Principal Propagation [page 548]
Enable SAP Web IDE Full-Stack	In the cockpit, go to  Neo Environment  Services  SAP Web IDE Full-Stack  and enable the service.	
Grant user permissions	To enable working with SAP Web IDE, developers need to be assigned the <code>DiDeveloper</code> role.	Assign Users Permission for SAP Web IDE [page 532]
Connect to backend systems (optional)	If you need to use ABAP backend services in your applications, create destinations to the services.	Connecting to ABAP Systems
Migrate projects	If you have existing projects in SAP Web IDE, you can migrate them to SAP Web IDE Full-Stack.	Import Projects from the Previous Version of SAP Web IDE [page 105]
Enable identity provider (IdP)-based authentication for SAP Web IDE applications (optional)	If you define a custom identity provider for your subaccount in SAP BTP, be sure to configure the assertion-based attributes mapping for this IdP.	Configure Trust to the SAML Identity Provider

In order to develop applications to run in SAP BTP, Cloud Foundry environment, you also need to set up the following:

i Note

For scenarios involving SAP HANA, make sure you have 1GB of Application Runtime in your development space for DevOps services.

Step	Description	Links/Information
<p>Create subaccounts in the SAP BTP, Cloud Foundry environment.</p> <p>Select a region according to the guidelines in the Neo and Cloud Foundry Regions [page 61] topic.</p>	<p>When you create a subaccount in the SAP BTP, Cloud Foundry environment, a Cloud Foundry organization is automatically created for that subaccount.</p> <p>We recommend you create subaccounts for development, staging/test and production purposes.</p>	<p>Create Subaccounts In Your Global Account</p>
<p>Create spaces</p>	<p>You can create and delete spaces in a Cloud Foundry organization using the SAP BTP cockpit or the console client (Cloud Foundry command line interface).</p> <p>We recommend at least one space for a development team working on the same project (that is, one space per project).</p> <p>For staging/test and production organizations, one space is sufficient.</p>	<p>Managing Spaces</p>
<p>Assign members to your Cloud Foundry organizations and spaces</p>	<p>Enable your developers to work with your SAP BTP, Cloud Foundry environments.</p> <p>Your developers should be assigned to the space developer role to be able to use the space from SAP Web IDE.</p>	<p>Managing Members</p>
<p>Connect to Cloud Foundry services (optional)</p>	<p>You can create destinations to Cloud Foundry services and consume them in the applications you create with SAP Web IDE.</p> <p>For each service, you need to create a destination for each space to which the service is deployed.</p> <p>The step is optional at this point. You can set up your destinations now as you get started, or you can create them when you create new applications.</p>	<p>Connect to Cloud Foundry Services [page 56]</p> <p>Create Destinations for Cloud Foundry Services [page 497]</p>

4.1 Open SAP Web IDE

You open SAP Web IDE in a web browser.

Prerequisites

To work with SAP Web IDE, you must be assigned the `DiDeveloper` role. For more information, see [Assign Users Permission for SAP Web IDE \[page 532\]](#).

Procedure

1. Log on with a user (who is an account member) to the SAP BTP cockpit, using the developer account at one of the URLs list in [Regions and Hosts](#).
2. Select the *Neo* environment.
3. In the navigation pane, choose *Services*.
4. On the *Services* pane, select *SAP Web IDE Full-Stack*.
5. Click *Open SAP Web IDE Full-Stack*, and then *Go to Service*.

The following browsers are supported:

i Note

Unless specifically stated, only the latest browser version is supported.

- Microsoft Edge
- Mozilla Firefox
- Google Chrome
- Safari (on iOS platforms only)

i Note

Opening multiple instances of SAP Web IDE in parallel may cause issues. For more information, see [Known Issues \[page 563\]](#).

4.2 Connect to Cloud Foundry Services

You can create destinations to HTTP-based Cloud Foundry services and consume them in the applications you create with SAP Web IDE for Full-Stack Development.

Prerequisites

- You must be an administrator in your SAP BTP account.
- You are using the SAP ID service for CF login. SAP IDP is not supported.

Procedure

1. Create a destination (on the SAP BTP account running your SAP Web IDE) to your Cloud Foundry service, as described in [SAML Bearer Assertion Authentication](#).

Set the following parameters:

- **Type:** `HTTP`
- **URL:** `https://[myorg-myspace]-[service].cfapps.sap.hana.ondemand.com/`
- **Proxy Type:** `internet`
- **Authentication:** `OAuth2SAMLBearerAssertion`
- **Client Key, Client Secret, Audience, and Token Service URL:** Go to the Cloud Foundry command line and enter `cf env <application-name>`, which will provide service credentials.
 - **Client Key:** Use the value in `clientid`
 - **Client Secret:** Use the value in `clientsecret`
 - **Audience and Token Service URL:** Use the value in `url` followed by the path `/oath/token`.

Make sure to add the following additional parameters to the destination:

- **TrustAll:** `true`
 - **WebIDEEEnabled:** `true`
 - **WebIDESystem:** `API`
 - **WebIDEUsage:** `odata_gen`
2. Establish trust from your SAP BTP account (where you are running SAP Web IDE) to your SAP BTP, Cloud Foundry organization, as described in [Principal Propagation from the Neo to the Cloud Foundry Environment](#).

Troubleshooting

When trying to connect to your Cloud Foundry system, here are some errors you may get and some probable causes:

- **500:**
 - The identity provider (IDP) is not in an active state.
 - No trust was configured
 - The default user account and authentication (UAA) needs to be used
- **403:**
 - Roles collection is missing
 - Application roles are missing
 - Roles collection was assigned incorrectly

4.3 Connect to ABAP Systems

For applications that do not need to run on Cloud Foundry, establish a connection to an ABAP back-end system by creating one destination for multi-usage.

Prerequisites

- You must have an Administrator or Developer role to modify destinations.
 - Make sure you are using OData Version 2.0 only.
 - If your remote system resides behind a firewall (proxy type `OnPremise`), the following prerequisites must be met:
 - You have set up the Cloud Connector and defined a virtual host mapping for the system.
See [Cloud Connector](#).
 - In the [Access Control](#) tab page of the Cloud Connector, you have granted access to the URL paths (Resources) for the required usages (for [Access Policy](#), select option [Path and all sub-paths](#)):
 - `/sap/opu/odata`
for the OData functionality of Gateway
 - `/sap/bc/ui5_ui5`
for executing SAPUI5 applications from the SAPUI5 ABAP Repository
 - `/sap/bc/adt`
 - `/sap/bc/ui2/app_index/`
for extensibility/import scenarios and developing or deploying to SAPUI5 ABAP Repository
 - `/sap/bc/bsp`
 - `/sap/bc/ui2`
for working with fact sheets
 - `/sap/hba`
for SAP HANA XS OData services
- See [Configuring Access Control \(HTTP\)](#).

Context

For every system that you want to connect to, you create one multi-usage destination. You enter the required usages as a value for the property *WebIDEUsage*.

Procedure

1. Log on with a user (who is an account member) to the SAP BTP cockpit in your region. See [Regions and Hosts](#) for a list of all available regions.

i Note

You can access the SAP BTP cockpit from SAP Web IDE by selecting **Tools > SAP BTP Cockpit**.

2. Open the *Destinations* editor.
3. Choose *New Destination*.
4. Configure the destination as follows:

Property	Value
Name	Provide a name for the destination that includes the desired service.
Type	HTTP
Description	(Optional) Provide a description for the destination
URL	<code><protocol>://<host>:<port></code> <code><protocol>://<virtual host>:<virtual port></code> (if you are using Cloud Connector)
Proxy Type	Select Internet or OnPremise , depending on the connection you need to provide for your application. <ul style="list-style-type: none">○ If your remote system is publicly accessible on the internet, select Internet.○ If your remote system resides behind a firewall, select OnPremise.
Authentication	Select <i>No Authentication</i> or <i>Basic Authentication</i> depending on the authentication you need for the connection. For more information, see Creating HTTP Destinations .

i Note

- Do not add the URL path for the usage (for example, `/sap/opu/odata`).
- When connecting an ABAP system that has several application servers, make sure the host is not configured as the message server used for load balancing of HTTP requests. This will cause issues during deployment.

5. In the *Additional Properties* section, choose *New Property* for each of the properties described below.

Key	Value
WebIDEUsage	Enter one or more of the following values: <ul style="list-style-type: none">○ <code>odata_abap</code>: for the OData functionality of Gateway (corresponds to URL path <code>/sap/opu/odata</code>)

Key	Value
	<ul style="list-style-type: none"> ◦ <code>odata_gen</code>: for generic OData functionality (service URL must be provided manually in the New Project wizard)
	<div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p>⚠ Caution</p> <p>Enter either the <code>odata_abap</code> value or the <code>odata_gen</code> value. Do not use both values at the same time.</p> </div>
	<ul style="list-style-type: none"> ◦ <code>ui5_execute_abap</code>: for executing SAPUI5 applications from the SAPUI5 ABAP Repository (corresponds to URL path <code>/sap/bc/ui5_ui5</code>) ◦ <code>dev_abap</code>: for extensibility scenarios and developing or deploying to SAPUI5 ABAP Repository (corresponds to URL path <code>/sap/bc/adt</code>) ◦ <code>bsp_execute_abap</code>: for working with fact sheets (corresponds to URL path <code>/sap/bc/bsp</code>) ◦ <code>plugin_repository</code>: for exposing optional plugin repositories (corresponds to URL path <code>/plugins/pluginrepository</code>) ◦ <code>odata_xs</code> for SAP HANA XS OData services (corresponds to URL path <code>/sap/hba</code>) ◦ <code>api_mgmt_catalog</code> for the API management system ◦ <code>api_mgmt_proxy</code> for the API's endpoint ◦ <code>odata_smp</code> for accessing OData Services in an SAP Mobile Platform system ◦ <code>odata_hcp_odp</code> for accessing OData services in an OData Provisioning system on SAP BTP ◦ <code>odata_hci</code> for accessing OData services in a HANA Cloud Integration system on SAP BTP. To complete the setup, you also need to configure an additional destination, representing an integration flow node, with <code>hci_ifl_node</code> as the value for <code>WebIDEUsage</code> property. ◦ <code>hci_ifl_node</code> for setting up the integration flow node in the HANA Cloud Integration system on SAP BTP ◦ <code>smart_business_odata</code> for smart business OData services ◦ <code>smart_business_gen</code> for smart business annotation generation services
	<div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p>i Note</p> <ul style="list-style-type: none"> ◦ In order to import an application from your ABAP system and execute it, the following <code>WebIDEUsage</code> property values are needed: <code>dev_abap</code>, <code>odata_abap</code>, and <code>ui5_execute_abap</code>. ◦ When you enter multiple usages for a destination, separate them by commas without spaces (for example, <code>odata_abap,ui5_execute_abap</code>). </div>

Key	Value
WebIDEEEnabled	true
WebIDESystem	<SAP system_ID>
sap-client	<SAP client from the ABAP system>
WebIDEAdditionalData	<p>This property is optional.</p> <p>You can add one or more of the following values:</p> <ul style="list-style-type: none"> ◦ api_mgmt for specifying an API key to retrieve metadata ◦ full_url for specifying the full URL of a service so that metadata is shown without having to choose the service

i Note

When specifying this value, the *URL* field of your destination should contain the full URL of your service. Otherwise, there will be no communication between SAP Web IDE and your system.

- Repeat steps 4 and 5 for any additional destinations.
- Choose *Save*.

Related Information

[Cloud Connector](#)

[Creating HTTP Destinations](#)

[Securing the SAPUI5 ABAP Repository](#)

[Configuring the ABAP Back-end for ABAP Development Tools](#)

[Connect Remote Systems in SAP Web IDE Personal Edition](#)

4.3.1 Requirements for Connecting to ABAP Systems

The following is prerequisite information for connecting to ABAP systems.

- You have installed `SAP_BASIS` 7.31 SP 14 or 7.40 SP 8 or later.

i Note

If you are using earlier releases of `SAP_BASIS` than those mentioned above, you must apply the following SAP Notes:

- [2046730](#)
- [2047506](#)
- [1684342](#)

- If you are using `SAP_BASIS 7.31`, make sure that the software components of the UI add-on for SAP NetWeaver (`UI_700` and `UI5_731`) are installed.

Related Information

[Connect to ABAP Systems \[page 57\]](#)

[Connect Remote Systems in SAP Web IDE Personal Edition \[page 523\]](#)

4.4 Neo and Cloud Foundry Regions

You can find the Cloud Foundry region where we recommend you create subaccounts based on the Neo region of your SAP Web IDE.

Region Mapping for Enterprise Accounts

Neo Region (Operated by SAP)	Host	Recommended Cloud Foundry Region
Australia (Sydney 1)	ap1.hana.ondemand.com	Europe (Frankfurt) https://api.cf.eu10.hana.ondemand.com
Australia (Sydney 2)	ap2.hana.ondemand.com	Europe (Frankfurt) https://api.cf.eu10.hana.ondemand.com
Brazil (São Paulo)	br1.hana.ondemand.com	US East (VA) https://api.cf.us10.hana.ondemand.com
Brazil (São Paulo)	br2.hana.ondemand.com	US East (VA) https://api.cf.us10.hana.ondemand.com
Canada (Toronto)	ca1.hana.ondemand.com	US East (VA) https://api.cf.us10.hana.ondemand.com
China (Shanghai)	cn1.platform.sapcloud.cn	Europe (Frankfurt) https://api.cf.eu10.hana.ondemand.com

Neo Region

(Operated by SAP)	Host	Recommended Cloud Foundry Region
Europe (Rot/Germany)	hana.ondemand.com eu1.hana.ondemand.com	Europe (Frankfurt) https://api.cf.eu10.hana.ondemand.com
Europe (Amsterdam/Netherlands)	eu2.hana.ondemand.com	Europe (Frankfurt) https://api.cf.eu10.hana.ondemand.com
Russia (Moscow)	ru1.hana.ondemand.com	Europe (Frankfurt) https://api.cf.eu10.hana.ondemand.com
Japan (Tokyo)	jp1.hana.ondemand.com	Europe (Frankfurt) https://api.cf.eu10.hana.ondemand.com
US East (Ashburn/VA)	us1.hana.ondemand.com	US East (VA) https://api.cf.us10.hana.ondemand.com
US West (Chandler)	us2.hana.ondemand.com	US East (VA) https://api.cf.us10.hana.ondemand.com
US East (Sterling/VA)	us3.hana.ondemand.com	US East (VA) https://api.cf.us10.hana.ondemand.com

Region Mapping for Trial Accounts

Neo Region

(Operated by SAP)	Host	Recommended Cloud Foundry Region
Europe (Rot/Germany) - Trial	hanatrial.ondemand.com	Europe (Frankfurt) https://api.cf.eu10.hana.ondemand.com

4.5 Migrate your Projects

Learn how to migrate your existing projects from SAP Web IDE to SAP Web IDE Full-Stack as well as the post-migration steps.

i Note

If you are using SAP Web IDE with SAP S/4 HANA Cloud, please wait until you receive specific instructions before migrating to SAP Web IDE Full-Stack.

The following aspects are relevant to your migration to SAP Web IDE Full-Stack:

- [Differences Between SAP Web IDE Versions \[page 63\]](#)
- [Enable SAP Web IDE Full-Stack \[page 64\]](#)
- [Migration Procedure \[page 68\]](#)

4.5.1 Differences Between SAP Web IDE Versions

Pay attention to the following features that are handled differently in SAP Web IDE Full-Stack.

- **Project Metadata:** Each project in SAP Web IDE Full-Stack contains a special system/metadata folder called `.che` that is hidden by default.
When you migrate a project from SAP Web IDE to SAP Web IDE Full-Stack, a new metadata folder named `.che` is generated.
The project settings that were in your `.project.json` file are converted into a new format and saved in the generated folder.

→ Remember

Do not change the contents of the `.che` folder.

i Note

The old `.project.json` metadata file appears in the project sources but is no longer in use.

- **Application Build:** If you used the SAPUI5 Client build in SAP Web IDE to minify and bundle files before deployment, in the Full-Stack Development version of SAP Web IDE, you should use the UI5 build or the Grunt build. For more information, see [Application Build](#) and [Building Applications](#).
- **Optional Plugins:** Optional plugins are available in SAP Web IDE Full-Stack as extensions.
If you enabled optional plugins in SAP Web IDE, you should enable the corresponding extensions manually in SAP Web IDE Full-Stack as well. For more information, see [Enable SAP Web IDE Extensions \[page 510\]](#).
If you developed a plugin in SAP Web IDE, you will have to convert it to an extension as described in the [Creating Our First Extension \(MTA\)](#) tutorial in the SAP Web IDE SDK.
- **SAP Enterprise Portal Plugin:** [SAP Enterprise Portal Plugin](#), for deploying SAPUI5 applications to the SAP Enterprise Portal, has been deprecated in the SAP Web IDE Full-Stack. To get the same functionality, read the blog post [How to Import SAP UI5 Applications to SAP NetWeaver Portal](#).
- Use of SAP Web IDE, personal edition. The personal edition can be used for SAP Web IDE Full-Stack but with the following restrictions:

- The UI5 build and the Grunt build are not supported.
- Changes to the Project Settings are not saved when switching between versions.
- Performance measurements: For more information, see [Running an Application with Performance Measurements](#).
- Default Run Configurations are not generated when creating a project from a template. To learn how to create the configurations manually, see [Configuring How to Run Projects](#).

i Note

We recommend all developers working on the same project use the same SAP Web IDE version. If you choose different cloud versions or use the personal edition together with the full-stack development version, all changes to the project settings must be copied manually.

4.5.2 Enable SAP Web IDE Full-Stack

Set up your SAP BTP account so developers in your organization can start developing applications with SAP Web IDE Full-Stack.

Prerequisites

You must be an administrator on the SAP BTP account.

Procedure

1. Log on to your SAP BTP account.

i Note

Make sure you are in the Neo environment, and not the Cloud Foundry environment. Neo is the classic, proprietary SAP BTP, and you can return to it by navigating to ► [Home](#) ► [Go to Neo](#) ►.

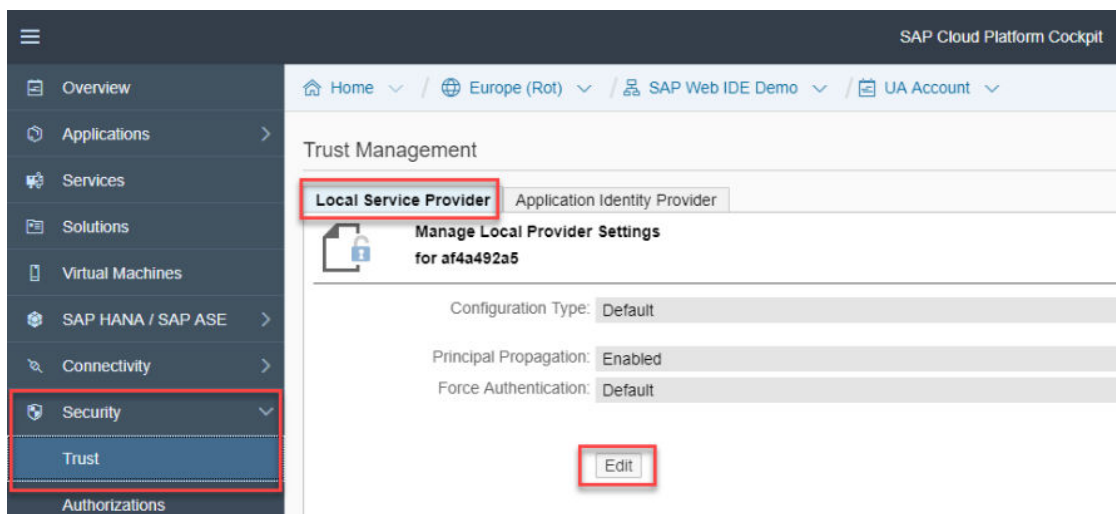
The screenshot shows the SAP Cloud Platform Cockpit interface. The left sidebar contains navigation options: Overview, Applications, Services, Solutions, SAP HANA / SAP ASE, Connectivity, Security, and Repositories. The main content area displays the 'System Status' for a subaccount. It features two panels: 'JAVA' and 'HTML5'. The 'JAVA' panel shows 'Overall Health' as 'No applications yet' and provides links to 'Get started here!', 'Create an application', and 'Run sample applications'. The 'HTML5' panel shows 'Overall Health' as 'OK' with a green checkmark, and lists '2 Applications' (1 Started, 1 Stopped).

2. Enable *Principal Propagation*.

i Note

If you are working in a Trial account, you can skip this step.

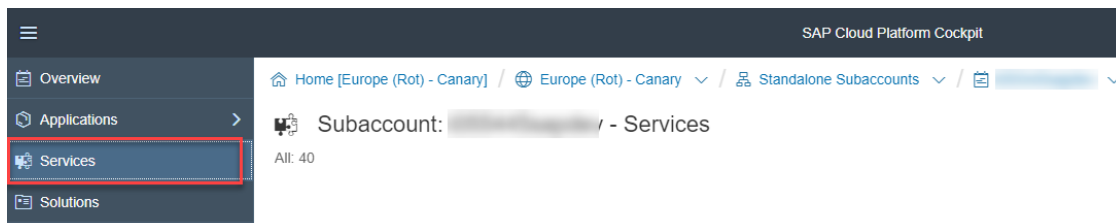
- a. Go to **Security > Trust**, and select the **Local Service Provider** tab.
- b. Choose **Edit**.



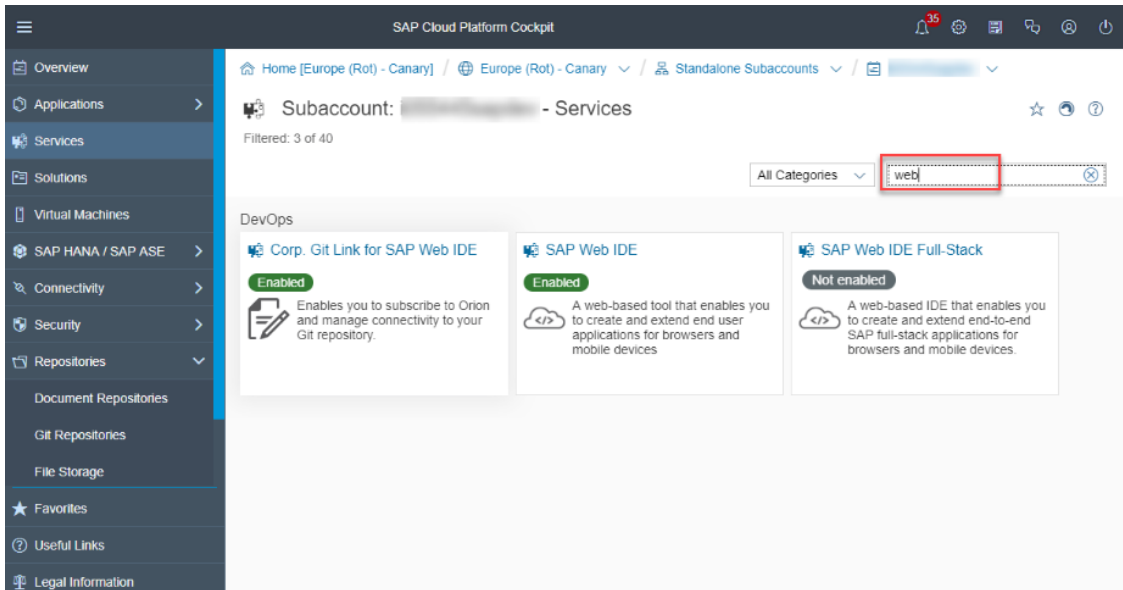
- c. Change the **Principal Propagation** field to **Enabled**, and choose **Save**.

3. Enable the SAP Web IDE Full-Stack service.

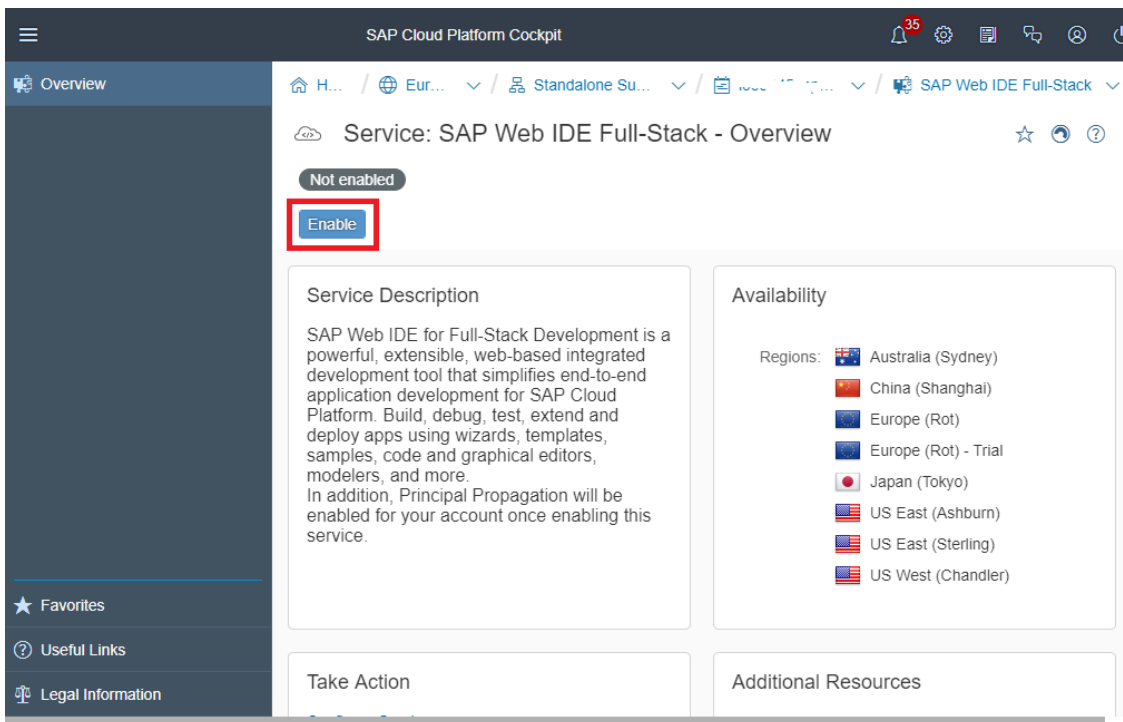
- a. Click *Services*.



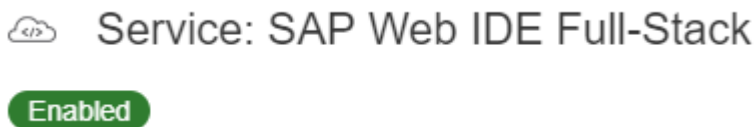
- b. In the search box, search for **Web**, and then click the **SAP Web IDE Full-Stack** tile.



c. Click **Enable**. This may take a few minutes.



d. Wait for the status to change to the green, **Enabled** status.



4. Give developers permission to SAP Web IDE Full-Stack as described in [Authorization Migration \[page 535\]](#) or assign permissions manually as described in the [Assign Users Permission for SAP Web IDE \[page 532\]](#) topic.

i Note

You may skip this step if the following conditions are met:

- The developers are members of an account with the `Developer` predefined platform role.
- The platform identity provider and the application identity provider in the account are both SAP ID services. This is the default configuration.

5. If you have integrated SAP Web IDE with SAP S/4HANA Cloud 1805 or earlier using the `SAP_COM_0013` communication scenario, you need to update the integration as described in [Updating the Integration of SAP S/4HANA and SAP Web IDE \[page 67\]](#).
6. If needed, add any additional configurations. See [Getting Started](#).

4.5.2.1 Updating the Integration of SAP S/4HANA and SAP Web IDE

If you have integrated SAP Web IDE with SAP S/4HANA Cloud 1805 or earlier using the communication scenario `SAP_COM_0013`, you need to update the integration.

Context

The communication system used in the existing communication arrangement for the communication scenario `SAP_COM_0013` (`SAP Web IDE Integration`) needs to be adjusted to refer to the host name of the SAP Web IDE Full-Stack edition.

Procedure

1. In your subaccount of SAP BTP, Neo environment, go to the service page [Service: SAP Web IDE Full-Stack - Overview](#) and copy the URL behind [Go to Service](#).
2. In the app Communication Arrangements, open the communication arrangement with the scenario ID `SAP_COM_0013`.
3. Display the communication system of the communication arrangement.
4. Edit the host name located under **Technical Data** **General**.

The new host name is the URL copied in *Step 1*.

Example

```
webidecp-<SAP CP NEO subaccount>.dispatcher.<region such as us1>.sap.hana.ondemand.com
```

5. Save the updated communication arrangement.

Results

The communication system of the communication arrangement now communicates with the host of the SAP Web IDE Full-Stack edition.

4.5.3 Migration Procedure

List of steps required to migrate your existing projects from SAP Web IDE to SAP Web IDE Full-Stack.

Procedure

1. If you enabled optional plugins in SAP Web IDE, you should enable the corresponding extensions manually in SAP Web IDE Full-Stack as well. For more information, see [Enable SAP Web IDE Extensions \[page 510\]](#).
2. Migrate your projects by cloning your projects with Git into SAP Web IDE Full-Stack. For more information on using Git, see [Using Source Control \(Git\)](#).
3. If your projects are connected to Git, commit and push the newly generated metadata `.che` folders.
4. If you used the [SAPUI5 Client Build](#) functionality before you deployed your application to SAP BTP or SAPUI5 ABAP Repository, the following files are created for you:
 - `package.json`
 - `Gruntfile.js`
 - `package-lock.json`
 - `.npmrc`

For information on these files, see [Building Applications](#)

If your projects are connected to Git, commit and push these files.

5. If a project contains an ESLint configuration file `.eslintrc`, the `customRulesPath` property will be removed and its value will be maintained in the `.che` metadata folder.

If the project is connected to Git, commit and push this file.


6. If you were using the SAP Web IDE Hybrid App Toolkit Add-on (HAT) plugin to perform a cloud build of a mobile project, you must perform a new build of the project in SAP Web IDE Full-Stack.

A new mobile app containing a different app ID will be created in the SAP Mobile Service for development and operations.

To avoid confusion, we recommend you delete the old mobile app from SAP Mobile Service for development and operations and remove the previously installed app from your device.


i Note

The user preferences you configured in SAP Web IDE are not automatically migrated to SAP Web IDE

Full-Stack. If you want to keep your user preferences, open the Preferences perspective  in the left sidebar, and make the desired changes.

7. See the results of previous migrations by clicking the Migration icon and selecting *Show Report*.


i Note

You can learn about migration tips and tricks in [this blog post](#) .





5 Setting User Preferences

You can set your preferences for working in SAP Web IDE.

Procedure

1. To open the *Preferences* perspective, in the left sidebar, choose  (Preferences).
2. Select the type of preference that you want to change.

For example, you can set the following SAP Web IDE preferences:

Preferences	More Information
Code check settings to enable and disable inline code validation. The project-level appearance settings are used to validate code and flag messages.	Configure Code Checking [page 156]
Code completion settings to enable inline code completion and comment completion.	Configure Code Completion [page 139]
Code editor settings to select the theme to be used by the code editor and the font size. You can also choose to have the changes in all open documents saved automatically at preset intervals.	Configure the Code Editor [page 133]
Editor settings to select which editor to use for which type of file, for example, the code editor or the layout editor for SAPUI5 XML views.	Go to  Tools  Preferences  Default Editors  .
Set up Git for source control.	Set Up Git [page 366]
Customize SAP Web IDE keyboard shortcuts.	Customize Keyboard Shortcuts [page 88]
Enable and disable SAP Web IDE features (extensions) in order to enrich and add tools to your SAP Web IDE.	Enable SAP Web IDE Extensions [page 510]

Preferences

Delete or ignore workspace persistence.

More Information

If you start SAP Web IDE after you have been logged out, the system restores the latest status of the workspace with its preference settings and all the editors that have been opened.

If you want to reload SAP Web IDE while ignoring this persistence feature, for example, because an editor cannot be loaded or is frozen, add the parameter

settings=ignore to the URL and refresh your

browser. The persistence information is not deleted (meaning that when you remove the parameter from the URL and refresh your browser, SAP Web IDE restores the latest status of the workspace and the editors).

To delete the persistence information, add the parameter **settings=delete** to the URL.

6 SAP Web IDE Basics

Familiarize yourself with the basic tasks for working in SAP Web IDE.

[Navigating SAP Web IDE \[page 72\]](#)

In SAP Web IDE, you can use the menu options and toolbar icons to perform various operations.

[Working in the Workspace \[page 74\]](#)

You can view the projects contained in your workspace from the *Files* or the *Project Explorer* tabs.

[Status Bar \[page 77\]](#)

The status bar enables you to see which tasks are running in the background and their status.

[Workspace Manager \[page 78\]](#)

Work in multiple workspaces to organize your projects better. You can easily switch between workspaces.

[Search Options \[page 79\]](#)

SAP Web IDE's search options include a simple search within an open file in the code editor, advanced search and replace across multiple files in a project, and a search to find references.

[Resizing Panes \[page 85\]](#)

You can show and hide various panes, as well as change their sizes, to fit your way of working.

[Keyboard Shortcuts \[page 85\]](#)

You can use keyboard shortcuts to perform actions in SAP Web IDE.

6.1 Navigating SAP Web IDE

In SAP Web IDE, you can use the menu options and toolbar icons to perform various operations.

SAP Web IDE contains two perspectives: development and database explorer. Each perspective has its own set and layout of menus, toolbars and panes, as described below.

Development Perspective

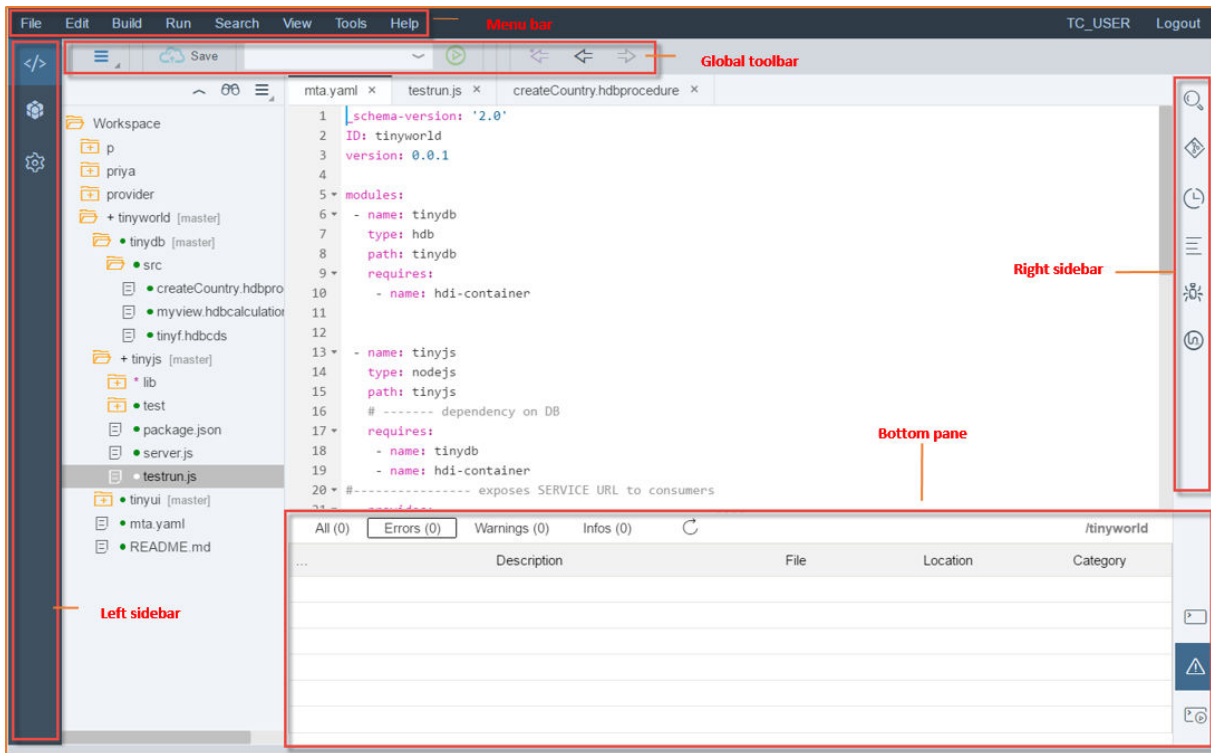
Menu bar - Provides access to all operations available in SAP Web IDE.

Global toolbar - Depending on the item that is activated in the workspace, you can choose from the icons in the global toolbar (icons of actions that are not applicable are grayed out).

Left sidebar - Use the buttons to switch between the development workspace, the database explorer, and user preferences.

Right sidebar - Use the buttons to switch between the different panes available in SAP Web IDE (for example, Git pane, Outline pane, and so on).

Bottom pane - Use the toggle buttons to display the console or the problems view.



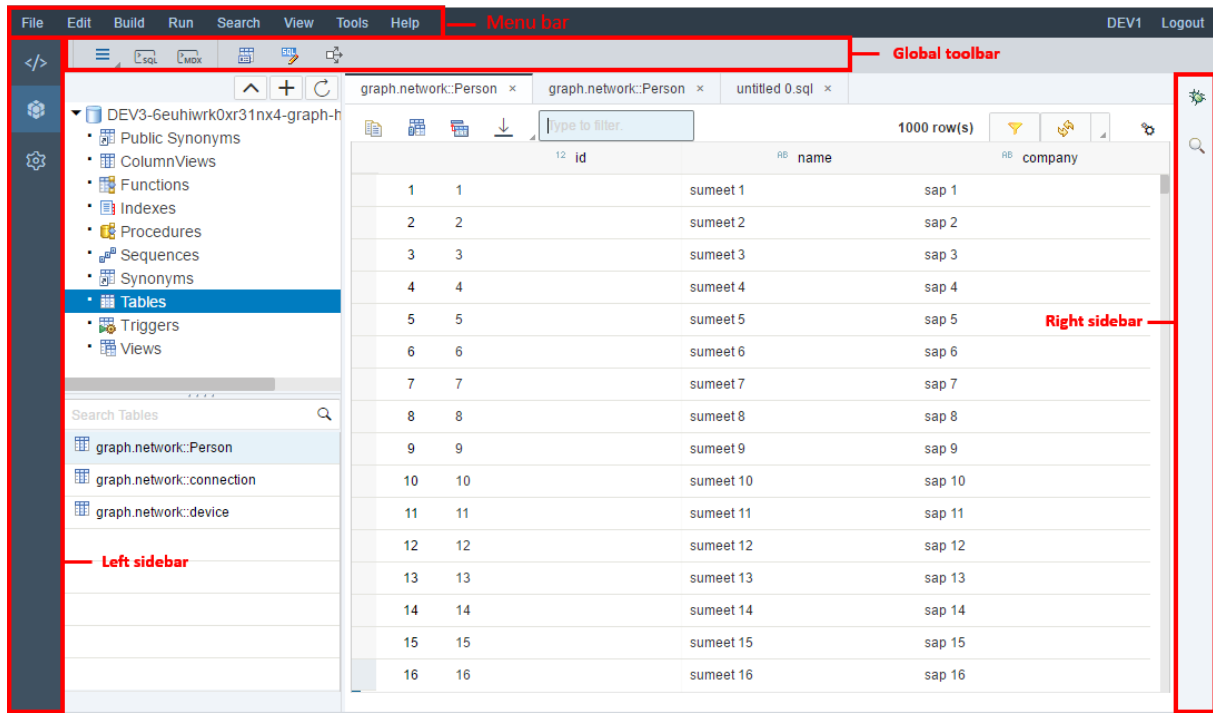
Database Explorer Perspective

Menu bar - Only the Help menu bar item is activated for the database explorer.

Global toolbar - Depending on the item that is activated in the workspace, you can choose from the icons in the global toolbar (icons of actions that are not applicable are grayed out).

Left sidebar - Use the buttons to switch between the development workspace, the database explorer, and user preferences.

Right sidebar - Use the buttons to switch between the different panes available in the database explorer.



Parent topic: [SAP Web IDE Basics \[page 72\]](#)

Related Information

[Working in the Workspace \[page 74\]](#)

[Status Bar \[page 77\]](#)

[Workspace Manager \[page 78\]](#)

[Search Options \[page 79\]](#)

[Resizing Panes \[page 85\]](#)

[Keyboard Shortcuts \[page 85\]](#)

[Working with the Database Explorer](#)

6.2 Working in the Workspace

You can view the projects contained in your workspace from the *Files* or the *Project Explorer* tabs.

The *Files* tab displays all your project folders and files.

The *Project Explorer* tab displays 4 levels of logical objects:

- UI - the logical representation of the *UI* modules in the project.
- Services - the logical representation of the *Service* modules in the project.

- Data Modules - the logical representation of the *Database* modules in the project.
- External Data Modules - the logical representation of the *Database* modules in the project obtained from an external source.

i Note





External database modules are located within the *external* folder in the *Files* structure.

From the *Project Explorer* tab, you can:

- See the logical view of the project elements.
- Open any object from the structure leaves by double-clicking it.
- Add new services
- Add new data models
- Add new elements
- Delete elements





Adding Services

You can add services to SAP BTP business applications.

1. Select the *Services* folder in your project.
2. Click     (Add control).
 1. If this is the first service for your project, you can choose a new module name or accept the default one.
 2. If there are already services in your project, add a name for the new service.
3. Click *Add*.

Adding Data Models


You can add data models to SAP BTP business applications.

1. Select the *Data Models* folder in your project.
2. Click     (Add control).
3. Provide a name for the new database module that will be created in your project.
4. Click *Add*.


Adding Elements

You can add views to Fiori Elements and freestyle projects only.


For Fiori Elements projects:

1. Select the Fiori Elements *Views* container.
2. Click  (Add control).
3. Enter the required details in the *Add Object Page* dialog box. (This is the same logic used in the **Storyboard**.)


For freestyle projects:

1. Select the *Views* container in the freestyle project.
2. Click  (Add control).
3. Enter the details required in the *New SAPUI5 View* wizard. (This is the same logic used in the **New** *SAPUI5 View* menu.)


For data models:

1. Open the *Data Models* container in the project.
2. Select the desired namespace within the data model.
3. Click  (Add control).
4. Provide a name for the entity.

For services:

1. Open the *Services* container in the project.
2. Select the desired service.
3. Click  (Add control).
4. Provide a name for the entity set.

For UI applications:

1. Select the *UI* container in the project.
2. Click  (Add control).
3. Enter the details required in the *New SAPUI5 Application* wizard. (This is the same logic used in the **New** *HTML5 Module* menu.)

Deleting Elements

- You can delete all entities within a database module.
- You can delete an attribute within an entity.
- You can delete an entity set within a service.
- You can delete an attribute within an entity set.

Parent topic: [SAP Web IDE Basics \[page 72\]](#)

Related Information

[Navigating SAP Web IDE \[page 72\]](#)

[Status Bar \[page 77\]](#)

[Workspace Manager \[page 78\]](#)

[Search Options \[page 79\]](#)

[Resizing Panes \[page 85\]](#)

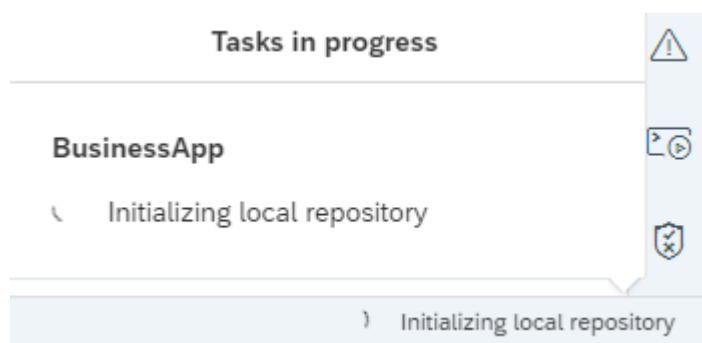
[Keyboard Shortcuts \[page 85\]](#)

6.3 Status Bar

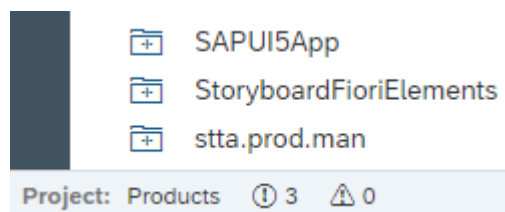
The status bar enables you to see which tasks are running in the background and their status.

An indication that one or several tasks are running is displayed at the bottom-right corner of the screen.

Click on the text to see the status of each process per project and cancel any process from the popup.



At the bottom-left corner of the screen you can see the selected project. If there are any problems or warnings, they are indicated in the status bar as well. Double-clicking these icons will open the Problems View.



Parent topic: [SAP Web IDE Basics \[page 72\]](#)

Related Information

[Navigating SAP Web IDE \[page 72\]](#)

[Working in the Workspace \[page 74\]](#)

[Workspace Manager \[page 78\]](#)

[Search Options \[page 79\]](#)

[Resizing Panes \[page 85\]](#)

[Keyboard Shortcuts \[page 85\]](#)

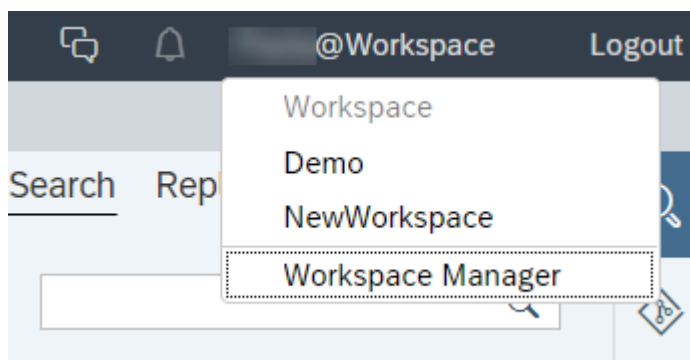
6.4 Workspace Manager

Work in multiple workspaces to organize your projects better. You can easily switch between workspaces.

Using the Workspace Manager

You can use the [Workspace Manager](#) to do the following:

- Create one or more new workspaces.
 - Open another workspace in the current browser tab or a new one.
 - Delete a workspace, including all its related projects and settings.
 - See which features (extensions) are enabled with each workspace.
1. Open the [Workspace Manager](#) from the `<name>@Workspace` menu or [Tools](#) menu. The [Workspace Manager](#) opens in the currently open tab.



Create a New Workspace

1. On the [Workspace Manager](#) screen, on the upper right side of the screen, choose the [Create Workspace](#) button.
2. In the [New Workspace](#) dialog box, enter a name that contains only letters, numbers, underscores (_) and hyphens (-) and choose [Create Workspace](#).
The new workspace appears in the list in the [Workspace Manager](#) screen. You may open it and immediately start adding content.

i Note

SAP Web IDE allows you to create up to 15 workspaces.

Open an Existing Workspace

From the *Workspace Manager* screen, you can open any of your workspaces as follows:

- *Open*
Opens the workspace in the same tab, which closes the *Workspace Manager*.
- *Open in new tab*
Opens the workspace in a new tab, leaving the *Workspace Manager* open in its tab.

Delete a Workspace

In the *Workspace Manager* screen, click the *Delete*  icon of the workspace you want to delete.

i Note

When you delete a workspace, all of its related projects and settings are permanently deleted.

Switch Between Workspaces

You can switch between workspaces by selecting the desired option from the *<name>@Workspace* menu.

Parent topic: [SAP Web IDE Basics \[page 72\]](#)

Related Information

[Navigating SAP Web IDE \[page 72\]](#)

[Working in the Workspace \[page 74\]](#)

[Status Bar \[page 77\]](#)

[Search Options \[page 79\]](#)

[Resizing Panes \[page 85\]](#)

[Keyboard Shortcuts \[page 85\]](#)

6.5 Search Options

SAP Web IDE's search options include a simple search within an open file in the code editor, advanced search and replace across multiple files in a project, and a search to find references.

[Find and Replace in an Open File \[page 80\]](#)

Perform a simple find and replace within a single open file from the code editor.

[Search in the Workspace \[page 81\]](#)

Search for a string within a folder, a project, or your workspace.

[Find References \[page 84\]](#)

You can find and go to all references of a specific function, variable, or object property, or find all files that reference a specific JavaScript file.

Parent topic: [SAP Web IDE Basics \[page 72\]](#)

Related Information

[Navigating SAP Web IDE \[page 72\]](#)

[Working in the Workspace \[page 74\]](#)

[Status Bar \[page 77\]](#)

[Workspace Manager \[page 78\]](#)

[Resizing Panes \[page 85\]](#)

[Keyboard Shortcuts \[page 85\]](#)




6.5.1 Find and Replace in an Open File

Perform a simple find and replace within a single open file from the code editor.

Context

These searches are limited to the file that is currently in view. To search and replace across multiple files, perform an advanced search. See [Search in the Workspace \[page 81\]](#).

Procedure

1. Open the file that you want to perform a simple string search in.
2. Select  **Search**  **Find and Replace** .
3. Do one of the following:
 - To perform a basic search only, enter the search string in the search field (the first field). Use the adjacent up and down arrows to find instances previous to or following the current cursor location. Use the icons to further limit the search for:
 - Regular expressions
 - Case-sensitive results
 - Whole words only
 - Search in selection

- To replace text, enter both a search string and a *Replace with* string. To search and replace one by one, click *Replace*. If other instances exist, the next one will be selected. To automatically search and replace all instances, click *All*.

i Note

You cannot undo the global replace operation.

4. If you replaced strings, save the file.

Task overview: [Search Options \[page 79\]](#)

Related Information

[Search in the Workspace \[page 81\]](#)

[Find References \[page 84\]](#)

6.5.2 Search in the Workspace

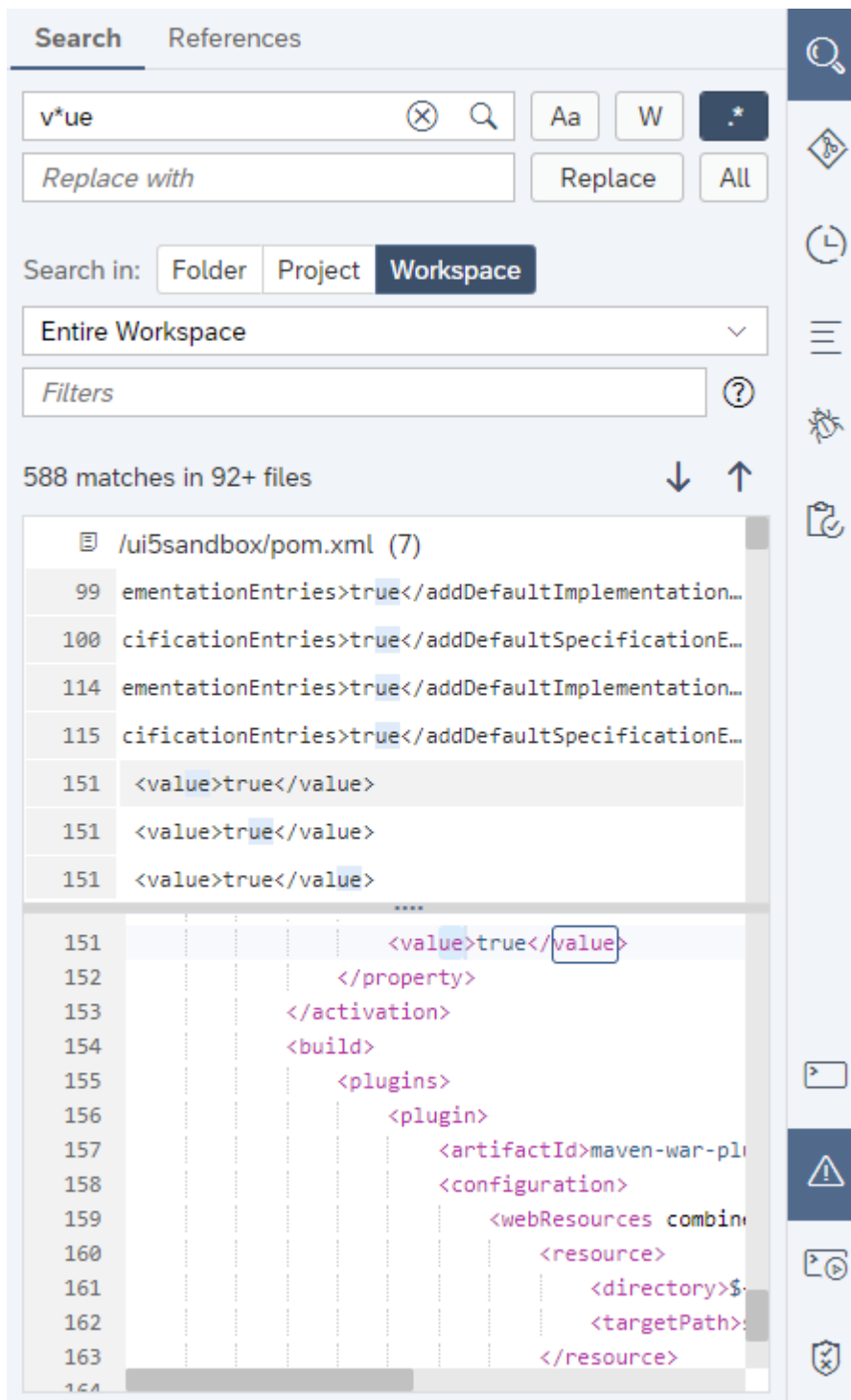
Search for a string within a folder, a project, or your workspace.

Context

To determine the success of your search, use the preview to quickly scan the results, and refine your search further as needed.

Procedure

1. Open the *Search* pane by choosing the search icon in the right sidebar.







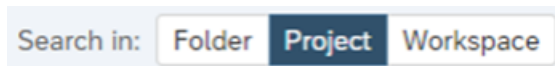
2. Enter the string you want to find and click `Enter`.

Note

Use abbreviated uppercase notation to search for words in CamelCase. For example, enter GTZ to find GetTimeZone.

3. If needed, use the following to narrow your search:

- a.  to perform a case-sensitive search.
 - b.  to search for whole words.
 - c.  to search for regular expressions. To learn about which regular expressions are supported by SAP Web IDE, see [here](#) .
4. If needed, use Folder, Project, or Workspace to define the scope of your search.



- a. If you click *Folder*, the dropdown list contains a list of all the folders in the selected project.
- b. If you click *Project*, the dropdown list contains a list of all the projects in the workspace.

i Note

Selecting a project or folder in the workspace sets the search scope in the *Search* pane to that selected project or folder.

5. If needed, enter wildcards to filter your results.
- Use an asterisk (*) as a substitute for a part of a file or folder name.
 - Use a backslash (/) to search only in specific folders.
 - Use an exclamation mark (!) to exclude a file name or folder.
 - Use CamelCase abbreviations to search only in files that have CamelCase names. For example, entering BCP will search in these files:
 - .../BreadCrumbsPage.js
 - .../BlockConnectionProtocol.java
 - .../LogBigClockPivotData.xml
6. In the results list at the bottom of the *Search* pane, you see search results, organized according to file.
7. Click a row to see the file's preview in the bottom part of the pane.
8. When you have identified the file and row that you were looking for, double-click the specific search result to open the file with the appropriate row highlighted.

Task overview: [Search Options \[page 79\]](#)

Related Information

[Find and Replace in an Open File \[page 80\]](#)

[Find References \[page 84\]](#)

6.5.3 Find References

You can find and go to all references of a specific function, variable, or object property, or find all files that reference a specific JavaScript file.

Context

i Note

You can find references in projects that contain up to 80 files.

Procedure

1. In an open JavaScript file in the code editor, right-click a function, variable, or object property – or in the workspace, right-click a JavaScript file – and choose *Find References* from the context menu.

The search results are displayed in the *References* tab in the *Search* pane and are grouped by document. The file name and code line for each reference is displayed.

2. Click a link to open the file and go to the code call for the function, variable, or object property, or to the file that references your JavaScript file.

Task overview: [Search Options \[page 79\]](#)

Related Information

[Find and Replace in an Open File \[page 80\]](#)

[Search in the Workspace \[page 81\]](#)

6.6 Resizing Panes

You can show and hide various panes, as well as change their sizes, to fit your way of working.

Context

You can make the following adjustments:

- Show and hide the workspace by choosing **View > Workspace**.
- Change the sizes of the workspace, editors area, and panes by dragging the splitters between them.
- Maximize an editor window (that is, hide the workspace and any panes on the right) by double-clicking the title bar of the editor window.

Note

Double-click again to restore the editor window to its original size. This always redisplay the workspace, and displays any previously opened pane on the right.

- Restore SAP Web IDE pane defaults by choosing **View > Reset to Default**.

Changes to the layout are saved, so the SAP Web IDE will appear the same the next time you open it.

Task overview: [SAP Web IDE Basics \[page 72\]](#)

Related Information

[Navigating SAP Web IDE \[page 72\]](#)

[Working in the Workspace \[page 74\]](#)

[Status Bar \[page 77\]](#)

[Workspace Manager \[page 78\]](#)

[Search Options \[page 79\]](#)

[Keyboard Shortcuts \[page 85\]](#)

6.7 Keyboard Shortcuts

You can use keyboard shortcuts to perform actions in SAP Web IDE.

The following shortcuts are available in SAP Web IDE:

i Note

- The majority of these shortcuts are available only for English language keyboards.
- On Mac OS keyboards, the `Alt` key is also called the `Option` key.

You can modify the SAP Web IDE predefined keyboard shortcuts as described in [Customize Keyboard Shortcuts \[page 88\]](#).

Action	Microsoft Windows Keyboard Shortcut	Mac OS Keyboard Shortcut
New file	<code>Ctrl</code> + <code>Alt</code> + <code>N</code>	<code>Command</code> + <code>Alt</code> + <code>N</code>
New folder	<code>Ctrl</code> + <code>Alt</code> + <code>Shift</code> + <code>N</code>	<code>Command</code> + <code>Alt</code> + <code>Shift</code> + <code>N</code>
New project	<code>Ctrl</code> + <code>Alt</code> + <code>Shift</code> + <code>O</code>	<code>Command</code> + <code>Alt</code> + <code>Shift</code> + <code>O</code>
New extension project	<code>Ctrl</code> + <code>Alt</code> + <code>Shift</code> + <code>E</code>	<code>Command</code> + <code>Alt</code> + <code>Shift</code> + <code>E</code>
Close file	<code>Alt</code> + <code>W</code>	<code>Alt</code> + <code>W</code>
Close all files	<code>Alt</code> + <code>Shift</code> + <code>W</code>	<code>Alt</code> + <code>Shift</code> + <code>W</code>
Save file	<code>Ctrl</code> + <code>S</code>	<code>Command</code> + <code>S</code>
Save all files	<code>Ctrl</code> + <code>Shift</code> + <code>S</code>	<code>Command</code> + <code>Shift</code> + <code>S</code>
Show Code Completion Suggestions	<code>Ctrl</code> + <code>Space</code>	<code>Ctrl</code> + <code>Space</code>
Undo	<code>Ctrl</code> + <code>Z</code>	<code>Command</code> + <code>Z</code>
Redo	<code>Ctrl</code> + <code>Y</code>	<code>Command</code> + <code>Y</code>
Cut	<code>Ctrl</code> + <code>X</code>	<code>Command</code> + <code>X</code>
Copy	<code>Ctrl</code> + <code>C</code>	<code>Command</code> + <code>C</code>
Paste	<code>Ctrl</code> + <code>V</code>	<code>Command</code> + <code>V</code>
Rename file or folder	<code>F2</code>	<code>F2</code>
Delete	<code>Del</code>	<code>Del</code>
Move to the tab on the right	<code>Alt</code> + <code>R</code>	<code>Alt</code> + <code>]</code>
Move to the tab on the left	<code>Alt</code> + <code>Q</code>	<code>Alt</code> + <code>[</code>
Navigate back to the previous file	<code>Ctrl</code> + <code>Alt</code> + <code>R</code>	<code>Command</code> + <code>Alt</code> + <code>R</code>
Navigate forward	<code>Ctrl</code> + <code>Alt</code> + <code>Y</code>	<code>Command</code> + <code>Alt</code> + <code>Y</code>
Navigate to the file that was edited last	<code>Ctrl</code> + <code>Shift</code> + <code>9</code>	<code>Command</code> + <code>Shift</code> + <code>9</code>
Show/Hide all characters	<code>Ctrl</code> + <code>I</code>	<code>Command</code> + <code>I</code>
Toggle line comment	<code>Ctrl</code> + <code>/</code>	<code>Command</code> + <code>/</code>
Toggle line comment (German language keyboard)	<code>Alt</code> + <code>7</code>	<code>Alt</code> + <code>H</code>

Action	Microsoft Windows Keyboard Shortcut	Mac OS Keyboard Shortcut
Toggle block comment	Ctrl + Shift + /	Command + Shift + /
Toggle block comment (German language keyboard)	Ctrl + Shift + 7	Command + Shift + 7
Add todo comment	Ctrl + Alt + T	Command + Alt + T
Indent line	Tab	Tab
Outdent line	Shift + Tab	Shift + Tab
Move lines up	Alt + Up Arrow	Shift + Tab
Move lines down	Alt + Down Arrow	Alt + Down Arrow
Copy lines up	Alt + Shift + Up Arrow	Alt + Shift + Up Arrow
Copy lines down	Alt + Shift + Down Arrow	Alt + Shift + Down Arrow
Beautify file format	Ctrl + Alt + B	Command + Alt + B
Generate JSDoc Comment	Ctrl + Alt + J	Command + Alt + J
Goto JavaScript definition	Ctrl + Alt + G	Command + Alt + G
Run	Alt + Shift + R	Alt + Shift + R
Run without frame	Ctrl + Alt + Shift + R	Command + Alt + Shift + R
Find	Ctrl + F	Command + F
Find and replace	Ctrl + H	Command + H
Find references	Ctrl + Alt + W	Ctrl + Alt + W
Advanced repository search	Ctrl + Shift + F	Command + Shift + F
Maximize/Restore Active Editor	Ctrl + M	Command + M
View console	Ctrl + Shift + M	Command + Shift + M
View Git pane	Ctrl + Shift + V	Command + Shift + V
View outline	Ctrl + Shift + U	Command + Shift + U
View Problems	Ctrl + Alt + P	Command + Alt + P
Open Preferences perspective	Ctrl + .	Command + .
Open Extensibility pane	Ctrl + Shift + E	Command + Shift + E
Open Resource	Ctrl + Shift + R	Command + Shift + R
Open <i>Quick Access</i> dialog box	Ctrl + 3	Command + 3
Go to line	Ctrl + L	Command + L

Action	Microsoft Windows Keyboard Shortcut	Mac OS Keyboard Shortcut
Refactor	<code>Alt</code> + <code>J</code>	<code>Alt</code> + <code>J</code>
Open i18n	<code>Alt</code> + <code>I</code>	<code>Alt</code> + <code>I</code>

Parent topic: [SAP Web IDE Basics \[page 72\]](#)

Related Information

[Navigating SAP Web IDE \[page 72\]](#)

[Working in the Workspace \[page 74\]](#)

[Status Bar \[page 77\]](#)

[Workspace Manager \[page 78\]](#)

[Search Options \[page 79\]](#)

[Resizing Panes \[page 85\]](#)

6.7.1 Customize Keyboard Shortcuts

You can modify the SAP Web IDE predefined keyboard shortcuts according to your preferences by recording them.

Context

Follow the instructions below to change a keyboard shortcut.

Caution


Using certain browser shortcuts can trigger the assigned browser action. For example, the shortcuts below are already assigned in the Chrome browser and should not be used as SAP Web IDE shortcuts.

- `Ctrl` + `N`
- `Ctrl` + `T`
- `Ctrl` + `W`
- `Ctrl` + `Shift` + `N`
- `Ctrl` + `Shift` + `T`
- `Ctrl` + `Shift` + `W`

Procedure

1. In the *Preferences* area, choose *Keyboard Shortcuts*.
2. In the *Shortcut* column, double-click a shortcut that you want to modify.
3. Record your new shortcut by pressing a key combination on your keyboard.

i Note

You can revert to the SAP Web IDE default setting by clicking the *Revert*  button.

4. Choose the *Save* button.

Results

The new customized keyboard shortcuts can now be used.

To use your customized shortcuts in SAP Web IDE, select the *User-Defined* option and then choose the *Save* button. You can switch between the SAP Web IDE default shortcuts and your customized shortcuts without losing your shortcut definitions. If needed, you can remove all your modified shortcuts by choosing the *Clear User-Defined* button. This action restores the SAP Web IDE default shortcuts and deletes your customized shortcuts.

Related Information

[Keyboard Shortcuts \[page 85\]](#)

7 Developing

SAP Web IDE supports development of Web applications as well as multi-target applications.



- [Developing Multitarget Applications \[page 438\]](#)
- [Developing Web Applications \[page 91\]](#)

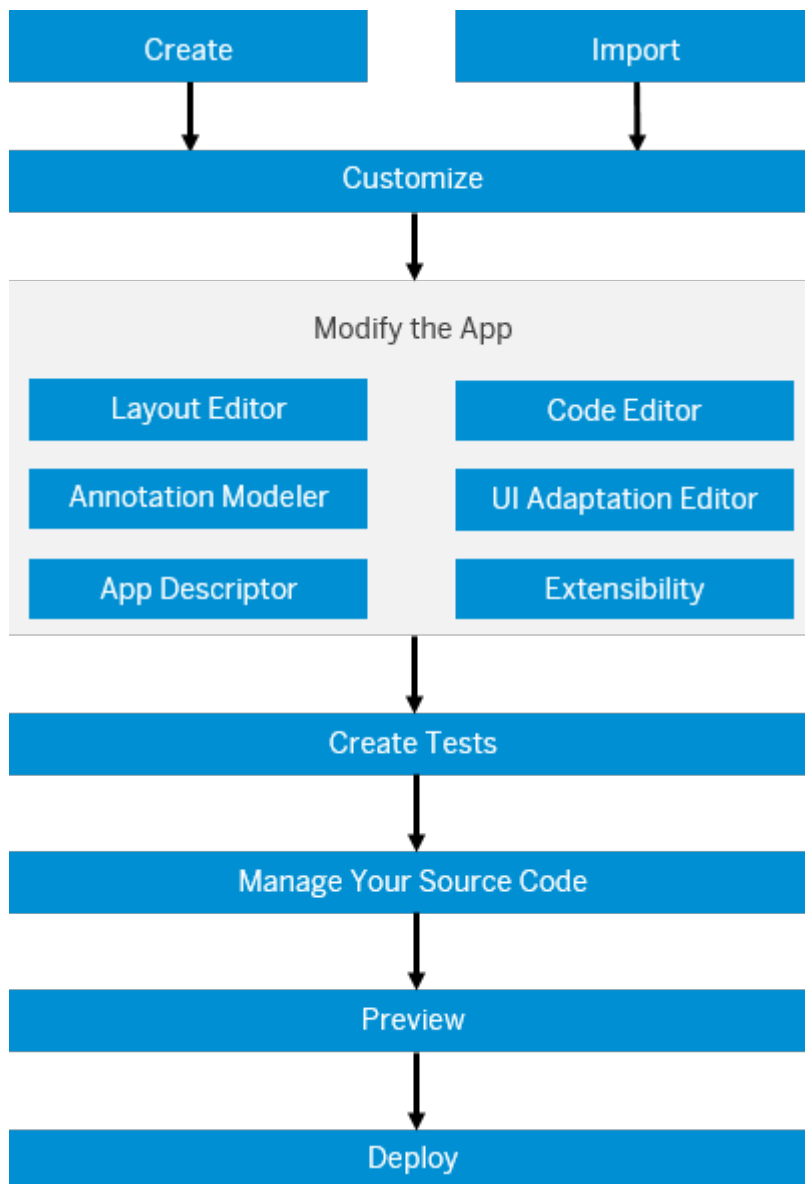
Related Information

[Developing Web Applications \[page 91\]](#)

[Developing Multitarget Applications \[page 438\]](#)

7.1 Developing Web Applications

SAP Web IDE supports two of the life cycle stages of Web applications: development and stabilization.



- [Importing Projects \[page 104\]](#)
- [Creating Projects \[page 92\]](#)
- [Customizing Your Project \[page 109\]](#)
- [Layout Editor \[page 256\]](#)
- [Developing Applications \[page 129\]](#)
- [Annotation Modeler \[page 300\]](#)

-
- [SAPUI5 Visual Editor Usage \[page 355\]](#)
- [Developing Application Tests \[page 253\]](#)
- [Using Source Control \(Git\) \[page 363\]](#)
- [Running Applications in Development Mode \[page 391\]](#)
- [Modifying the Application Descriptor Configuration \[page 121\]](#)
- [Deploying Applications \[page 414\]](#)

7.1.1 Creating Projects

SAP Web IDE offers various methods for creating new projects. However, even before selecting a method, make sure that you fully understand the application that you are creating and the data sources that the application connects to.

[Create Projects from a Template \[page 92\]](#)

You can create a new project for an application based on a specific template.

[Create Projects from a Sample Application \[page 103\]](#)

You can create a new project based on an existing application which is used as a reference.

[Add a New Component \[page 103\]](#)

You can extend an application project and customize it to suit your needs by adding components to it.

Related Information

[Creating a Quick Start Application with the Layout Editor \[page 274\]](#)

7.1.1.1 Create Projects from a Template

You can create a new project for an application based on a specific template.

Procedure

1. From the *File* menu, choose **► New ► Project from Template ▾**.
2. Select the relevant template based on the project that you want to generate:
 - From the *Environment* dropdown list, select the relevant environment for your project. The templates that can be used for deploying projects to the selected environment are displayed.
The generated project structure depends on the selected environment. If you selected *Cloud Foundry*, an MTA project is created. (The project name is mta_<modulename>.)

i Note

Your choice of environment is used by default the next time you open the new project wizard.

- Use the [Category](#) filter to only show templates for creating a specific type of application.
 - Use the [SAPUI5 Version](#) to show only SAPUI5-based templates that work with the selected version. Templates that do not depend on SAPUI5 are always shown.
 - Click the heart at the top of any template tile to make it a favorite. It is then included in the [Favorites](#) category.
3. Enter a name for your project and choose [Next](#).

i Note

The project name cannot be changed once the project has been generated.

4. If relevant for your template, enter the required App Descriptor data and choose [Next](#).
5. If relevant for your template, select the data source (OData service) on which you want to base your project, in one of the following ways:
- Choose [Service Catalog](#), select the desired data source, and then select a service.

i Note

If the data source is an API Management system, you are required to select a product to see the service details.

- Choose [Workspace](#) and browse for the relevant metadata in the SAP Web IDE system.
- Choose [File System](#) and browse for the relevant metadata in your file system.
- Choose [Service URL](#) and select the desired data source from the list. Then paste the relevant URL in the field beneath the data source.

i Note

If the system belongs to an API Management service, you are required to enter an application key in the relevant field.

For more information, see [Select a Data Source \[page 95\]](#).

After you select the data source, the service details are displayed.

i Note

If you select an OData service, a `model` folder containing the `metadata.xml` file is created automatically when the project is generated.

i Note

If you select a data source from a local file (using [Browse](#)), the created application project can only be run using mock data, unless the service URL is manually added to the generated application code. For more information, see [Running Applications in Development Mode \[page 391\]](#).

Choose [Next](#).

6. If relevant for your template, select an annotation file for your OData service.

If the metadata received from the OData service contains annotations, the metadata file is added to the Annotation Files table.

- a. Click [Add Annotation Files](#) and select the relevant data source where the annotation file is located, in one of the following ways:

- Choose [From Service](#) to select one or more annotation files provided by the OData service.

i Note

If the OData service provides one annotation file only, it is added to the list by default.

- Choose [File System](#) and browse for the relevant annotation file in your file system.
 - Choose [Workspace](#) and browse for the relevant annotation file in the SAP Web IDE system.
 - Choose [Annotation URL](#) and select the desired data source from the list. Then paste the relevant annotation URL in the field beneath the data source.
- b. Use the up and down arrows in the table to set the order in which the files will be loaded.

i Note

You cannot delete or change the ranking of metadata files that contain annotations.

7. Choose [Next](#).
8. If relevant for your template, in the next wizard steps, customize the template parameters and choose [Next](#).

i Note

If your project requires a namespace, make sure that it does not start with a reserved word.

9. Confirm your project information and choose [Finish](#).
The project wizard creates the project structure in the workspace under a new folder with the project name that you specified.

7.1.1.1.1 Worklist Template

The [SAP Fiori Worklist Application](#) template implements a typical worklist floorplan, one of the patterns that is specified by the SAP Fiori Design Guidelines.

A worklist displays a collection of items to be processed by the user and usually involves reviewing details of a list item and taking action. If the data needs to be organized into columns or the overview of the items is more important than showing the item details directly, this template can be used as a starting point.

For more information, see the SAPUI5 documentation of the [Worklist Template](#).

7.1.1.1.2 Master-Detail Template

The SAP Fiori Master-Detail Application template implements a typical split-screen layout, one of the design patterns that is specified by the SAP Fiori Design Guidelines.

The split-screen layout is optimized for displaying and processing a list of items. On the left side of the screen, users can quickly scan and navigate through the list. On the right side, they then see the details of the selected item, and can trigger related actions or edit the data. Use this template if the users need to review and process different items quickly with minimal navigation.

For more information, see the SAPUI5 documentation of the [Master-Detail Template](#).

7.1.1.1.3 Select a Data Source

SAP Web IDE helps you explore existing OData services deployed on various SAP systems, in your organization, and select one.

Prerequisites

You are selecting an OData service from the service catalog or using a service URL. The list of systems that you can select is based on the destinations configured in your SAP BTP account.

If you do not have a data source, you can create a new one. See [Create New Data Source \[page 99\]](#).

For [Service Catalog](#), the list of systems consists of all destinations where the **WebIDEUsage** property is configured with one of the following values:

- `odata_abap`: for accessing OData services hosted by Gateway systems (corresponds to URL path `/sap/opu/odata`)
- `odata_smp`: for accessing OData Services in an SAP Mobile Platform system
- `odata_hcp_odp`: for accessing OData services in an OData Provisioning system on SAP BTP
- `odata_hci`: for accessing OData services in an SAP BTP Integration system. To complete the setup, you also need to configure an additional destination, representing an integration flow node, with `hci_ifl_node` as the value for WebIDEUsage property.
- `api_mgmt_catalog`: for accessing an API Management system
- `api_mgmt_proxy`: for accessing a specific OData service on an API Management system

i Note

When configuring a destination hosted on an API Management system, add both `api_mgmt_catalog` and `api_mgmt_proxy` as values for the **WebIDEUsage** property.

For [Service URL](#), the systems that you can access consists of all destinations where the **WebIDEUsage** property is configured with the value `odata_gen`.

i Note

When setting up a destination, ensure that only one of these WebIDEUsage properties are used. Do not specify multiple usages for this destination.

For more information, see [Connect to ABAP Systems \[page 57\]](#).

Context

You can use this procedure to explore various OData services and select one that meets your requirements.

Procedure

1. Select a system.

If you are selecting an OData service from the [Service Catalog](#), you can see that all the OData services hosted by the selected system, are listed in a table. If there are numerous services listed, enter the service name or part of the name in the [Search Service](#) text box. The list of services in the table is filtered to match the pattern of the name you entered.

! Restriction

Currently, OData V4 services are not listed for the [Service Catalog](#) option.

If you are selecting an OData service using the [Service URL](#), enter the relative URL of the OData service and choose [Test](#).

i Note

If **WebIDEAdditionalData** property has been set to the value `full_url`, you do not need to enter and test the URL. For more information on this property, see [Connect to ABAP Systems \[page 57\]](#).

2. Select a service.

i Note

If the selected service is hosted by an **API Management** system, you can access the service details once you select a product associated with the service. To select a product, choose [Select](#) and choose a product from the list.

3. To explore the selected service:
 - a. Expand the service in the service table to view its elements, including entity sets and function imports.
 - b. To get more information about the service, choose [Show Details](#).

! Restriction

Currently, additional information for OData V4 services, including any referenced metadata, cannot be shown.



The details of the selected service appear to the right of the table. The following table describes the service details that are listed:

Service Details	Description
Name	Name of the OData service
Description	Description of the OData service and its purpose
Status	Indicates the status of the OData service. The typical ones that you can come across are: <ul style="list-style-type: none">○ Running○ Authentication failed○ Service is unavailable
System	Name of the system that is hosting the OData service
Type	Indicates the type of service. For example, OData.
OData Version	Indicates the version of the OData protocol followed by the service. For example, 2.0 or 4.0.
Author	User that created and deployed the OData service
Last Updated Date	Date when the OData service was last updated
Service Version	Indicates the version of the OData service
Tags	Enables quicker searching of the OData service
URL	Endpoint URL of the OData service

i Note

If you are selecting an OData service using the [Service URL](#), you can only see the following service details:

- Name
- Status
- OData Version
- URL

- c. Choose *Diagram* in the right pane to view the entity relationship diagram of the OData service. To look at the diagram in more detail, choose  (*Open Full Screen*). Once you are done viewing the diagram, choose  (*Close Full Screen*).
- d. Expand an entity set or function import in the service table to view its properties.
- e. To get more information about an entity set, select it in the service table and choose *Show Details*. You can see the following detailed information about the entity set and all its properties to the right of the service table.

Entity Set Details	Description
PROPERTY NAME	Name of the property
KIND	Indicates whether the property is a primary key, a complex type, or a navigation property.
DATATYPE	Type of data held by the property. For example, Edm.String.
LENGTH	Specifies the length or size of the property
DESCRIPTION	Describes the property

- f. Choose *Annotations* in the right pane. If the selected service uses SAP Annotations, the operations that are supported by the entity set are displayed here. For example, **Creatable** and **Updatable**.
- g. Choose *Live Data* in the right pane to see live production data associated with the entity set. For more information, see [View Live Data \[page 100\]](#).
- h. To get more information about a specific property, select it in the service table and choose *Show Details*. You can see detailed information about the property to the right of the service table.
- i. To get more information about a function import, select it in the service table and choose *Show Details*. You can see the following detailed information about the function import to the right of the service table.

Function Import Details	Description
Function Import	Name of the function import
Return Type	Specifies the scalar, entity, or complex type, or a collection of scalar, entity, or complex types returned by the function import
Method Type	Specifies the HTTP method that triggers the function import. For example, GET or POST.
Entity Set	If the return type is an entity or collection of entities, this field specifies the name of the entity set to which the returned entities belong.

The input and output parameters (if any) of the function import are also listed in a table.

4. Once you have selected the required OData service, proceed to the next step. If you are creating a project from a template, see [Create Projects from a Template \[page 92\]](#) for more information.

7.1.1.1.3.1 Create New Data Source

You can reach new data sources by creating from SAP Web IDE a new destination in your Neo environment. This saves you the need to go to the SAP BTP cockpit.

Context

To create a destination, you must be a member of the SAP BTP account.

Once you create the destination with the [New Project from Template](#) wizard or from the [Tools](#) menu, the data source is available in the system dropdown list.

Procedure

1. In the [Data Connection](#) step of the [New Project from Template](#) wizard, select [create a new data source](#).
2. In the [New Data Source](#) dialog, fill out the following fields:

Field	Description
Destination Name	A name for the destination.
Destination Description	A description for the destination.
Neo Environment User ID	Your SAP BTP user ID.
Neo Environment Password	Your SAP BTP password.
URL	The new destination URL.
Proxy Type	You can select Internet or On Premise. If you select On Premise, you have to set up the cloud connector as described here .
Authentication	Select No Authentication or Basic Authentication .

Field	Description
WebIDEUsage	Select one of the following values: <ul style="list-style-type: none"> ○ odata_gen: for generic OData service URL. ○ odata_abap: for accessing OData services hosted by Gateway systems (corresponds to URL path /sap/opu/odata). ○ odata_smp: for accessing OData Services in an SAP Mobile Platform system ○ odata_hcp_odp: for accessing OData services in an OData Provisioning system on SAP BTP

3. Select *OK*.

This is basic destination functionality, tailored for SAP Web IDE projects. To enhance or maintain destinations, go to the cockpit. For more information, see [Create HTTP Destinations](#).

7.1.1.1.3.2 View Live Data


SAP Web IDE helps you view live production data associated with an entity set of an OData service.

Context

You are selecting an OData service from the service catalog or using a service URL. You can view live production data associated with the entity sets of that service. Viewing live production data just might be the edge you need, to choose the right service.


You can use the following procedure to view live production data.

Procedure

1. Select an entity set of an OData service and choose *Show Details*. The details of the selected service appear to the right of the table.
2. Choose *Live Data* in the right pane. You can see a preview of the live production data associated with the entity set. This preview includes only the top 100 records and 5 properties of that entity set.
3. To see all the records and properties of the entity set, choose  (*Open Full Screen*).





i Note



The order in which you see the properties in this view may not match that of the preview.

4. To customize your view of the data in the entity set, choose  (*Settings*). The *View Settings* dialog box appears.

5. You can perform any of the following customizations:
- On the *Columns* tab, select only those properties that you wish to see as part of the live data. You can also search for a property by typing its name in the text box.

Choose *Show Selected* to see only the selected properties in the tab. This proves handy, especially when you are dealing with many properties.



- On the *Columns* tab, select a property and choose  (*Move to Top*),  (*Move Up*),  (*Move Down*), or  (*Move to Bottom*) to rearrange its position in the live data view.
- On the *Sort* tab, select the property using which you want to sort the data. Also, select whether you want to sort it in the ascending or descending order.


Choose  (*Add Sort Criterion*) to add more properties for sorting the data. You can remove properties used for sorting by choosing  (*Remove Sort Criterion*).

- On the *Filter* tab, add a filter by selecting the property, selecting the operation, and entering a value.

You can specify two types of filters:

- Include:** These filters include all data that matches the condition specified by the filter. You can define filtering conditions that match a range of values or a specific value.
- Exclude:** These filters exclude all data that matches the condition specified by the filter. You can only set filtering conditions that match a specific value.

Choose  (*Add Filter*) to add more filters. You can remove filters by choosing  (*Remove Filter*).

- To restore the default view, choose *Restore*.
- To implement your customizations, choose *OK*. The live data view refreshes with the customizations you have specified.
- To close the full screen view of the live data, choose  (*Close Full Screen*).

Related Information

[Select a Data Source \[page 95\]](#)

7.1.1.1.4 Consume APIs from SAP API Business Hub

You can now use the OData APIs that have been published on the SAP API Business Hub to create an SAP Fiori application in SAP Web IDE. You can create the application using the OData APIs, and test it using the sandbox environment of the SAP API Business Hub.

Prerequisites

- You have accepted the Terms of Use for [SAP API Business Hub](#). For more information about SAP API Business Hub Terms of Use, see [here](#).

Procedure

1. Create your project using the required template.

i Note

If you need to use OData APIs from the SAP API Business Hub, you must choose a template that gives you access to a service catalog. For example, the SAP Fiori Master Master-Detail Application template.

2. In the *Basic Information* tab, provide a name for your project.
3. In the *Data Connection* tab, select *SAP API Business Hub* service from the *Sources* list.
All the API packages that are available on SAP API Business Hub are listed when you click on the dropdown list.
4. Choose an API package from the dropdown list.
All APIs of the selected API package are listed.
5. Choose the required API from the list. You will be prompted to log in using your SAP credentials. The SAP credentials are asked only once per session.
You can search for an API within an API package using the **Search** functionality. Upon choosing an API, the API description and its details are displayed on the right-hand pane.
To experience the API, you can click on the *Open in API Hub* link.
6. Choose *Next*.
7. In the *Template Customization* tab, enter the required information to create a UI binding and click *Next*.
8. In the *Confirmation* tab, click *Finish* to create the project.
9. In your workspace, run the created project as a Web Application.

i Note

Depending on the type of template you have chosen, the options to run the project varies. Choose the relevant option.

You can see the data derived from the OData API you chose.

7.1.1.2 Create Projects from a Sample Application

You can create a new project based on an existing application which is used as a reference.

Procedure

1. From the *File* menu, choose ► *New* ► *Project from Sample Application* ⌵.
2. Select the relevant sample application and choose *Next*.

i Note

You can only use a sample application once to create a project.

3. Read and accept the terms of use.
4. Choose *Finish*.
The wizard creates a project containing the application files in the workspace.

i Note

Make sure to use this project as a sample only and not for productive usage.

7.1.1.3 Add a New Component

You can extend an application project and customize it to suit your needs by adding components to it.

Procedure

1. From the workspace, select and then right-click the desired project or folder to which you want to add a new component.
2. Select ► *New* ► *<desired component>* ⌵. Only the templates relevant for the selected project are displayed. If there are more than four options, choose *More...* and select the desired template based on the component that you want to generate from the wizard.
3. If relevant for your template, select the data source (OData service) on which your project will be based, in one of the following ways:
 - Choose *Service Catalog* and select the desired data source from the list. Once you select the desired data source, choose a service and then choose *Select*.
 - Choose *Workspace* and browse for the relevant metadata in the SAP Web IDE system.
 - Choose *File System* and browse for the relevant metadata in your file system.
 - Choose *Service URL* and select the desired data source from the list. Then paste the relevant URL in the field beneath the data source.

i Note

If the system belongs to an API Management service, you are required to enter an application key in the relevant field.

After the data source is selected, the service details are displayed.

i Note

If you select an OData service, a `model` folder containing the `metadata.xml` file is automatically created during the project generation.

i Note

If the data source is selected from a local file (using Browse), the created application project can be run only using Mock Data, unless the service URL is manually added to the generated application code. For more information, see [Running Applications in Development Mode \[page 391\]](#).

Choose *Next*.

4. If relevant for your template, in the next wizard steps customize the template parameters and choose *Next*.
5. Confirm your component information and choose *Finish* to confirm and add your component.

7.1.2 Importing Projects

You can import the generated code for an application and the resources that are contained in a project into the SAP Web IDE workspace.

[Import Projects from an Archive \[page 105\]](#)

You can import a project and its resources from the local file system into the SAP Web IDE workspace.

[Import Projects from the Previous Version of SAP Web IDE \[page 105\]](#)

You can import your SAP Web IDE projects to SAP Web IDE Full-Stack.

[Import Applications from SAP BTP \[page 107\]](#)

You can import an existing application from SAP BTP into the SAP Web IDE workspace.

[Import Applications from the SAPUI5 ABAP Repository \[page 108\]](#)

You can import an existing application from the SAPUI5 ABAP repository into the SAP Web IDE workspace.

7.1.2.1 Import Projects from an Archive

You can import a project and its resources from the local file system into the SAP Web IDE workspace.

Prerequisites

- You must have a compressed (.zip) file of the project that you want to import.
- The size of the (.zip) file must not exceed 20MB.

Procedure

1. From the workspace, select the folder to which you want to import the project, for example, the top-level workspace folder.
2. Choose **File > Import > From File System**.

The *Import* dialog box is displayed, and the selected folder is displayed in the *Import to* field. You can change the destination folder by choosing *Select Folder* and browsing to the required folder, or typing in a new path to create a new folder.

3. In the *File* field, browse to the location of your archived project file, select it, and choose *Open*.
4. Choose *OK*. The project is created in a new folder with the same name as the archive file.

i Note

If the target folder already exists in your workspace, you are prompted to approve the overwriting of the existing files.

7.1.2.2 Import Projects from the Previous Version of SAP Web IDE

You can import your SAP Web IDE projects to SAP Web IDE Full-Stack.

Context

Pay attention to the following features that are handled differently in SAP Web IDE Full-Stack:

- **Project Metadata:** Each project in SAP Web IDE Full-Stack contains a special system/metadata folder called `.che` that is hidden by default.

When you import or Git-clone a project to the Full-Stack Development version of SAP Web IDE, a new metadata folder named `.che` is generated automatically from the `project.json` file, and all your project settings are moved to the new Full-Stack Development version.

Caution

Do not change the contents of the `.che` folder.

Note

The old `project.json` metadata file appears in the project sources but is no longer in use.

- **Application Build:** If you used the SAPUI5 client-side build in the standard SAP Web IDE to minify and bundle files before deployment, in the Full-Stack Development version of SAP Web IDE, you must use the UI5 build or the Grunt build. For more information, see [Application Build \[page 423\]](#) and [Building Applications \[page 405\]](#).

Note

We recommend all developers working on the same project use the same SAP Web IDE version. If you choose different cloud versions or use the personal edition together with the full-stack development version, you can migrate the project between SAP Web IDE versions as described in [Import Projects from the Previous Version of SAP Web IDE \[page 105\]](#). However, all changes to the project settings must be copied manually.

Procedure

1. In the previous version of SAP Web IDE or SAP Web IDE, personal edition, for each project you want to import, do one of the following:
 - **If your project is managed in Git**, commit and push your changes.
 - **If your project is not managed in Git**, export your project to a local folder.
2. Open SAP Web IDE Full-Stack.
3. Do one of the following:
 - **If your project is managed in Git**, clone your project.
 - **If your project is not managed in Git**, in SAP Web IDE Full-Stack, import the project you exported in step 1.

Next Steps

For each project you want to import, repeat this procedure.

Note

You can use a similar procedure if you want to export your project back to older versions of SAP Web IDE. However, if you (or your colleagues in cases where Git is used) change the project settings using a different version, you will have to manually update them.

Related Information

[Introducing SAP Web IDE Full-Stack \[page 51\]](#)

7.1.2.3 Import Applications from SAP BTP

You can import an existing application from SAP BTP into the SAP Web IDE workspace.

Prerequisites

- You must be working in the cloud edition of SAP Web IDE.
- Make sure that the application you want to import has an active version in SAP BTP.

Procedure

1. In SAP Web IDE, choose **File > Import > Application from SAP BTP**. The **Select Application from SAP BTP** dialog box is displayed.
2. Browse for the application that you want to import, or find it using the search mechanism
3. Select the desired application. The *Target Folder* field is populated with the selected application's name. You can edit this name if required.
4. Choose *OK*. The application is displayed in the workspace under the root folder.

i Note

You cannot import an application to an already existing project.

Importing an application from SAP BTP detaches it from the Git repository.

Results

You can follow the progress and the completion of the import process in the SAP Web IDE console. To open the console, choose **View > Console**.

A success notification message is displayed once the application is imported.

7.1.2.4 Import Applications from the SAPUI5 ABAP Repository

You can import an existing application from the SAPUI5 ABAP repository into the SAP Web IDE workspace.

Prerequisites

Make sure you have complied with all the items described in the [Requirements for Connecting to ABAP Systems \[page 60\]](#) and [Connect to ABAP Systems \[page 57\]](#) sections.

⚠ Caution

- Don't modify SAP delivered applications. Instead, you can use application extension tooling, including [extension](#) and [adaptation](#) projects.
- Artifacts deployed to the SAPUI5 ABAP repository are the results of application builds, which represent only the **productive** version of the application and **don't reflect the projects' source files**. Don't import built projects into the workspace for further development. If more development is needed, use source control (such as Git) to maintain your source code.

Procedure

1. In SAP Web IDE, choose **File > Import > Application from SAPUI5 ABAP Repository**. The *Select Application from SAPUI5 ABAP Repository* dialog box is displayed.
2. Select the desired system.
3. Browse for the application that you want to import, or find it using the search mechanism.
4. Select the application. The *Target Folder* field is populated with the selected application's name. You can edit this name if required.
5. Choose **OK**. The application is displayed in the workspace under the root folder.

i Note

You cannot import an application to an already existing project.

Results

You can follow the progress and the completion of the import process in the SAP Web IDE console. To open the console, choose **View > Console**.

A success notification message is displayed once the application is imported.

Related Information

[The SAPUI5 ABAP Repository and the ABAP Back-End Infrastructure](#)

7.1.3 Customizing Your Project

Customize the developer experience for your SAP Web IDE project.

Any project customization you make in one session persists to the next. Customizations can vary from project to project.

7.1.3.1 Configure Annotation Modeler to Use Mock Data

Use a run configuration to use annotation modeler on mock data.

Prerequisites

You have created a list report page project, an overview page application project or a module in an MTA project that is based on the list report page.

Context

When you want to use annotation modeler with the local metadata and the local copy of the back-end annotation files, configure your project to use mock data.

i Note

To use annotation modeler on real data again, select a run configuration that does not apply mock data.

You can use annotation modeler with mock data if you develop your service locally in an SAP Web IDE project or if you want to try out how annotations affect the UI.

To use mock data, in the project settings, you assign a run configuration that uses mock data.

When setting the mock data flag, the following applies:

- Metadata and OData annotations are fetched from the local URI, which are stored in the `localService` folder, as given in the `manifest.json` file.
- The mock server generates the data locally.

i Note

Productively using the mock data instead of those from the back-end systems (for example, when the remote system is temporarily not available) may lead to unexpected impact on your UI, if the metadata and annotations are updated in the back end while you are working with the local copy data.

Procedure

1. From the project context menu, select *Project Settings*.
2. From the project settings menu, select *Annotation Modeler*.
3. Select the run configuration to be used when running your project in annotation modeler. To run annotation modeler on mock data, select a run configuration that uses mock data. If there is no such run configuration available, create a new run configuration by choosing *Open Run Configurations*:
 - a. In the *Run Configuration for <project>* screen, choose the plus sign.
 - b. Choose the *Web Application* type.
 - c. Enter a matching name.
 - d. On the *General* tab page, select the `flpSandboxMockServer.html` file.
 - e. In the *Mock Data* section, choose *Run with mock data*.

Annotation modeler only uses the following settings of a run configuration:

- o Name
 - o File name
 - o Mock data
- f. Choose *OK*.
4. Make sure that you selected the right run configuration and save your changes.
 5. Refresh SAP Web IDE.

7.1.3.2 Customize JavaScript Beautifier Properties

Beautify JavaScript files to reformat the source code to make it more readable.

Context

The beautifier configuration applies to all JavaScript, `.js`, `.json`, `.xsjs`, and `.xsjslib` files in your project.

Procedure

1. From the context menu of any file in your project, select **► Project Settings ► Beautifier ► JavaScript**.
2. Determine how you want lines to be broken:

Option	Description
<i>Break lines on chained methods</i>	Select this option to add a line break when you add a chained method. By default, this option is cleared.
<i>New lines</i>	Set the maximum number of new lines between tokens. Choose from: <i>No New Lines</i> , <i>1</i> , <i>2</i> , <i>5</i> , <i>10</i> , <i>Unlimited New Lines</i> . For example, if there are two <code>\n</code> s between tokens, but you set this value to <i>1</i> , the second <code>\n</code> is removed. The default is 2.
<i>Wrapping</i>	Determine the number of characters that triggers a line wrap. A line break is inserted before the first word that reaches the limit that you set. Choose from <i>Do not wrap lines</i> , <i>40</i> , <i>70</i> , <i>80</i> , <i>110</i> , <i>120</i> , or <i>140</i> (default).

3. Determine how you want indents to be handled:

Option	Description
<i>Keep array indentation</i>	Select this option to keep the indentation of arrays as is. Clear this option to remove array indentation. By default, this option is cleared.
<i>Indent with</i>	Select the length of each code indent: <i>a Tab</i> (default), <i>2 Spaces</i> , <i>3 Spaces</i> , <i>4 Spaces</i> , or <i>8 Spaces</i> .

4. Determine additional code formatting options:

Option	Description
<i>Space before conditional</i>	Select this option to insert spaces before conditional <code>if/then</code> statements. By default, this option is selected.
<i>Unescape printable chars encoded</i>	Select this option to decode printable characters that are encoded. By default, this option is cleared.
<i>Braces</i>	Determine the positioning of braces: <i>Braces with control statement</i> (default), <i>Braces on own line</i> , or <i>End braces on own line</i> .

5. Choose *OK*.

You can now beautify any open JavaScript file according to these customized properties, by choosing *Beautify* from the context menu.

7.1.3.3 Customize Code Checking Rules

SAP Web IDE provides validators to check your code. You can customize code checking for each project.

You can customize and use the SAP Web IDE JavaScript validators and their rule configurations, or you can customize code checking for your project:

- Customize the SAP Web IDE JavaScript validator configuration with your own settings. For more information, see [Customize JavaScript Validator Configuration \[page 161\]](#).
- Configure your own JavaScript code checking rules and use them for your project. For more information, see [Create JavaScript Code Checking Rules \[page 160\]](#).

- Create your own code checking plugin, then choose the plugin in the corresponding code checking configuration pane.

7.1.3.4 Customize Code Checking Triggers

You can define the code checking process flow when pushing code to the source control repository.

Context

You can configure to run code checking when pushing any code changes to the source control repository in a specific project. You can configure to notify about code checking errors and/or warnings before the push starts and allow the user to decide whether to continue the push. You can also configure to block the push process if there are problems.

If you choose not to notify, code is pushed to the source control repository without code checking.

Procedure

1. From the context menu of any file in your project, select **► Project Settings ► Code Checking ► Triggers ▾**.
2. To receive notifications of problems found before pushing code, select *Notify before push*. Select the severity of problems for which you want notification.
3. To block the push for problems of a specific severity, select *Block push* and select the severity.

Example

Notification is configured for errors and warnings, and the push process is blocked if errors are found.

- Users who receive notifications about warnings only can choose to continue with the push or view and fix the problems before pushing.
- Users who receive notification about errors can view the problems but cannot continue with the push until the errors are fixed.

7.1.3.5 Define Application Languages

You can define the languages for your application from [Project Settings](#) so that when you preview your application, you can view it in any of the languages that you selected.

Procedure

1. From the project context menu, select [Project Settings](#).
2. From the [Project Settings](#) options, select [Languages](#).
3. From the list of [Supported Languages](#), select the languages for your project. The supported languages that you select appear in the [Default Language](#) dropdown list.
4. Select the required language from the dropdown list.
5. Choose [Save](#) and [Close](#).

The updated language settings are saved in the `.project.json` file in your application folder.

Related Information

[Create Run Configurations \[page 394\]](#)

7.1.3.6 Use SAP Translation Hub

You can translate your project's `properties` file using SAP Translation Hub.

Context

The SAP Translation Hub extension allows you to translate from English into other languages.

Before you can translate your project's `properties` file, you must configure your system to use SAP Translation Hub.

Configure the SAP Translation Hub

Procedure

1. Make sure your project is attached to a Git repository in SAP BTP.

2. Make sure that the properties files are present in the `i18n` folder.
3. Make sure that *Show hidden files* is enabled. Otherwise, you will not be able to see the translation files.
4. Make sure that the SAP Translation Hub is enabled in your account. If not, enable it:
 - a. In the SAP BTP cockpit, choose *Services* in the navigation pane.
 - b. Choose *SAP Translation Hub*.
 - c. Choose *Enable*.

If you do not see the SAP Translation Hub listed in the *Services* page, contact [SAP Store](#).

i Note

If the SAP Translation Hub is not present in your data center, you can locate and purchase it from the nearest one. See [Regions and Hosts](#).

5. Assign the *User* role to the users requiring access to SAP Translation Hub. See [Setting Up Roles and Authorizations](#).
6. Create a destination for your account. In the pane on the left, choose *Destinations*, and create a destination for your account with the following data.

Field	Entry
<i>Name</i>	translationHubBeta
<i>Type</i>	HTTP
<i>Description</i>	TranslationPlugin
<i>URL</i>	<p>Enter the URL. URLs adhere to the following naming convention: <base URL of your SAP Translation Hub account>/translationhub/</p> <p>You can copy the required URL from SAP Translation Hub as follows:</p> <ol style="list-style-type: none"> 1. In the SAP BTP cockpit, choose <i>Services</i>. 2. Choose <i>SAP Translation Hub</i> from the list of services 3. Choose <i>Go To UI for Translation Workflow</i> and copy the URL directly from the address bar.
<i>Proxy Type</i>	Internet
<i>Authentication</i>	BasicAuthentication
<i>User</i>	Enter your SAP BTP user name. The user must have been assigned to the <code>User</code> role. See Setting Up Roles and Authorizations .
<i>Password</i>	Enter your SAP BTP password.

i Note

For more information, see [Building Base URL of SAP Translation Hub](#).

7. In *Additional Properties*, enter the following data:

Field	Entry
<i>TrustAll</i>	true
<i>WebIDEEnabled</i>	true

8. Choose [Save](#); depending on the system performance, it may take several minutes for the changes to take effect.
9. Check your trust management settings in the **SAP BTP Cockpit**. Principal propagation must be enabled to ensure your SAP Web IDE identity is correctly exchanged with SAP Translation Hub. The exchange format uses SAML tokens for authentication information. To check whether principal propagation is enabled, check your [Local Service Provider](#) settings within SAP BTP Cockpit > Security > Trust.

Translate your Properties File

Context

i Note

For more information about translation projects within SAP Translation Hub, see [Create Translation Projects for Properties Files in Git Repository](#)

Procedure

1. Right-click and select [Generate Translation Files](#).
2. Select the application domain and the desired target languages.
3. Choose [Generate](#).
4. Enter your Git password and choose [Submit](#). SAP Translation Hub will then generate the translation files and push them to your Git repository. To get these files to your workspace, pull the changes from Git.

Example

For an overview of how to translate a properties file, check the following video that shows how to translate a properties file in a sample SAP Fiori app:

Contact SAP Support

Context

If you encounter any problem with the service, you can open a ticket on BCP using the `LOD-TH-STH` component.

For information about opening tickets, see [Troubleshooting and Monitoring](#).

7.1.3.7 Configure Mock Data Usage

Configure settings to run an application using a client mock server.

Context

i Note

If your application uses the application descriptor, the *Root URI* and the *Metadata File Path* are taken from the file and are not editable in the settings. If you need to make changes, they should be made in the application descriptor file.

Procedure

1. From the project context menu, select *Project Settings*.
2. From the *Project Settings* options, select *Mock Data*.
3. In the *Root URI* field, enter the URL of the service used in your project.
 - If you selected a service when you created your project, SAP Web IDE automatically populates this field.
 - If the service URL changes, edit this field accordingly.
 - If there is no service URL, leave this field empty.
4. In the *Metadata File Path* field, enter the path to the service `metadata.xml` file to fetch mock data to use in your project.

i Note

You can enter a path to the service `metadata.xml` file residing either in your current project, in a different project in your workspace, or in an application or library that you have deployed on SAP BTP. Pointing to a remote location is only possible if the appropriate application resource mapping is configured.

If you do not specify the path, SAP Web IDE looks for the mock data in the `model/metadata.xml` file. If this path changes, edit this field accordingly.

5. Select one of the following options as your *Mock Data Source*:
 - Generated data (default)
 - JSON files
6. If you have provided an additional file containing custom mock requests extending the mock server, do the following:
 - a. Select the *Add custom mock requests* checkbox.
 - b. In the *Extension File Path* field, enter the path to the `mockRequests.js` file.
7. *Save* your changes.

7.1.3.8 Set Project Types

Select project types to add type-specific behaviors to your project.

Context

Each project is configured with a base project type, for example, Web. You cannot change the base project type setting.

You can change the selection of additional project types to customize type-specific behaviors for the project.

Procedure

1. From the project context menu, select *Project Settings*.
2. From the *Project Settings* options, select *Project Types*.
The *Project Types* pane displays the base project type for the project and all the available additional project types with their descriptions.
3. Change the selection of additional project types as required.

7.1.3.9 Set the SAPUI5 Version

Configure the SAPUI5 version for your project.

Context

The SAPUI5 version you choose affects the following:

- Code completion
- Static validations
- The SAPUI5 version for the layout editor
- The default SAPUI5 version for new run configurations
- The SAPUI5 version for runtime

Procedure

1. From the project context menu, select *Project Settings*.

2. In the *SAPUI5 Settings* pane, select the design-time SAPUI5 version for your project.
You can also select *latest*, which uses the latest released version of SAPUI5.

7.1.3.10 Configure Run Configurations for the SAPUI5 Visual Editor

Select the Run Configuration to be used when running your project in the SAPUI5 Visual Editor.

Context

When an application is running in the SAPUI5 Visual Editor and not in a regular preview, there might be some run configurations that you wish to set on the application that will apply in the editor as well. For example, you might want to run it with mock data.

Procedure

1. From the project context menu, select **Project > Project Settings**.
2. In the *General* section, select *SAPUI5 Visual Editor*.

i Note

This option is only available for applications that have selected the type *Adaptation Project*.

The *SAPUI5 Visual Editor* pane displays the configurations that the application currently has in its Run Configuration.

3. Select the configuration to be used when running the app in the SAPUI5 Visual Editor.
4. Save your changes.

i Note

The following configurations will not apply in the SAPUI5 Visual Editor:

- Select application file to run.
- Run with frame.
- Use an SAPUI5 version lower than 1.38.

7.1.3.11 Use the Smart Business Service

Administrators can enable the use of the Smart Business service in SAP Web IDE.

Prerequisites

You must have an administrator role to perform this procedure.

Context

Procedure

1. Subscribe to the Smart Business service in SAP Cloud Platform cockpit. For more information, see [How to Subscribe to SAP Smart Business Service](#).

A new destination called `ssbservice` is automatically created with the following additional properties:

- `WebIDEEEnabled = true`
- `WebIDEUsage = smart_business_gen`

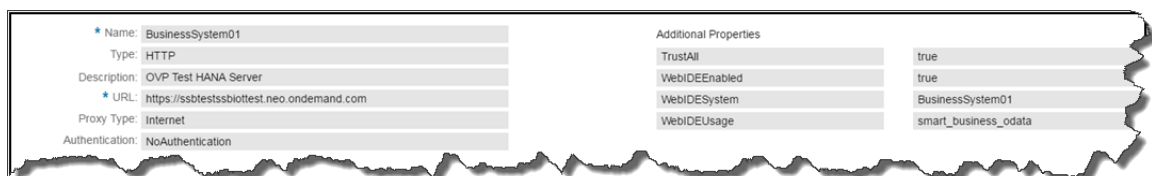
This destination is used to generate the smart business annotation.

2. Define the Smart Business OData service to be used in your application.

The service must have the following additional properties:

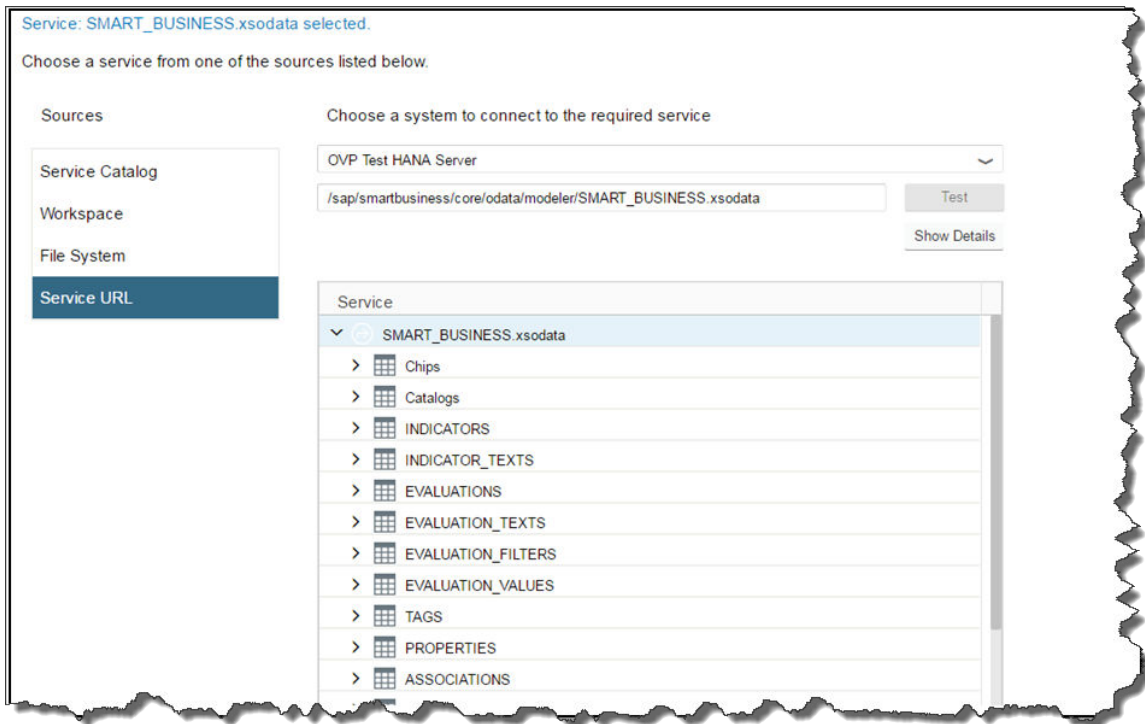
- `WebIDEEEnabled = true`
- `WebIDEUsage = smart_business_odata`
- `TrustAll = true`

This is an example of a Smart Business service destination:



3. Create an application using the Smart Business service. Use [Service URL](#) as the source for your destination.

For example:



Results

If the destination was defined properly (`WebIDEUsage = smart_business_odata`), an annotation is generated using the Smart Business service you subscribed to and it is added to the list of annotations in the *Annotation Selection* step.



7.1.4 Modifying the Application Descriptor Configuration

Use the application descriptor editor to modify your project's `manifest.json` file.

Context

You can modify your project application descriptor file (`manifest.json`) by editing the code in the application descriptor *Code Editor* or by using the *Descriptor Editor* to change the settings.

- *Code Editor*: Validation is automatically performed on the `manifest.json` file as you edit. Validation is performed first for syntax, and after confirmation of valid syntax, schema validation is performed. The schema used for validation is non-configurable. The schema errors are displayed in the corresponding line of code.
- *Descriptor Editor*: Contains configuration tabs with configuration fields, which assist you by providing available choices, placeholder suggestions, and input validation.

Changes that you make in the *Descriptor Editor* are automatically updated in the *Code Editor* and vice versa.

For more information about application descriptor attributes, see [Descriptor for Applications, Components, and Libraries](#).

Procedure

1. Open the project `manifest.json` file in your workspace.

The *Descriptor Editor* opens, containing the following tabs:

- *Settings* tab
- *Data Sources* tab
- *Models* tab
- *Routing* tab
- *Navigation* tab

2. Change the settings in each tab as required.

7.1.4.1 Settings Tab Options

General Section

This section contains general application attribute settings in the `sap.app` namespace.

Option	Description
Version	Required. The version of application descriptor schema.
ID	Required. A unique application identifier in the format, <code>sap.fiori.<appName></code> .
Type	Required. The type of application: <i>application</i> , <i>component</i> , or <i>library</i> .
Title	Required. Enter the title of the application inside double curly brackets (<code>{{}}</code>).
Description	Enter the description of the application inside double curly brackets (<code>{{}}</code>).
Source Template	The ID and version of the template from which the application was generated.
l18n File Path	The path inside the application to the properties file containing text symbols for the descriptor.
Version	Required. The version of the application.
Tags (Keywords)	Application keywords. To add a keyword, click + , and in the popup, enter the new keyword inside double curly brackets (<code>{{}}</code>).
Application Component	The application component hierarchy.

User Interface Section

This section contains UI attribute settings in the `sap.ui` namespace.

Option	Description
Technology	The UI technology. The default is <i>UI5</i> .
Devices	Select the device types on which the application can run - desktop, tablet, and phone.
Themes	The supported SAP themes that the application can use, such as <code>sap_hcb</code> , <code>sap_bluecrystal</code> , and so on.
Application Icons	The icons used in the application: <ul style="list-style-type: none"> • <i>Main</i> - Select the icon that you want to use as the main application icon. • <i>Phone 57 px</i> - Select the 57x57 pixel application icon for non-Retina iPhones. • <i>Phone 114 px</i> - Select the 114x114 pixel application icon for Retina iPhones. • <i>Tablet 72 px</i> - Select the 72x72 pixel application icon for non-Retina iPads. • <i>Tablet 144 px</i> - Select the 144x144 pixel application icon for Retina iPads. • <i>Favorites</i> - Enter the ICO file to be used in the browser and for desktop shortcuts.

SAPUI5 Section

This section contains SAPUI5 attribute settings in the `sap.ui5` namespace.

Option	Description
Minimum SAPUI5 Version	Required: The minimum SAPUI5 version required for your application.
Resources	JavaScript and CSS resources that are required by your application. To add a resource, click + , and in the popup, choose the resource type and enter its URI.
Dependencies	You can define external dependencies, such as libraries and components, which will be loaded by UI5 Core in the initialization phase of your component and can be used thereafter. To add a dependency, click + , and in the popup, choose the dependency type, <i>Library</i> or <i>Component</i> . Enter the ID and minimum version of the dependency.
Content Densities	You can define different content densities for certain controls that allow your application to adapt to specific devices: <ul style="list-style-type: none">• <i>Cozy</i>: This is the larger design - dimensions of the controls are optimized for touch-enabled devices, such as smartphones, to allow users to interact with controls more easily.• <i>Compact</i>: Reduced-size design - the font size is the same as for the cozy density, but the dimensions of the controls and the spacing between them are reduced. This density is more suitable for mouse-operated devices, such as desktops.

7.1.4.2 Data Sources Tab Options

Define the OData services and annotations that are the data sources for your application.

Services

Define the OData service to be used as the data source.

1. Click **+**.
2. Enter the service name and URI.
3. Click *OK*.
4. Configure the data source properties.

Property	Description
Name	Required: Unique name for the data source.

Property	Description
URI	Required: Relative URL in the component. The path must be relative to the location of the <code>manifest.json</code> file.
Local URI	Relative URL to the local metadata document or annotation URI. The path must be relative to the location of the <code>manifest.json</code> file.
OData Version	Version 2.0 (default) or version 4.0 .
Service Metadata	Choose Sync Metadata to sync the source file and the respective annotation files with the backend version.

Annotations

Define annotations that reference an existing data source.

1. Click **+**.
2. Change the name, if required.
3. Define the URI and local URI.
4. Use the up and down arrows to change the order of annotations in the list.

7.1.4.3 Models Tab Options

Define the data models for your application.

You can define models, and also select one to be the default model.

OData Models

Define models based on OData services. You must first define OData data sources in the [Data Sources](#) tab.

1. Click **+**.
2. Enter the model name.
3. Choose [Select data source](#).
4. From the dropdown list, select a data source.
5. Click **OK**.

The new model is added to the list of models. Select it, and you can set additional SAPUI5 properties for that model.

Other Models

Define models based on JSON, XML or other resource files, such as an i18n properties file.

1. Click **+**.
2. Enter the model name.
3. Choose *Select type*.
4. From the dropdown list, select either *JSON*, *Resource* or *XML*.
5. Click *OK*.

The new model is added to the list of models. Select it, and you can set additional SAPUI5 properties for that model. Set the URI property to indicate the location of the resource, relative to the location of the `manifest.json` file, such as `i18n/i18n.properties`.

Related Information

[Data Sources Tab Options \[page 123\]](#)

7.1.4.3.1 Batch Control for OData Services

You can enable or disable batch processing for OData services.

OData Version 2 Services

1. From the workspace, open your project's `manifest.json` file.
2. Select the *Descriptor Editor* tab at the bottom of the page.
3. Select the *Models* tab.
4. From the *Use Batch* dropdown list, select *true* to enable requests to be grouped into one batch request, or *false* to disable batch processing.

For more information, see [OData V2 Model](#) and [class sap.ui.model.odata.v2.ODataModel](#) in the SAPUI5 Demo Kit.

OData Version 4 Services

1. From the workspace, open your project's `manifest.json` file.
2. Select the *Descriptor Editor* tab at the bottom of the page.
3. Select the *Models* tab.
4. From the *Group ID* dropdown list, select *\$auto* to enable requests to be grouped into one batch request, or *\$direct* to disable batch processing.

For more information, see [Batch Control](#).

7.1.4.4 Routing Tab Options

Define routes and targets for the application.

Default Configuration

Define the default route and target properties.

Option	Description
View Path	Prefix that precedes the view name. For example, if the view name is <code>myView</code> and the <code>View Path</code> is <code>myApp</code> , the created view is <code>myApp.myView</code> .
View Type	Type of view that is created: <i>XML</i> , <i>JSON</i> , <i>JS</i> , <i>HTML</i> , or <i>Template</i> .
Control ID	ID of the control that contains the view that is created by the target.
Bypassed Targets	One or more names of targets that are displayed when no route is matched.
View Level	Level (number) of the current view. The <code>View Level</code> is used to define the transition direction when navigating to this view.
Control Aggregation	Name of an aggregation of the <code>Control ID</code> (target control) that contains views. For example, a <code>NavContainer</code> has an aggregation called <code>Pages</code> and the shell container has <code>Content</code> .
Transition	Type of transition when navigating from the previous view to this view: <i>slide</i> , <i>flip</i> , <i>fade</i> , or <i>show</i> .
Target Parent	ID of the view that contains the control specified by the <code>Control ID</code> parameter.
Parent	Name of another target that will be displayed once this target is displayed.
Clear Aggregation	Whether the aggregation should be cleared before adding the view.

Routes

Add and configure routes for the application.

1. Click **+**.
2. Change the name, if required.

3. Define the route properties.
4. Use the up and down arrows to change the order of routes in the list.

Option	Description
Name	Name of the route.
Pattern	URL pattern that the route is matched against.
Greedy	Whether the route should be matched when another route is already matched.
Targets	Names of the targets that are displayed when the route is matched.

Manage Targets

Add and configure targets.

1. Click **+**.
2. Enter a name for the target and click *OK*.
3. Configure the route properties:
 - *View Name* - Required: Name of the view that is created by the target.
 - *View ID* - ID of the created view.
 - Define the other route properties as required.

7.1.4.5 Navigation Tab Options

Navigation between SAP Fiori launchpad applications is based on abstract representations (intents) that are resolved to concrete navigation targets. An intent is a mechanism that allows users to perform actions on semantic objects (such as navigating to a sales order or displaying a fact sheet).

For more information, see [Configuring Navigation](#).

i Note

The *Navigation* tab is displayed from version 1.2.0 of the `sap.app` namespace.

Inbound Table

Each row in the *Inbound* table contains an intent for cross navigation for inbound targets.

1. To add an intent, click **+**.
2. In the new row:

- Enter the semantic object as defined in an app launcher tile, for example, `SalesOrder`.
 - Enter the action to be performed on the object. The action can be a verb or a short phrase starting with lower case and without blanks. Examples are `display`, `create`, and `release`.
3. For each intent, select the row in the table, and then define its tile and inbound parameters.

Inbound Tile

Define the tile to be used for navigation for the selected intent.

Option	Description
Title	The title that you want to appear on the tile. The title must be in double curly brackets <code>{{ }}</code> .
Subtitle	The subtitle that you want to appear on the tile.
Icon	Select the icon to appear on the tile.
Data Source	The data source defined in the <code>sap . app</code> section that returns the information for an SAP Fiori launchpad dynamic tile.
Path	The relative path of the OData service that returns the information for the dynamic tile (relative to the data source path).
Refresh Interval	The time interval between data refreshes in the dynamic tile.

Inbound Parameters

Click **+** to add parameters, also known as signatures, for the selected intent.

For each parameter set the following options.

Option	Description
Name	A unique name for the parameter.
Value	A string value for the parameter. This value can be a verbatim default value or a reference to another parameter, according to the value format. For optional parameters, this is the default.
Value Format	The format of the parameter value: <ul style="list-style-type: none"> • <i>plain</i> - the parameter value is taken as a literal string value. • <i>reference</i> - the parameter value is a reference to another parameter.

Option	Description
Required	Whether this is a mandatory or optional parameter.
Filter Value	A string value for the filter. This value can be a verbatim filter value, a regular expression, or a reference, according to the filter format.
Filter Format	The format of the filter value: <ul style="list-style-type: none"> • <i>plain</i> - the filter value must match the actual parameter value. • <i>regex</i> - the filter value represents a regexp, which must be present in the parameter value. • <i>reference</i> - the filter value represents a reference to another parameter, for example, a <code>UserDefault</code> parameter. The resolved parameter value is then directly compared with the actual parameter value.

Additional Parameters

In the *Additional Parameters* field, choose how to handle additional parameters in the application that are not defined in the *Inbound Parameters* table:

- *ignored* - parameters are not passed on to the application.
- *allowed* - parameters are passed on to application.
- *not allowed* - the application will not run if there are additional parameters.

7.1.5 Developing Applications

Once you have created a project, you can use interactive features to write code from scratch, add a new component to the application, or add an extension to it. Run the application at any time to see how it is evolving.

[Working in the Code Editor \[page 130\]](#)

Use keyboard shortcuts and context menus to easily edit and navigate through your code and code comments.

[Using Code Completion \[page 137\]](#)

The code completion feature assists you when you are writing your code by preventing typos and other common mistakes, and providing API reference information for SAPUI5 objects.

[Checking Code \[page 155\]](#)

SAP Web IDE performs code checking, also known as validation, and displays errors as annotations.

[Developing Apps Using SAP Fiori Elements \[page 243\]](#)

This section contains information about creating SAP Fiori apps using SAP Fiori elements in SAP Web IDE.

[Using the Outline Pane for JavaScript Files \[page 250\]](#)

The Outline pane helps you to understand the structure of JavaScript files and to navigate through the code.

[Creating an HTML5 Application Descriptor File \[page 251\]](#)

Create an HTML5 Application Descriptor file in a project that has not been imported or created via the project wizards. This is required for defining cloud connectivity for external resources required by the application, and allows you to run the application properly in the SAP Web IDE.

7.1.5.1 Working in the Code Editor

Use keyboard shortcuts and context menus to easily edit and navigate through your code and code comments.

Feature support depends on file types.

All edit actions are also available from the *Edit* menu.

i Note

ES6 is not supported in JavaScript files.

i Note

The following keyboard shortcuts are for Microsoft Windows. For Mac OS, see [Keyboard Shortcuts \[page 85\]](#).

Basic Navigating and Editing

Action	Keyboard Shortcut	Context Menu
Move one tab to the right	<code>Alt</code> + <code>R</code>	-
Move one tab to the left	<code>Alt</code> + <code>Q</code>	-
Close all tabs to the right of the open tab		<i>Close Tabs to the Right</i>
Undo edit	<code>Ctrl</code> + <code>Z</code>	<i>Undo</i>
Redo edit	<code>Ctrl</code> + <code>Y</code>	<i>Redo</i>
Show and hide control characters, such as space, tab, newline, and paragraph	<code>Ctrl</code> + <code>I</code> (uppercase i)	-
Move a line up	<code>Alt</code> + <code>Up arrow</code>	-
Move a line down	<code>Alt</code> + <code>Down arrow</code>	-
Move to a specific line	<code>Ctrl</code> + <code>L</code>	<i>Go to Line</i>

Action	Keyboard Shortcut	Context Menu
Copy a line to the line above or below	<code>Alt</code> + <code>Shift</code> + <code>Up</code> arrow <code>Alt</code> + <code>Shift</code> + <code>Down</code> arrow	-
Expand the entire hierarchy of file elements	<code>Alt</code> + <code>Shift</code> + <code>F2</code>	Gutter context menu: <i>Expand All</i>
Collapse the entire hierarchy of file elements	<code>Ctrl</code> + <code>Alt</code> + <code>F2</code>	Gutter context menu: <i>Collapse All</i>
Search for a string	<code>Ctrl</code> + <code>F</code>	<i>Find</i>
Search and replace a string	<code>Ctrl</code> + <code>H</code>	<i>Find and Replace</i>

Commenting

Action	Keyboard Shortcut	Context Menu
For JavaScript and XML files Comment out a line and restore to code. Comment syntax is appended to the code fragment automatically as a line. Each comment line is prefixed with <code>//</code> . For example:	<code>Ctrl</code> + <code>/</code>	<i>Toggle Line Comment</i>
<pre>//Use the comment style for short comments.</pre>		
Comment out a block and restore to code. Comment syntax is appended to the code fragment as a block using <code>/*</code> and <code>*/</code> to wrap the comment block. For example:	<code>Ctrl</code> + <code>Shift</code> + <code>/</code>	<i>Toggle Block Comment</i>
<pre>/* This is a comment block. Use this block comment style when comments span multiple lines.*/</pre>		
<pre>//Use the comment style for short comments.</pre>		

Action	Keyboard Shortcut	Context Menu
Flag JavaScript code with a TODO comment. A // TODO comment is added at the cursor location. If the line already contains a // TODO comment, the action is ignored.	<code>Ctrl</code> + <code>Alt</code> + <code>T</code>	<i>Add TODO Comment</i>

Beautify File Formatting

Beautify file formatting for JavaScript, JSON, XML, and CSS files using the context menu *Beautify* option or `Ctrl` + `Alt` + `B`.

- The XML beautifier formats code with line wrapping at 140 characters, and an indentation of one tab space. The beautifier is not configurable.
- The CSS beautifier formats indentation with one tab space and is not configurable.
- You can customize JavaScript beautifier settings for your project. For more information, see [Customize JavaScript Beautifier Properties \[page 110\]](#).

Using Multiple Cursors

You can use multiple cursors to rename several variables at once or to insert the same text in multiple locations.

To add cursors in your file, press `Ctrl` and click at the required locations. You can then type text, which will appear in all the cursor locations.

To remove multiple cursors click anywhere in the file.

Refactoring

You can change the name of a JavaScript function or variable, by using the context menu *Refactor* or `Alt` + `J`. Enter a valid new name and click *Rename*, and all references to the function or variable are automatically updated.

Viewing Git Blame

For files committed to Git, you can view who was the last developer to change each line, the commit that included the change, and the date of the change.

For more information, see [Git Blame \[page 388\]](#).

Navigating from View to Controller

While editing an XML view, you can navigate to the view's controller by using the context menu *Open Controller*. The controller JavaScript file is opened in a new tab.

If you right-click the name of an event handler and select *Open Controller*, the controller is opened to the event handler function (if it is defined in the controller).

Managing i18n Strings

In the context menu of an XML or JavaScript file, select *Open i18n* to open the i18n file defined in the `manifest.json`.

You can easily create a new string without having to open the i18n file by selecting *Create i18n String*. Add the string and its key, and they are added to the i18n file.


To edit a string, right-click on the i18n binding (for example, `{i18n>fileType_title}`) and select *Edit i18n String*.

You can get code completion when you need to select a string to add to an XML or JavaScript file. Put your cursor inside quotation marks and press `Ctrl` + `Space`, and get a list of strings defined in your i18n properties files. The properties files must be defined as i18n models in your manifest.

7.1.5.1.1 Configure the Code Editor

Define the appearance and behavior of the code editor, and whether to autosave all changes in SAP Web IDE.

Procedure

1. In the left sidebar, choose  (Preferences), then choose *Code Editor*.
2. Customize the appearance, input behavior, or save options.

Appearance

Option	Description
Code Editor Theme	The theme of the code editor determines the background color and color of text.
Font	The monospace font for text in the code editor
Font Size	The font size of text in the code editor.
Code Folding	Hides code from a marked begin point to a marked end point or to the end of the file if end points are not used.

Option	Description
	Click in the gutter to mark begin and end points.
Full line selection	Line selection extends to the end of the line instead of to the end of the text in the line.
Highlight selected word	Highlights all occurrences of the word at the cursor position.
Show invisible characters	Shows white-space characters such as spaces, tabs, and new lines.

Input Behavior

Option	Description
Auto-pair characters	Auto-pairs characters, such as quotation marks, parentheses, brackets, and so on.
Use spaces for tab indentation	Uses spaces for tab indentation, also known as soft tabs.
New Line Mode	Lets you choose one of the following: <ul style="list-style-type: none"> ○ auto ○ windows ○ unix

Save Options

Option	Description
Automatically save changes in all open documents at preset intervals.	For more information, see Setting User Preferences [page 70] .
Automatically beautify the code of an active document on manual save.	For more information about beautifying JavaScript code, see Customize JavaScript Beautifier Properties [page 110] .

Test your settings in the preview area on the right side of the screen..

3. Change the [Autosave](#) setting, if required.
4. Click [Save](#).

7.1.5.1.2 Generate JSDoc Comment Snippets

You can generate a snippet for JavaScript function declaration that creates a template for documenting the function.

Context

The JSDoc comment snippet provides a template for documenting the function that you create. You can use code completion for variables within the snippet.

Procedure

1. Select a function, and press `Ctrl` + `Alt` + `J`, or choose *Generate JSDoc Comment* from the context menu.

The snippet is displayed above the function code.

2. Enter information about the function variables in the corresponding placeholders in the comment.

7.1.5.1.3 Locate Objects in Code

The code editor allows you to locate objects or definitions of objects in code.

Context

Methods can be defined in other modules on which the source file depends. You can use *Goto Definition* to navigate to methods declared explicitly in the dependency file and generated methods for metadata properties, associations, aggregations, and events of SAPUI5 controls.

The dependencies can be defined using a dependency declaration in either of the following ways:

- Use `jQuery.sap.require(<moduleId1>, <moduleId2>, ...)`. The dependency (target) file should contain the module declaration `jQuery.sap.declare(<moduleId>)`.
- Use `sap.ui.define([<dependencyFile1>, <dependencyFile2>, ...], function(d1, d2, ...){ });` where `dependencyFile` is the relative file path in the current project or the logical path to the library module. The dependency (target) files should contain a module declaration using `sap.ui.define` or Asynchronous Module Definition (AMD).

Suggestions can be provided only if the dependency file is visible in the current project.

You can:

- Use goto services to locate the definition action of a user-defined object (for example, a variable, function, object, or property) in JavaScript files for in a project. Goto searches the active file as well as all files in the same project.

i Note

Goto is not supported for JavaScript native keywords (for example, `var`, `window`, `JSON`, or SAPUI5 library objects (for example, `sap`, `sap.ui`, `sap.ui.core`), as they are not user-defined objects. If you attempt to use goto with a restricted object, a message appears in the console pane to indicate that the definition was not found.

- Highlight all instances of the selected object in the active file.

Procedure

1. To locate the definition using goto services:
 - a. Validate whether the object is user-defined by placing the cursor over the user-defined object and pressing **CTRL** + **ALT**.
If the object has a user-defined definition, it appears underlined in blue. Otherwise, the console reports that the definition cannot be found.
 - b. To go to the location, press **CTRL** + **ALT** and click the underlined object.
If the definition is in the current file, the definition is highlighted. Otherwise, the correct file is first opened and the definition is highlighted.

Note

You can also conflate both of these steps into a single action by selecting the object and pressing **CTRL** + **ALT** + **G**, or selecting the object and choosing **Edit > JavaScript > Goto Definition**.

2. To locate all instances of an object in an open file, double-click the object.
All instances of the object are highlighted in a blue box.
For example, if you select the first instance of the string `content`, all remaining instances are identified in a blue box.



```
index.html x
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta http-equiv="X-UA-Compatible" content="IE=edge">
5
6 <script src="resources/sap-ui-core.js"
7     id="sap-ui-bootstrap"
8     data-sap-ui-libs="sap.ui.commons"
9     data-sap-ui-theme="sap_bluecrystal">
10 </script>
11 <!-- add sap.ui.table, sap.ui.ux3 and/or other libraries to 'data-sap-ui-libs' if required -->
12
13 <script>
14     sap.ui.localResources("view");
15     var view = sap.ui.view({id: "dfsd", viewName: "view.dfsd", type: sap.ui.core.mvc.ViewType.XML});
16     view.placeAt("content");
17 </script>
18
19 </head>
20 <body class="sapUiBody" role="application">
21 <div id="content"></div>
22 </body>
23 </html>
```


7.1.5.2 Using Code Completion

The code completion feature assists you when you are writing your code by preventing typos and other common mistakes, and providing API reference information for SAPUI5 objects.

Context

Code completion is triggered in one of the following ways:

- Auto hint code completion for JavaScript and XML files
Based on your cursor location, SAP Web IDE establishes the context and displays a constrained list of suggestions. Use the corresponding icon to visually identify the code type being completed (that is, XML or JavaScript). This option is configurable; see [Enabling Code Completion \[page 139\]](#).
- Manual code completion
Supported for various file types including: `i18n` and `messagebundle` property files.
 - `i18n` and `messagebundle`
 - JavaScript
 - XMLNo configuration is required.

Cross-file completion for JavaScript

The dependencies can be defined using a dependency declaration in either of the following ways: SAP Web IDE can provide suggestions for functions (methods) that are defined in other modules on which the source file depends. Suggestions include methods declared explicitly in the dependency file and generated methods for metadata properties, associations, aggregations, and events of SAPUI5 controls.

- Use SAP Web IDE can provide suggestions for functions (methods) that are defined in `jQuery.sap.require(<moduleId1>, <moduleId2>, ...)`. The dependency (target) file should contain the module declaration `jQuery.sap.declare(<moduleId>)`.
- Use `sap.ui.define([<dependencyFile1>, <dependencyFile2>, ...], function(d1, d2, ...) { });` where `dependencyFile` is the relative file path in the current project or the logical path to the library module. The dependency (target) files should contain a module declaration using `sap.ui.define` or Asynchronous Module Definition (AMD).

Note

Suggestions can be provided only if the dependency file is visible in the current project

Code completion for SAPUI5

The code editor provides code completion suggestions for:

- Properties in SAPUI5 methods. For example, suggestions for `Opa5.createPageObjects`, include all the properties for `baseClass`, `namespace`, `actions`, and `assertions`.
- SAPUI5 static and instance methods when you use dependency declarations with Asynchronous Module Definition (AMD).
- SAPUI5 module dependencies in AMD. Proposed suggestions are for the element type of module according to the element name entered.

API reference information for SAPUI5 objects is shown in a tooltip when you hover over a suggestion. If more information is available, a [More Details](#) button is provided that links to the full SAPUI5 documentation for the current element.

Code completion for SAPUI5 is dependent on the SAPUI5 version selected in the project settings.

Procedure

To use auto hint code completion:

1. Create or open an XML or JavaScript file.
2. Place your cursor in the context for which you require assistance, and press `Ctrl` + `Space`.

A list of suggestions is displayed. Deprecated APIs are indicated with strikethrough text. Hover over a suggestion and the tooltip also indicates whether an API is deprecated or experimental.

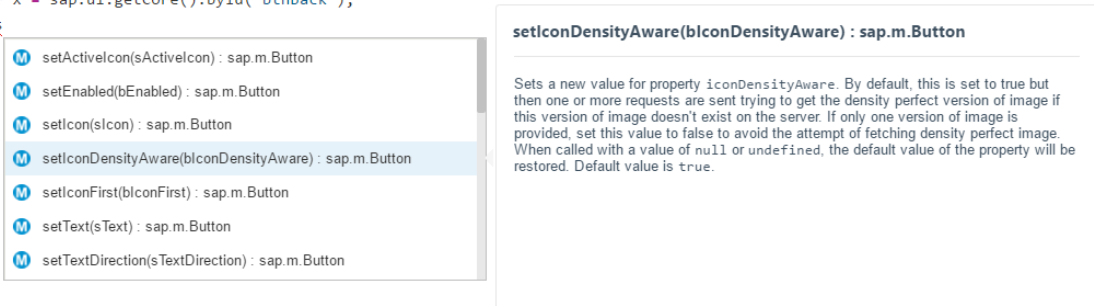
3. Scroll through the list, select the appropriate fragment, and press `Enter`.
The code fragment is added at the cursor location.

Example

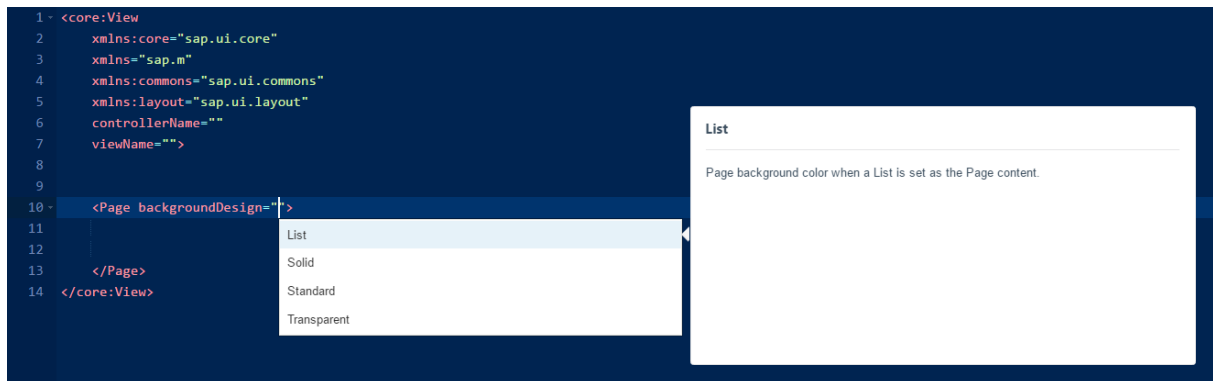
This example of embedded type code completion shows how you can attach a type to your variable definition to enable presentation of appropriate code completion suggestions for the variable.

Define the variable type in a JSDoc comment before the variable definition, and then press `Ctrl` + `Space` for code assistance. When you hover over a suggestion (e.g., `setIconDensityAware`), API reference information is shown in a tooltip.

```
formatConcatenateNameId : function(sName, sID) {  
    /** @type sap.m.Button */  
    var x = sap.ui.getCore().byId("btnBack");  
    x.  
}
```




In the next example, to determine which values you can use for `<Page backgroundDesign="">`, move the cursor between the quotation marks as shown below, and launch code completion. The options for the attribute are listed. Hover over one of the options to get a tooltip with API reference information about that option.



7.1.5.2.1 Configure Code Completion

You can enable code completion as you type (auto hint) for JavaScript and XML files.

Procedure

1. To open the *Preferences* perspective, in the left sidebar, choose  (Preferences).
2. Click *Code Completion* and select the *Enable* checkbox.
3. Click *Save*.

7.1.5.2.2 Code Completion for Property Files

You can insert code snippets into your `i18n` and `messagebundle` property files enabling you to easily insert UI elements (for example, button names, table titles) into your code. The elements are highlighted in different colors as you type.

This feature enables you to::

- Press `Ctrl` + `space` to choose the text type from a dropdown list.
- Type in the comments, keys, and values - as you type each element is highlighted in a different color.
- Use the `Tab` key to quickly navigate between the highlighted elements.

7.1.5.2.3 Code Completion for OPA and QUnit Tests

Use code completion for snippets when developing OPA and QUnit tests in SAP Web IDE.

OPA Tests

When you develop OPA tests in an OPA page, SAP Web IDE can provide snippets for OPA actions and OPA assertions. You can also use code completion to add the OPA tests that you create to an OPA journey.

At the relevant location in the file, enter `opa` `Ctrl` + `Space` and choose the relevant snippet from the list of suggestions. The code is inserted and you can then add the required definitions in the placeholders.

For more information about OPA testing, see [One Page Acceptance Tests \(OPA5\)](#)

QUnit Tests

When you develop QUnit tests, you can use code completion for QUnit modules and QUnit tests.

At the relevant location in the file, enter `qunit` `Ctrl` + `Space` and choose the relevant snippet from the list of suggestions. The code is inserted and you can then add the required definitions in the placeholders.

For more information about QUnit testing, see [QUnit Testing Fundamentals](#)

7.1.5.2.4 Try It: JavaScript Code Completion Basics

A self-guided example that demonstrates how to use JavaScript code completion services for SAPUI5 applications.

Prerequisites

Open a new JavaScript file in a new or existing project.

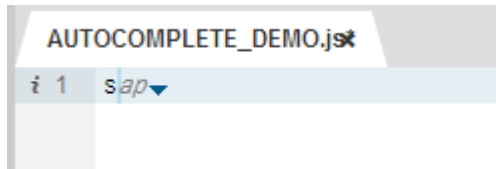
Context

This example demonstrates the following code completion features:

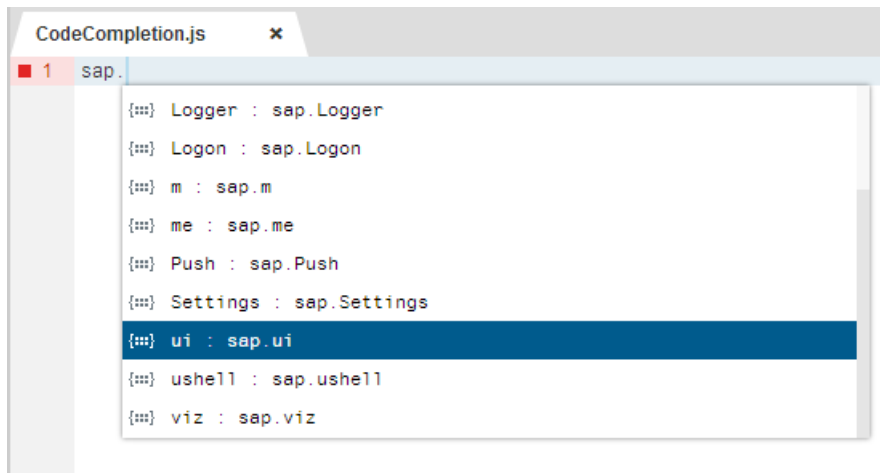
- JavaScript code completion
- Element hints
- Snippet insertion

Procedure

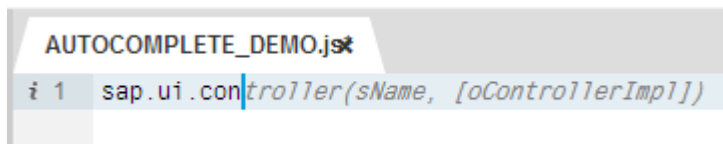
1. In the JavaScript editor, type `s`, and notice that the autocomplete suggestions automatically appear adjacent to the letter in context.



2. When you see `sap`, press `Enter`.
3. Continue the string and type `.`, then press `Ctrl` + `Space` to list all the namespaces for `sap`.
4. In the popup, use the arrow keys to select `ui : sap.ui`, and press `Enter`.



5. To append the `controller` function, type `cont`. The suggestion reduces to the correct one. Press `Enter` to append it in this location.



6. Add the `onInit` function in row 2, using the techniques demonstrated in previous steps.

```

AUTOCOMPLETE_DEMO.js*
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2   onInit: function() {
3
4   }

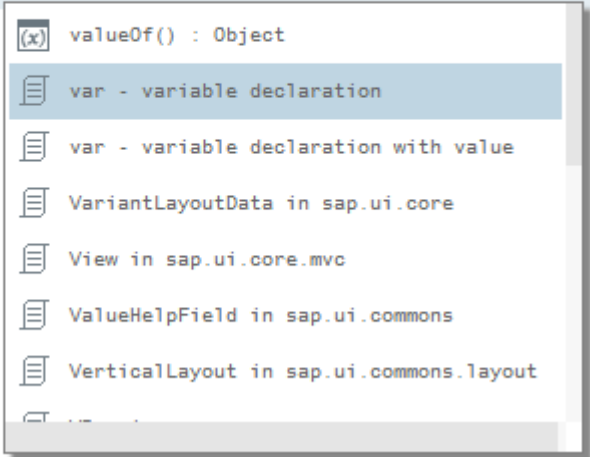
```

- To inject a code snippet, create an empty line after row 2, start typing `v`, then press `Ctrl` + `Space` to see the list of possible code fragments.

```

AUTOCOMPLETE_DEMO.js*
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2   onInit: function() {
3     v
4   }

```



- Use the arrow keys to find `var -variable declaration`, and press `Enter`. The correct syntax is used for this `var` declaration. You must still choose a name.

```

CodeCompletion.js* x
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2   onInit: function() {
3     var name;
4   }
5 });

```

- Name the `var` as `fc1 = form`, then press `Ctrl` + `Space` to retrieve a list of form templates for it.

```

AUTOCOMPLETE_DEMO.js*
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2   onInit: function() {
3     var fc1 = form
4   }

```

10. Use the arrow keys to choose `FormContainer in sap.ui.layout.form`, then press `Enter`.
The form and all its properties and values are added correctly to the file.

```

CodeCompletion.js* x
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2   onInit: function() {
3     var fc1 = new sap.ui.layout.form.FormContainer({
4       id: "id", // sap.ui.core.ID
5       expanded: true, // boolean
6       expandable: false, // boolean
7       visible: true, // boolean
8       tooltip: undefined, // sap.ui.core.TooltipBase
9       customData: [], // sap.ui.core.CustomData
10      formElements: [], // sap.ui.layout.form.FormElement
11      title: undefined // sap.ui.core.Title
12    });
13   };
14 }
15 });

```

7.1.5.2.5 Try It: JavaScript Code Completion with a User-Defined Object

A self-guided example that demonstrates how to use code completion with an object that you define. Create one file to define the object. Create a second file in which to use code completion with the object that you defined in the first file.

Context

SAP Web IDE code completion automatically displays user-defined objects in the same project.

Procedure

1. In a new or existing project, create a new JavaScript file.
2. Add `jQuery.sap.declare(FileName)` to the start of all source files; otherwise, objects will not appear as a hint.
3. Create an object called `obj1`, by replicating these lines in this empty file and saving the changes:

```
cross-file1.js  x
1  jQuery.sap.declare("cross-file1");
2
3  var obj1 = {a:0, b:"hello"};
4
5  function fun2(x) {
6    return x + 1;
7  }
```

4. In the same project, create another new JavaScript file.
5. Add another statement to the start of this file: `jQuery.sap.require(FileName)`. In this example, use `jQuery.sap.require(cross-file1)`.
6. Enter a new line and choose the object for which you want you want to use code completion:
 - o Either type `var n = obj1`, then press `Ctrl` + `Space` to get this hint from the first file:


```

cross-file1.js  x  cross-file2.js  x
1  jQuery.sap.require("cross-file1");
2
3  var n = obj1.
    (x) a : Number
    (x) b : String
    -----
    (x) hasOwnProperty(property) : Boolean
    (x) isPrototypeOf(object) : Boolean
    (x) propertyIsEnumerable(property) : Boolean
    (x) toLocaleString() : String
    (x) toString() : String
    (x) valueOf() : Object
    (x) prototype : Object
  
```

- Or type `var f = fun`, then press `Ctrl` + `Space` to get this hint from the first file:

```

cross-file1.js  x  *cross-file2.js  x
1  jQuery.sap.require("cross-file1");
2
3  var f = fun
    (x) fun2(x) : Number
    (x) Function() : Function
    (x) function - function declaration
    (x) function
  
```

- Select the first result and press `Enter`.

Results

The code updates accordingly in the second file.

7.1.5.2.6 Try It: XML Code Completion

A self-guided example that demonstrates how to use XML code completion services.

Prerequisites

Open a new XML file in a new or existing project.

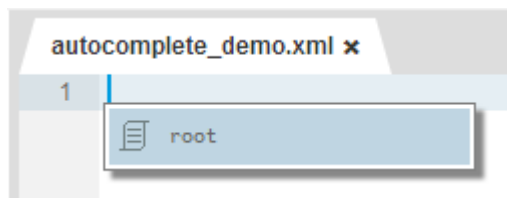
Context

This example demonstrates the following code completion features:

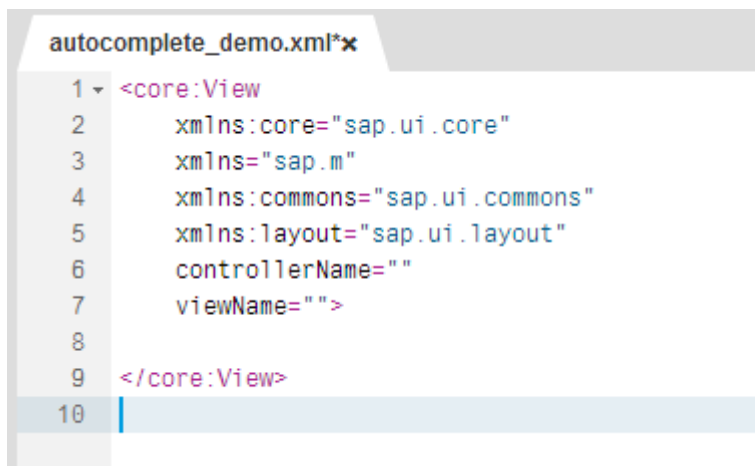
- XML node completion
- Element hints

Procedure

1. Press `Ctrl` + `Space` and select the `root` node suggestion.



The `root` node is automatically populated with required elements and values.



2. In row 8, type `<p`.

```

autocomplete_demo.xml*x
1 <core:View
2   xmlns:core="sap.ui.core"
3   xmlns="sap.m"
4   xmlns:commons="sap.ui.commons"
5   xmlns:layout="sap.ui.layout"
6   controllerName=""
7   viewName="">
8 <panel
9 </core:View>

```

3. Press the arrow keys to scroll through options. When you see `Page`, press `Enter`.
4. Press `Space`, then press `Ctrl` + `Space`.

A new list of hints appears, based on context. Suggestions can include XML syntax, property, and event proposals.


```


autocomplete_demo.xml*x
1 <core:View
2   xmlns:core="sap.ui.core"
3   xmlns="sap.m"
4   xmlns:commons="sap.ui.commons"
5   xmlns:layout="sap.ui.layout"
6   controllerName=""
7   viewName="">
8 <Page
9 </core:View
10


```


<> />


<> >


 backgroundDesign

 busy

 busyIndicatorDelay

 enableScrolling

 formatError

 icon

5. Select `backgroundDesign` and press `Enter`.
6. Define a value by typing `=` and pressing `Ctrl` + `Space`.
7. Select `List` and press `Enter`.
The syntax completes correctly.

```

7     viewName="">
8     <Page backgroundDesign="List"></Page>
9 </core:View>
10

```

7.1.5.2.7 Try It: XML Code Snippet Insertion

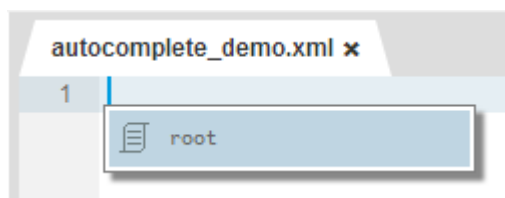
A self-guided example that demonstrates how to insert an XML code snippet.

Prerequisites

Open a new XML file in a new or existing project.

Procedure

1. Press `Ctrl` + `Space` and select the `root` node suggestion.



The `root` node is automatically populated with required elements and values.


```

autocomplete_demo.xml*x
1 <core:View
2     xmlns:core="sap.ui.core"
3     xmlns="sap.m"
4     xmlns:commons="sap.ui.commons"
5     xmlns:layout="sap.ui.layout"
6     controllerName=""
7     viewName="">
8
9 </core:View>
10











```

2. In the root node, type `<p`.

```
autocomplete_demo.xml*x
1 <core:View
2   xmlns:core="sap.ui.core"
3   xmlns="sap.m"
4   xmlns:commons="sap.ui.commons"
5   xmlns:layout="sap.ui.layout"
6   controllerName=""
7   viewName="">
8 <p|panel
9 </core:View>
```

3. Press the arrow keys to scroll through options. Select Page with the snippet icon ( Page), and press `Enter`.

```
autocomplete_demo.xml*x
1 <core:View
2   xmlns:core="sap.ui.core"
3   xmlns="sap.m"
4   xmlns:commons="sap.ui.commons"
5   xmlns:layout="sap.ui.layout"
6   controllerName=""
7   viewName="">
8 <p|
9 </core:
10
```

-  Page
-  Page
-  Panel
-  Panel
-  Popover
-  Popover
-  ProgressIndicator
-  ProgressIndicator
-  PullToRefresh
-  PullToRefresh

SAP Web IDE inserts a full snippet of code at the desired location:

```
8 <Page xmlns="sap.m"
9     id="id"
10    title=""
11    showNavButton="false"
12    showHeader="true"
13    navButtonText=""
14    enableScrolling="true"
15    icon=""
16    backgroundDesign="Standard"
17    navButtonType="Back"
18    showFooter="true"
19    navButtonTap=""
20    navButtonPress="">
21        <tooltip></tooltip> <!-- sap.ui.core.TooltipBase -->
22        <content></content> <!-- sap.ui.core.Control -->
23        <customHeader></customHeader> <!-- sap.m.Bar -->
24        <footer></footer> <!-- sap.m.Bar -->
25        <subHeader></subHeader> <!-- sap.m.Bar -->
26        <headerContent></headerContent> <!-- sap.ui.core.Control -->
27    </Page>
```

7.1.5.2.8 Try It: XML Metadata Completion from Schema Files

Display autocomplete metadata information from schema files. Metadata suggestions are context-aware. You can get suggestions for elements, attributes, attribute type, properties, and property values.

Prerequisites

Metadata and schema files must exist. For the purposes of this guided example, create two schema files:

Schema demo2a.xsd Includes these metadata definitions:

```
<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <xs:any minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute/>
  </xs:complexType>
</xs:element>
<xs:element name="employees">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="employee" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Schema demo2b.xsd Includes these definitions:

```

<xs:attributeGroup name="customattrgroup">
  <xs:attribute name="inheritable" type="xs:boolean"/>
  <xs:attribute name="birthday" type="xs:date"/>
</xs:attributeGroup>
<xs:group name="infogroup">
  <xs:sequence>
    <xs:element name="address" type="xs:string"/>
    <xs:element name="phone" type="xs:string"/>
  </xs:sequence>
</xs:group>
<xs:element name="customer">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="customername" type="xs:string"
        maxOccurs="unbounded"/>
      <xs:group ref="infogroup"/>
    </xs:sequence>
    <xs:attribute ref="gender"/>
    <xs:attribute name="mode" default="interleave" use="optional">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="none"/>
          <xs:enumeration value="interleave"/>
          <xs:enumeration value="suffix"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attributeGroup ref="customattrgroup"/>
  </xs:complexType>
</xs:element>

```

Procedure

1. In a new or existing project, create a new XML file, and specify `demo2a.xsd` and `demo2b.xsd` as the metadata schema instances.

For example, if the XSD file uses a namespace, define the same namespace in the XML file as shown; otherwise, autocompletion for metadata fails.

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:SchemaLocation="http://<domain> demo2a.xsd
http://<domain> demo2b.xsd"

```

→ Tip

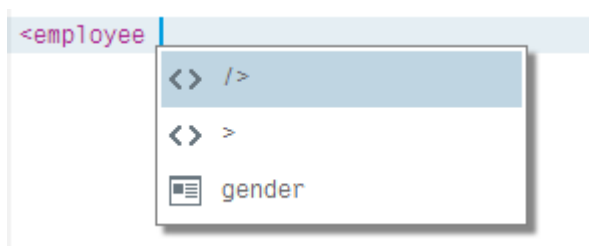
The XSD files that you define can be a relative path or a full URL.

However, if no namespace is used, then define the schema as:

```
xsi:noNamespaceSchemaLocation="demo2a.xsd demo2b.xsd"
```

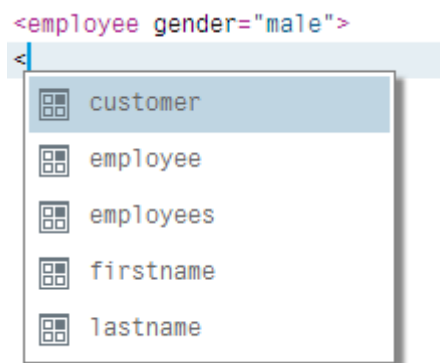
Autocompletion for metadata is ready to use from the new XML file.

2. In the next row, type `<e`, and press `Enter` to accept the `employees` node.
3. In another row, press `Ctrl` + `Space` to determine what lists of elements you can define. Select `employees`, and press `Enter`.
4. Continue this definition by pressing `Space` after `<employee` and pressing `Ctrl` + `Space` to see the list of suggestions for this context.



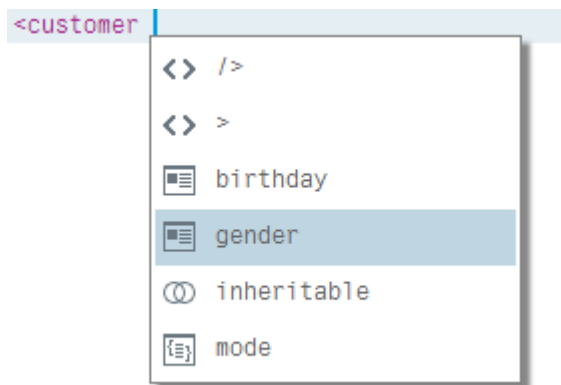
Because `<xs:anyAttribute/>` is defined for the `employee` element, the `gender` attribute from `demo2b.xsd` appears in this list.

5. Select `gender`, and press `Enter` to complete the syntax automatically.
6. In the next row type `<` and press `Ctrl` + `Space` to see the list of children for this element.



These suggestions appear for `employee`, because `<xs:any minOccurs='0' />` is defined for element `employee`, so more elements beyond `firstname` and `lastname` appear, including `customer` from `demo2b.xsd`.

7. Select `customer`, and press `Enter`.
8. Press `Space` and press `Ctrl` + `Space` to see the list of suggestions for this context.

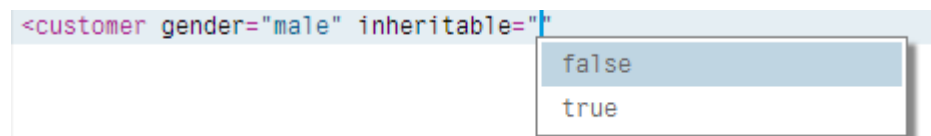


Note

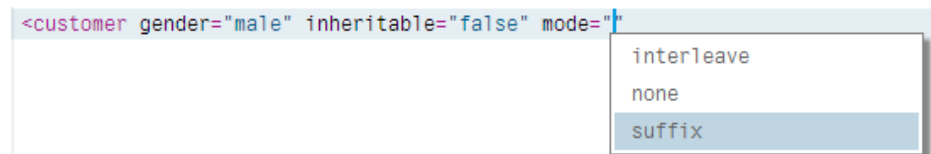
Note the icons in this example: the `mode` attribute's icon shows an enum type, and the `inheritable` attribute's icon shows a boolean type.

9. Select `gender`, and press `Enter` to complete the syntax automatically.
10. To display a list of values for a specific attribute type, move the cursor to the value's location, and press `Ctrl` + `Space`.

Boolean example



Enum example



7.1.5.2.9 Supported SAPUI5 Libraries

Code completion is supported for many SAPUI5 libraries.

Code completion is supported for the following SAPUI5 libraries:

- `sap.ca.ui`
- `sap.m`
- `sap.me`
- `sap.ui.commons`
- `sap.ui.comp`
- `sap.ui.core`
- `sap.ui.layout`

- `sap.ui.richtexteditor`
- `sap.ui.table`
- `sap.ui.unified`
- `sap.ui.ux3`
- `sap.ushell`
- `sap.uxapp`
- `sap.viz`

7.1.5.3 Checking Code

SAP Web IDE performs code checking, also known as validation, and displays errors as annotations.

By default, when you open a JavaScript, JSON, or XML file, code checking is triggered and the detected code issues are displayed as annotations within the editor. Understand how to configure and use code checking in your project.

[Configure Code Checking \[page 156\]](#)

Configure when to trigger code checking, also known as code validation, and the level of messages to display.

[Code Checking Annotations \[page 157\]](#)

If a syntax error is found during editing, the relevant lines are annotated with flags indicating the error severity. Understand the severity level of these flags so that when you open a file, you know how to interpret these annotations.

[JavaScript Validation \[page 159\]](#)

Review the default JavaScript validator configuration. Customizations always override these defaults.

[XML Validation \[page 213\]](#)

You can configure which XML validator to use in your project.

[Validation of neo-app.json Files \[page 239\]](#)

A project's `neo-app.json` file is validated on opening in the code editor.

[Validation of manifest.json Files \[page 239\]](#)

Application descriptor files (`manifest.json`) for SAP Fiori projects are validated for syntax and schema issues.

[Validation of Stable IDs \[page 240\]](#)

Check that stable IDs exist in your `manifest.json` file and your `view.xml` file.

[Validation of i18n Files \[page 241\]](#)

Your i18n files are validated in the SAP Web IDE code editor according to the SAP Fiori best practices.

[Using the Problems View \[page 242\]](#)

View information about problems in the projects in your workspace.

7.1.5.3.1 Configure Code Checking

Configure when to trigger code checking, also known as code validation, and the level of messages to display.

Context


By default, code checking is enabled when you make changes to your code, and all messages are displayed. You can change these defaults.

The code checking level is also applied to the Problems view.

i Note

You can customize code checking rules for each project. For more information about customizing and using code checking, see [Checking Code \[page 155\]](#).

Procedure

1. To open the *Preferences* perspective, in the left sidebar, choose  (Preferences).
2. Select *Code Check*.
3. In the *Run Code Check* section, select when to display code checking annotations:
 - Choose *On Save* to display annotations only when you save your file.
 - Choose *On Change* to display annotations every time you make a change to your code.
4. In the *Code Check Level* section, select which messages to display:
 - *All* displays all errors, warnings, and information messages
 - *Error* displays error messages only
 - *Error and Warning* displays error and warning messages
 - *Disable* suppresses message display
5. Choose *Save*.

Task overview: [Checking Code \[page 155\]](#)

Related Information

[Code Checking Annotations \[page 157\]](#)

[JavaScript Validation \[page 159\]](#)

[XML Validation \[page 213\]](#)

[Validation of neo-app.json Files \[page 239\]](#)

[Validation of manifest.json Files \[page 239\]](#)

[Validation of Stable IDs \[page 240\]](#)

[Validation of i18n Files \[page 241\]](#)

[Using the Problems View \[page 242\]](#)

7.1.5.3.2 Code Checking Annotations

If a syntax error is found during editing, the relevant lines are annotated with flags indicating the error severity. Understand the severity level of these flags so that when you open a file, you know how to interpret these annotations.

- All syntax errors are annotated on code line and tab levels and each annotation is colored according to its severity.

Color	Severity
Red	Error
Yellow	Warning
Blue	Information

- When you hover over an annotation, a tooltip displays one or more detected issues and possible resolutions for the annotated line. The detected issues of an annotated line are categorized (for easy identification), identified, and described, so that you can determine how best to resolve an issue.

Categories Used to classify the issue. For example, possible error, best practice, stylistic issue, and others.

Rule IDs Used to define the logic for detection or list known issue exceptions. For example, *semi* is a rule ID for ESLint.

Messages Detail the issue or suggest a possible resolution.

You can resolve the issue and continue development with iterative fixes.

Parent topic: [Checking Code \[page 155\]](#)

Related Information

[Configure Code Checking \[page 156\]](#)

[JavaScript Validation \[page 159\]](#)

[XML Validation \[page 213\]](#)

[Validation of neo-app.json Files \[page 239\]](#)

[Validation of manifest.json Files \[page 239\]](#)

[Validation of Stable IDs \[page 240\]](#)

[Validation of i18n Files \[page 241\]](#)

[Using the Problems View \[page 242\]](#)

7.1.5.3.2.1 Try It: Code Correction Using the Default JavaScript Validator

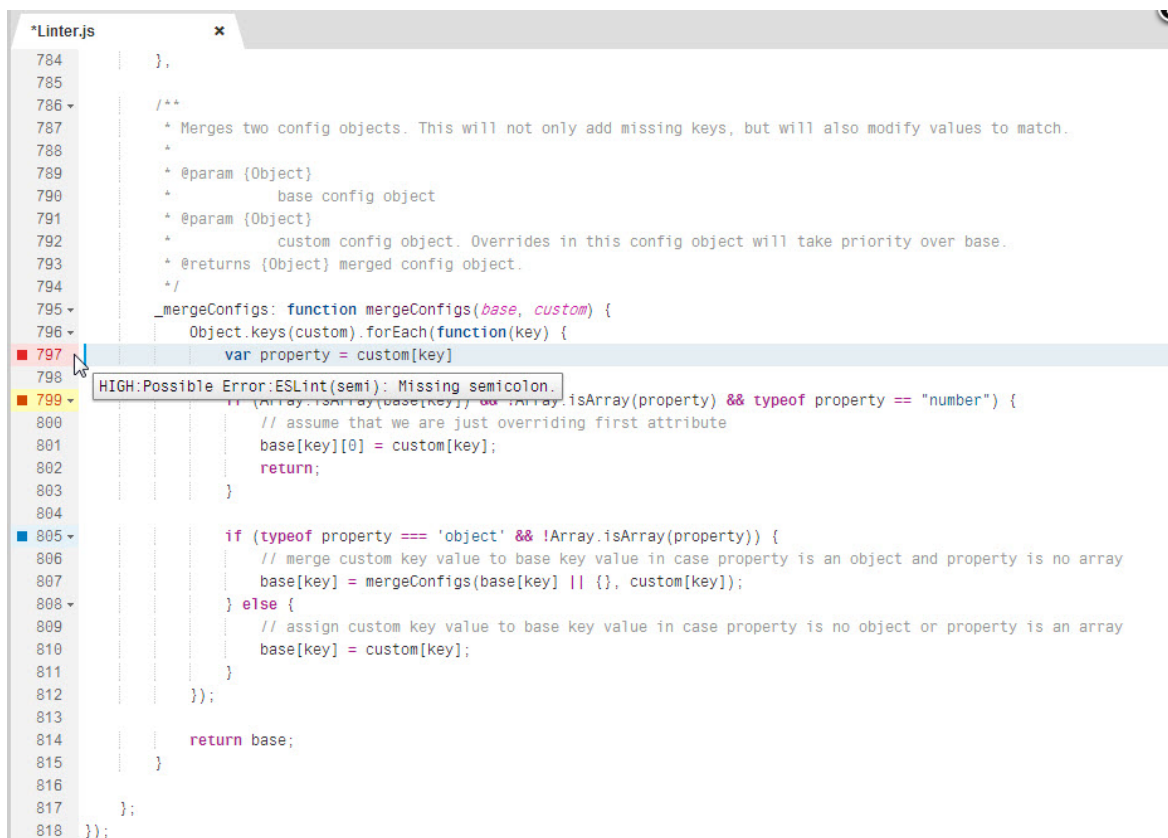
A self-guided example that demonstrates how to use the default JavaScript validator on an open JavaScript file.

Prerequisites

Configure validation to run on change as described in [Configure Code Checking \[page 156\]](#). Then follow the steps in [Opening and Reading Files with Annotations \[page 157\]](#) to open a file named `Linter.js`.

Procedure

1. Once the `Linter.js` file is open, use the colored square flags to determine which potential issues are the most serious.



The screenshot shows a code editor window titled `*Linter.js`. The code is a JavaScript function `mergeConfigs` that takes two objects, `base` and `custom`, and returns a merged object. The function iterates over the keys of the `custom` object and merges their values into the `base` object. A tooltip is visible over line 799, indicating a 'HIGH: Possible Error: ESLint(semi): Missing semicolon.' The tooltip points to the end of the line: `return;`. The code is as follows:

```
784     },
785
786     /++
787     * Merges two config objects. This will not only add missing keys, but will also modify values to match.
788     *
789     * @param {Object}
790     *     base config object
791     * @param {Object}
792     *     custom config object. Overrides in this config object will take priority over base.
793     * @returns {Object} merged config object.
794     */
795     _mergeConfigs: function mergeConfigs(base, custom) {
796         Object.keys(custom).forEach(function(key) {
797             var property = custom[key]
798
799             if (Array.isArray(base[key]) || !Array.isArray(property) && typeof property == "number") {
800                 // assume that we are just overriding first attribute
801                 base[key][0] = custom[key];
802                 return;
803             }
804
805             if (typeof property === 'object' && !Array.isArray(property)) {
806                 // merge custom key value to base key value in case property is an object and property is no array
807                 base[key] = mergeConfigs(base[key] || {}, custom[key]);
808             } else {
809                 // assign custom key value to base key value in case property is no object or property is an array
810                 base[key] = custom[key];
811             }
812         });
813
814         return base;
815     }
816
817 };
818 });
```

2. Look at the image and use the tooltips to explore the most critical error. In this case, the error on row 799 is a fatal error.

```

9 - if (Array.isArray(base[key]) && !Array.isArray(property) && typeof property === "number") {
    // assume that we are just overriding first attribute
1  FATAL: Syntax Error: ESLint(syntax-parse): Unexpected token {

```

3. Read the hover text and see that an unexpected token is a solution to the problem. Correct this token and see that the flag disappears.
4. Correct the issue on row 797 by repeating the hover process.
5. Insert a code fragment that transgresses a well-known rule. For example, JavaScript allows you to write a condition statement like this:

```
if (condition) statement;
```

However, the validator flags this because the statement is not blocked. Note that real-time errors are emphasized, meaning that all other flagged issues temporarily disappear. This allows your attention to be focused on the current problem and fix it instantaneously. This is a key feature of the validating on change.

6. Insert the block by placing braces around the statement. Note that when you correct the syntax error, all other detected issues reappear.
7. Save and exit the file.

7.1.5.3.3 JavaScript Validation

Review the default JavaScript validator configuration. Customizations always override these defaults.

The SAP Web IDE default JavaScript validator uses ESLint code checking.

ESLint Rule Execution Defaults	<p>See these resources for additional support:</p> <ul style="list-style-type: none"> • For ESLint configuration details, see Configuring ESLint . • For ESLint rule information, see Rules .
ESLint Rule Metadata Defaults	<p>Attributes can have multiple supported values. Use the following:</p> <ul style="list-style-type: none"> • <code>severity</code> attribute to define whether an issue renders as <code>error</code>, <code>warning</code> (default), or <code>information</code>. • <code>category</code> attribute for a better semantic classification: <code>possible error</code>, <code>best practice</code>, <code>stylistic issue</code>, and others. • <code>help</code> attribute (optional) to override the default help links listed.

Parent topic: [Checking Code \[page 155\]](#)

Related Information

- [Configure Code Checking \[page 156\]](#)
- [Code Checking Annotations \[page 157\]](#)
- [XML Validation \[page 213\]](#)
- [Validation of neo-app.json Files \[page 239\]](#)
- [Validation of manifest.json Files \[page 239\]](#)
- [Validation of Stable IDs \[page 240\]](#)
- [Validation of i18n Files \[page 241\]](#)
- [Using the Problems View \[page 242\]](#)

7.1.5.3.3.1 Create JavaScript Code Checking Rules

You can replace the SAP Web IDE default rule configurations in the JavaScript validator by configuring your own code checking rule configurations.

Context

The SAP Web IDE default JavaScript validator uses ESLint code checking. You can override the default rule configuration by creating a folder containing your custom rules.

Procedure

1. Choose **File** > **New** > **Folder**.
2. Name the new folder and add your customized code checking rules to this folder.

i Note

The rule files in the folder must be JavaScript files.

3. From the context menu of any file in your project, choose **Project Settings**.
4. From the **Project Settings** options, select **Code Checking** > **JavaScript**.
5. Next to the **Custom Rules Folder** field, choose **Browse**. A popup window displays the projects in the workspace.
6. Open your project and select the rules folder that you created.
7. Choose **OK**.

Your custom rules appear in the **Rules** table instead of the default rules. All the rules are disabled by default.

8. To implement your custom code checking rules, in the **Rules** table, enable each one.

9. Choose [Save](#) to save the new rule configuration for the project.

7.1.5.3.3.2 Customize JavaScript Validator Configuration

You can customize the configuration of SAP Web IDE JavaScript validators.

Context

You can customize the configuration and rules of SAP Web IDE JavaScript validators directly in the SAP Web IDE UI.

If you change the default settings and turn on or off specific rules, this information is stored in a `.eslintrc` file in the root folder of the project. If you have already configured ESLint rules in another development environment, you can import your `.eslintrc` file into the root folder, instead of using the UI. Additional SAP Web IDE-specific information about each rule – such as category, severity and help URL – is stored in an automatically generated `.eslintrc.ext` file, also in the root folder.

Procedure

1. From the context menu of any file in your project, choose [Project Settings](#).
2. From the [Project Settings](#) options, choose [Code Checking](#) [JavaScript](#).
- The default JavaScript validator opens displaying the SAP Web IDE default rules.
3. In the [Validator Configuration](#) field, you can define the validator configuration for globals and environments for the selected validator. The configuration should conform to `.json` file structure.
4. In the [Rules](#) table, configure the rules for the selected validator as follows:
 - a. Enable each rule that you want to use to check your code by selecting the checkbox by the rule name.
 - b. Configure the error level of the rules by setting the severity and category.
 - c. Use the help link for each rule to access detailed rule information about how you can fix the detected issue.
5. Choose [Save](#) and refresh your browser page.

The enabled rules will be implemented when you write your code.

i Note

You can restore the default validator configuration and rules by clicking the [Reset](#) button next to the [Validator](#) field.

7.1.5.3.3 Fiori JavaScript Validator Rules

The Fiori JavaScript validator rules are used when building Fiori projects.

[sap-cross-application-navigation \[page 163\]](#)

No static cross-application navigation targets are allowed.

[sap-forbidden-window-property \[page 165\]](#)

Detects the usage of forbidden window properties.

[sap-no-navigator \[page 166\]](#)

Detects `window.navigator` usage.

[sap-no-override-rendering \[page 168\]](#)

Override of control methods is not allowed.

[sap-no-override-storage-prototype \[page 169\]](#)

Override of `csap-no-override-storage-prototype` is not allowed.

[sap-no-proprietary-browser-api \[page 170\]](#)

Discourage usage of proprietary browser APIs.

[sap-no-sessionstorage \[page 172\]](#)

Usage of session storage is not allowed.

[sap-no-ui5-prop-warning \[page 173\]](#)

Usage of private members of SAPUI5 objects is not allowed.

[sap-no-ui5base-prop \[page 174\]](#)

Usage of private members of SAPUI5 objects is not allowed

[sap-timeout-usage \[page 176\]](#)

Discourage usage of `setTimeout`.

[sap-ui5-no-private-prop \[page 178\]](#)

Detects usage of private properties of SAPUI5 objects.

[sap-usage-basemastercontroller \[page 180\]](#)

Detects usage of `BaseMasterController`.

[sap-message-toast \[page 182\]](#)

Wrong usage of `sap.m.MessageToast` is not allowed.

[sap-no-absolute-component-path \[page 183\]](#)

Absolute paths to component `includes` are not allowed.

[sap-no-br-on-return \[page 185\]](#)

Detects the usage of `document.queryCommandSupported`.

[sap-no-dom-access \[page 186\]](#)

Usage of certain methods of `document` is discouraged.

[sap-no-dom-insertion \[page 188\]](#)

Usage of DOM insertion methods is not allowed.

[sap-no-dynamic-style-insertion \[page 190\]](#)

Detects dynamic style insertion.

[sap-no-element-creation \[page 191\]](#)

Direct DOM insertion is not allowed.

[sap-no-encode-file-service \[page 192\]](#)

Detects the usage of `encode_file` service

[sap-no-exec-command \[page 194\]](#)

Detects direct DOM manipulation.

[sap-no-global-define \[page 195\]](#)

Detects definition of globals via window object.

[sap-no-global-event \[page 197\]](#)

Detects global event handling overrides.

[sap-no-global-selection \[page 198\]](#)

Usage of global selection is discouraged.

[sap-no-global-variable \[page 200\]](#)

Global variables should not be used in SAP Fiori applications.

[sap-no-hardcoded-color \[page 201\]](#)

Usage of hard coded colors is not allowed.

[sap-no-hardcoded-url \[page 202\]](#)

Use of hardcoded URLs is not allowed.

[sap-no-history-manipulation \[page 204\]](#)

Direct history manipulation is discouraged.

[sap-no-jquery-device-api \[page 205\]](#)

Usage of the jQuery device APIs is not allowed.

[sap-no-localhost \[page 207\]](#)

Usage of localhost is not allowed.

[sap-no-localstorage \[page 208\]](#)

Usage of local storage is not allowed.

[sap-no-location-reload \[page 209\]](#)

Detects location reload.

[sap-no-location-usage \[page 211\]](#)

Override of location properties and methods is not allowed.

7.1.5.3.3.1 sap-cross-application-navigation

No static cross-application navigation targets are allowed.

SAP Fiori-as-a-Service Enablement guideline prohibits the use of a static list of cross-application navigation targets.

Rule Details

This check prevents the usage of static cross-application navigation targets.

Instead, use the `IntentSupported` function of the `CrossApplicationNavigation` service. See the corresponding Cross Application Navigation and JSDOC API documentation. Note that the function is mass-enabled, so you can check an array of all relevant navigation targets in one call.

The following patterns are considered warnings:

```
sap.ushell.Container.getService("CrossApplicationNavigation").toExternal({});
```

The following patterns are not warnings:

```
checkPromoFactSheetAvailable : function() {
  // By default: promo factsheet not available
  this._bPromoFactSheetAvailable = false;
  if (this._oCrossAppNav) {
    // Check if the intent for the promotion factsheet is supported
    var sIntent = "#Promotion-displayFactSheet";
    var oDeferred = this._oCrossAppNav.isIntentSupported([sIntent]);
    oDeferred.done(jQuery.proxy(function(oIntentSupported) {
      if (oIntentSupported && oIntentSupported[sIntent] &&
oIntentSupported[sIntent].supported === true) {
        // Remember that the navigation to the promotion factsheet is
possible
        this._bPromoFactSheetAvailable = true;
        // Activate the promotion links if they were already added to
the view
        this.activatePromotionLinks();
      }
    }, this));
  }
}
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)

[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.2 **sap-forbidden-window-property**

Detects the usage of forbidden window properties.

Warning Message

Usage of a forbidden window property.

Rule Details

The following patterns are considered warnings:

```
var top = window.top;
window.addEventListener(listener);
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-no-navigator](#) [page 166]

[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]
[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3 sap-no-navigator

Detects `window.navigator` usage.

Warning Message

`navigator` usage is forbidden, use `sap.ui.Device` API instead.

Rule Details

The `window.navigator` object should not be used at all, instead the `sap.ui.Device` API should be used.

The following patterns are considered warnings:

```
var language = navigator.language;  
var name = navigator.appCodeName;
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)

[sap-no-location-reload](#) [page 209]

[sap-no-location-usage](#) [page 211]

7.1.5.3.3.4 sap-no-override-rendering

Override of control methods is not allowed.

Rule Details

The check detects override of getters, setters, and the functions `onBeforeRendering` and `onAfterRendering` for SAPUI5 controls.

The following patterns are considered warnings:

```
var myButton = new sap.m.Button();
myButton.onAfterRendering = function render(){foo.bar = 1; };
myButton.getWidth = function width(){return 3; };
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation](#) [page 163]

[sap-forbidden-window-property](#) [page 165]

[sap-no-navigator](#) [page 166]

[sap-no-override-storage-prototype](#) [page 169]

[sap-no-proprietary-browser-api](#) [page 170]

[sap-no-sessionstorage](#) [page 172]

[sap-no-ui5-prop-warning](#) [page 173]

[sap-no-ui5base-prop](#) [page 174]

[sap-timeout-usage](#) [page 176]

[sap-ui5-no-private-prop](#) [page 178]

[sap-usage-basemastercontroller](#) [page 180]

[sap-message-toast](#) [page 182]

[sap-no-absolute-component-path](#) [page 183]

[sap-no-br-on-return](#) [page 185]

[sap-no-dom-access](#) [page 186]

[sap-no-dom-insertion](#) [page 188]

[sap-no-dynamic-style-insertion](#) [page 190]

[sap-no-element-creation](#) [page 191]

[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.5 sap-no-override-storage-prototype

Override of `csap-no-override-storage-prototype` is not allowed.

Rule Details

Storage prototype must not be overridden as this can lead to unpredictable errors.

The following patterns are considered warnings:

```
Storage.prototype.setObj = function(key, obj) {};
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-forbidden-window-property](#) [page 165]
[sap-no-navigator](#) [page 166]
[sap-no-override-rendering](#) [page 168]
[sap-no-proprietary-browser-api](#) [page 170]
[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-no-ui5base-prop](#) [page 174]

[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.6 sap-no-proprietary-browser-api

Discourage usage of proprietary browser APIs.

Certain browser APIs are considered to be risk, when used directly and not wrapped via jQuery.

Warning Message

Proprietary Browser API access, use jQuery selector instead.

Rule Details

The check detects the following browser APIs: `document.body.*`, `screen.*`, `window.innerWidth`, `window.innerHeight`.

The following patterns are considered warnings:

```
var variabl = window.innerWidth;  
var variabl = window.innerHeight;  
var myscreen = screen;  
var x = myscreen.something;  
document.body.appendChild(x);  
document.body.style.backgroundColor = 'yellow';
```

The following patterns are not considered warnings:

```
var width= $(window).innerWidth();
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)

[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.7 sap-no-sessionstorage

Usage of session storage is not allowed.

For security reasons, the usage of session storage is not allowed in a SAP Fiori application.

Rule Details

The following patterns are considered warnings:

```
sessionStorage.setObj(this.SETTINGS_NAME, this.objSettings);
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-forbidden-window-property](#) [page 165]
[sap-no-navigator](#) [page 166]
[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]

[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.8 sap-no-ui5-prop-warning

Usage of private members of SAPUI5 objects is not allowed.

Private members of SAPUI5 objects must never be used in SAP Fiori applications. They can be changed by SAPUI5 at anytime and the application might not work anymore.

Rule Details

The rule checks usage of a member that has the same name as the following SAPUI5 members:

`sap.ui.model.odata.ODataModel`, `sap.ui.model.odata.v2.ODataModel` `oData`

False Positives

As the check cannot determine whether the property used is from a SAPUI5 object, there might be false positives if you defined a property with the same name in your own object. In this case, you can disable the check in your coding as follows:

```
/* eslint-disable sap-no-ui5base-prop */  
...some code false positives...  
/* eslint-enable sap-no-ui5base-prop */
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.3.9 sap-no-ui5base-prop

Usage of private members of SAPUI5 objects is not allowed

Private members of SAPUI5 objects must never be used in SAP Fiori applications. They can be changed by SAPUI5 at anytime and the application might not work anymore.

Rule Details

The rule checks usage of a member that has the same name as the following SAPUI5 members:

<code>sap.ui.base.ManagedObject</code>	<code>mProperties, mAggregations, mAssociations, mMethods, oParent, aDelegates, aBeforeDelegates, iSuppressInvalidate, oPropagatedProperties, oModels, oBindingContexts, mBindingInfos, sBindingPath, mBindingParameters, mBoundObjects</code>
<code>sap.ui.base.EventProvider</code>	<code>mEventRegistry, oEventPool</code>
<code>sap.ui.base.Event</code>	<code>oSource, mParameters, sId</code>
<code>sap.ui.model.odata.ODataModel,</code> <code>sap.ui.model.odata.v2.ODataModel</code>	<code>oServiceData, bCountSupported, bCache, oRequestQueue, aBatchOperations, oHandler, mSupportedBindingModes, sDefaultBindingMode, bJSON, aPendingRequestHandles, aCallAfterUpdate, mRequests, mDeferredRequests, mChangedEntities, mChangeHandles, mDeferredBatchGroups, mChangeBatchGroups, bTokenHandling, bWithCredentials, bUseBatch, bRefreshAfterChange, sMaxDataServiceVersion, bLoadMetadataAsync, bLoadAnnotationsJoined, sAnnotationURI, sDefaultCountMode, sDefaultOperationMode, oMetadataLoadEvent, oMetadataFailedEvent, sRefreshBatchGroupId, sDefaultChangeBatchGroup, oAnnotations, aUrlParams</code>

False Positives

As the check cannot determine whether the property used is from a SAPUI5 object, there might be false positives if you defined a property with the same name in your own object. In this case, you can disable the check in your coding as follows:

```
/* eslint-disable sap-no-ui5base-prop */  
...some code false positives  
/* eslint-enable sap-no-ui5base-prop */
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)

[sap-forbidden-window-property \[page 165\]](#)

[sap-no-navigator \[page 166\]](#)

[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]
[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.10 sap-timeout-usage

Discourage usage of `setTimeout`.

This rule finds calls to the `setTimeout` method with a timeout greater than 0.

Warning Message

```
Timeout with value > 0
```


Rule Details

Executing logic with timeouts is often a workaround for faulty behavior and does not fix the root cause. The timing that works for you may not work under different circumstances (other geographical locations with greater network latency, or other devices that have slower processors) or when the code is changed. Use callbacks or events instead, if available. Please check the SAPUI5 guidelines for more details.

The following patterns are considered warnings:

```
window.setTimeout(jQuery.proxy(processChanges, this), 50)
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

- [sap-cross-application-navigation \[page 163\]](#)
- [sap-forbidden-window-property \[page 165\]](#)
- [sap-no-navigator \[page 166\]](#)
- [sap-no-override-rendering \[page 168\]](#)
- [sap-no-override-storage-prototype \[page 169\]](#)
- [sap-no-proprietary-browser-api \[page 170\]](#)
- [sap-no-sessionstorage \[page 172\]](#)
- [sap-no-ui5-prop-warning \[page 173\]](#)
- [sap-no-ui5base-prop \[page 174\]](#)
- [sap-ui5-no-private-prop \[page 178\]](#)
- [sap-usage-basemastercontroller \[page 180\]](#)
- [sap-message-toast \[page 182\]](#)
- [sap-no-absolute-component-path \[page 183\]](#)
- [sap-no-br-on-return \[page 185\]](#)
- [sap-no-dom-access \[page 186\]](#)
- [sap-no-dom-insertion \[page 188\]](#)
- [sap-no-dynamic-style-insertion \[page 190\]](#)
- [sap-no-element-creation \[page 191\]](#)
- [sap-no-encode-file-service \[page 192\]](#)
- [sap-no-exec-command \[page 194\]](#)
- [sap-no-global-define \[page 195\]](#)
- [sap-no-global-event \[page 197\]](#)
- [sap-no-global-selection \[page 198\]](#)
- [sap-no-global-variable \[page 200\]](#)
- [sap-no-hardcoded-color \[page 201\]](#)
- [sap-no-hardcoded-url \[page 202\]](#)
- [sap-no-history-manipulation \[page 204\]](#)
- [sap-no-jquery-device-api \[page 205\]](#)

[sap-no-localhost \[page 207\]](#)

[sap-no-localstorage \[page 208\]](#)

[sap-no-location-reload \[page 209\]](#)

[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.11 sap-ui5-no-private-prop

Detects usage of private properties of SAPUI5 objects.

The SAP Fiori guideline prohibits the use of SAPUI5 properties or private functions.

Rule Details

The rule detects SAPUI5 objects of the following namespaces:

- `sap.ui.core`
- `sap.apf`
- `sap.ca.scfld.md`
- `sap.ca.ui`
- `sap.chart`
- `sap.collaboration`
- `sap.fiori`
- `sap.landvisz`
- `sap.m`
- `sap.makit`
- `sap.me`
- `sap.ndc`
- `sap.ovp`
- `sap.portal.ui5`
- `sap.suite.ui.commons`
- `sap.suite.ui.generic.template`
- `sap.suite.ui.microchart`
- `sap.tnt`
- `sap.ui.commons`
- `sap.ui.comp`
- `sap.ui.dt`
- `sap.ui.fl`
- `sap.ui.generic.app`
- `sap.ui.generic.template`
- `sap.ui.layout`
- `sap.ui.richtexteditor`

- `sap.ui.rta`
- `sap.ui.server.abap`
- `sap.ui.server.java`
- `sap.ui.suite`
- `sap.ui.table`
- `sap.ui.unified`
- `sap.ui.ux3`
- `sap.ui.vbm`
- `sap.ui.vk`
- `sap.uiext.inbox`
- `sap.ushell`
- `sap.uxap sap.viz`

A reference to a property or private function of these objects (indicated by a leading `_`) is not permitted.

The following patterns are considered warnings:

```
var me = sap.me; me.age = 42;
var me = sap.me; me._setAge(10);
sap.ca.ui.utils.BUSYDIALOG_TIMEOUT = 0;
var btn = new sap.m.Button(); btn.myPrivateProperty = "X";
```

The following patterns are not considered warnings:

```
var me = sap.me; me.getAge();
var me = sap.me; me.setMood("lazy");
```

Custom Namespaces

It is possible to add custom namespaces to this check:

- **Inline Config**
You can add an inline comment to override the rule configuration, as follows:

```
/*eslint sap-ui5-no-private-prop: [1, {"ns": ["<myOwnNamespace>",
"<anotherNamespace>"]}】*/
```
- **ESLint Config File**
You can add your custom namespaces to the `.eslintrc` config file

```
"sap-ui5-no-private-prop": [1, {"ns": ["<myOwnNamespace>",
"<anotherNamespace>"]}】,
```
- **WebIDE Validator Settings**
You can configure custom namespaces in the validator settings in SAP Web IDE.

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-forbidden-window-property](#) [page 165]
[sap-no-navigator](#) [page 166]
[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]
[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.12 sap-usage-basemastercontroller

Detects usage of `BaseMasterController`.

`BaseMasterController` is a deprecated controller and should be replaced by `sap.ca.scfld.md.controller.ScfldMasterController`.

Rule Details

The rule detects the usage of the object `sap.ca.scfld.md.controller.BaseMasterController` and the usage of the string `sap/ca/scfld/md/controller/BaseMasterController`, like in `define-methods`.

The following patterns are considered warnings:

```
sap.ca.scfld.md.controller.BaseMasterController.extend('myBaseController', {
  config: 'myconfig'
});
define(['sap/ca/scfld/md/controller/BaseMasterController'], function(Controller) {
  Controller.extend('myBaseController', {
    config: 'myconfig'
  });
});
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)

[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.3.13 sap-message-toast

Wrong usage of `sap.m.MessageToast` is not allowed.

The SA[Fiori design guidelines require a certain behavior of a message toast.

Rule Details

This check looks for any call of the method `show` on the `sap.m.MessageToast-Object` and checks the following properties:

- `duration` must not be smaller than 3000
- `width` must not be greater than 35em
- `my` must be *center bottom*
- `at` must be *center bottom*

The following patterns are considered warnings:

```
sap.m.MessageToast.show("This is a warning!", { duration: 1000 })
```

The following patterns are not considered warnings:

```
sap.m.MessageToast.show("This is a warning!");
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)

[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.3.14 sap-no-absolute-component-path

Absolute paths to component `includes` are not allowed.

Rule Details

The rule checks if `includes` inside a component have a leading `/`.

The following patterns are considered warnings:

```
code:
"sap.ui.core.UIComponent.extend('sap.ui.demokit.explored.Component', { "
    + "metadata : { "
    + "includes : [ "
    + "'css/style2.css', "
```

```

'sap.ui.demokit.explored.view', "
+ "/css/style2.css", "
+ "/css/titles.css" "
+ "], "
+ "routing : { "
+ "config : { "
+ "routerClass : MyRouter, "
+ "viewType : 'XML', "
+ "viewPath :
+ "targetControl : 'splitApp', "
+ "clearTarget : false "
+ "}, "
+ "routes : [ { "
+ "pattern : 'entity/{id}/{part}', "
+ "name : 'entity', "
+ "view : 'entity', "
+ "viewLevel : 3, "
+ "targetAggregation : 'detailPages' "
+ "} ]" + "}" + "}" + "});"

```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)

[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.15 sap-no-br-on-return

Detects the usage of `document.queryCommandSupported`.

This rule checks any call of `queryCommandSupported` on `document`. Calls with argument `sap-no-br-on-return` are not allowed because this is a browser specific command.

Warning Message

```
insertBrOnReturn is not allowed since it is a Mozilla specific method, other browsers don't support that.
```

Rule Details

The rule checks each `IconTabBar`. All tags with just one `IconTabFilter` will be found.

The following patterns are considered warnings:

```
var abc = document.queryCommandSupported('insertBrOnReturn');
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-forbidden-window-property](#) [page 165]
[sap-no-navigator](#) [page 166]
[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]

[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.16 **sap-no-dom-access**

Usage of certain methods of document is discouraged.

Accessing the DOM directly is considered risky. If necessary, a jQuery selector should be used instead.

Warning Message

```
Direct DOM access, use jQuery selector instead
```

Rule Details

The following methods are not allowed to be used:

- `getElementById`
- `getElementsByName`
- `getElementsByTagName`
- `getElementsByClassName`

The following patterns are considered warnings:

```
document.getElementById('test');
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)

[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.17 sap-no-dom-insertion

Usage of DOM insertion methods is not allowed.

The SAPUI5 guidelines do not allow insertion of elements into the DOM. Instead usage of a custom control should be considered.

Rule Details

The rule detects all method calls of `insertBefore` , `appendChild` , `replaceChild` , `after` , `before` , `insertAfter` , `insertBefore` , `append` , `prepend` , `appendTo` , `prependTo` .

The following patterns are considered warnings:

```
$("#container").append("Test");  
var list = document.getElementById("myList1");  
list.insertBefore(node, list.childNodes[0]);  
myObject.after(document.body);
```

False Positives

There might be cases where the check produces a false positive, i.e. when you have a method containing one of the strings given above. In this case, you can change the method name or deactivate the rule by placing the following pseudo-comment block around your code. It is recommended to have your code reviewed before you enter such a pseudo-comment.

```
/*eslint-disable sap-no-dom-insertion*/  
  <your code>  
/*eslint-enable sap-no-dom-insertion*/
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.18 sap-no-dynamic-style-insertion

Detects dynamic style insertion.

Warning Message

```
Dynamic style insertion, use library CSS or lessifier instead
```

Rule Details

The check detects any usage of `document.styleSheets`.

The following patterns are considered warnings:

```
var sheet = document.styleSheets[i];  
var abc = document.styleSheets.length;
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-element-creation \[page 191\]](#)

[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.3.19 sap-no-element-creation

Direct DOM insertion is not allowed.

The SAPUI5 guidelines do not allow creation of elements in the DOM. Instead usage of a custom control should be considered.

Warning Message

Direct DOM insertion, create a custom control instead

Rule Details

The rule detects all method calls of `createElement`, `createTextNode`, `createElementNS`, `createDocumentFragment`, `createComment`, `createAttribute`, `createEvent`.

The following patterns are considered warnings:

```
document.createElement (foo) ;
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-forbidden-window-property](#) [page 165]
[sap-no-navigator](#) [page 166]
[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]
[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.20 sap-no-encode-file-service

Detects the usage of `encode_file` service

The `encode_file` service is deprecated and not available on SAP BTP.

Rule Details

The rule detects the usage of the string `/sap/bc/ui2/encode_file`.

The following patterns are considered warnings:

```
oFileUpload.setEncodeUrl('/sap/bc/ui2/encode_file' + (sUrlParams ? '?' +  
sUrlParams : ''));  
var service = '/sap/bc/ui2/encode_file';
```

How to fix:

Use the `sap.m.UploadCollection` with the `sap.m.UploadCollectionItem` instead.

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)

[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.21 sap-no-exec-command

Detects direct DOM manipulation.

Warning Message

```
Direct DOM Manipulation, better to use jQuery.appendTo if really needed
```

Rule Details

The rule detects usage of the `execCommand` method.

The following patterns are considered warnings:

```
document.execCommand(cmd, false, args);
```

```
document['execCommand'](cmd, false, args);
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-forbidden-window-property](#) [page 165]
[sap-no-navigator](#) [page 166]
[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]

[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.22 sap-no-global-define

Detects definition of globals via window object.

Warning Message

```
Definition of global variable/api in window object is not permitted.
```

Rule Details

Global variables should not be used in SAP Fiori applications. This check detects global definitions by attachments to the window object or override of window properties.

The following patterns are considered warnings:

```
window.MyVar = "A";  
window.name = "New Name";
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.3.23 sap-no-global-event

Detects global event handling overrides.

The SAPUI5 guidelines do not allow overriding global event handling.

Warning Message

```
Global event handling override is not permitted, please modify only single events.
```

Rule Details

This rule detects overrides of the following global events: `onload` , `onunload` , `onabort` , `onbeforeunload` , `onerror` , `onhashchange` , `onpageshow` , `onpagehide` , `onscroll` , `onblur` , `onchange` , `onfocus` , `onfocusin` , `onfocusout` , `oninput` , `oninvalid` , `onreset` , `onsearch` , `onselect` , `onsubmit`.

The following patterns are considered warnings:

```
window.event.returnValue = false;
window.onload = function(){ return Hammer; };
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)

[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.24 sap-no-global-selection

Usage of global selection is discouraged.

According to SAP Fiori design guidelines, it is not allowed to have an `IconTabBar` with just a single `IconTabFilter`.

Warning Message

```
Global selection modification, only modify local selections
```

Rule Details

The following patterns are considered warnings:

```
window.getSelection().rangeCount = 9;
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.25 sap-no-global-variable

Global variables should not be used in SAP Fiori applications.

Rule Details

The rule checks if a variable is declared as global (defined outside of any function scope) and returns an error message in this case.

Allowed variables are = ["undefined", "NaN", "arguments", "PDFJS", "console", "Infinity"]

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-hardcoded-color \[page 201\]](#)

[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.26 sap-no-hardcoded-color

Usage of hard coded colors is not allowed.

It is not allowed to style SAP Fiori applications with colors in JavaScript code as they will break the SAP Fiori themes.

Rule Details

The following patterns are considered warnings:

```
$(\"<div id='lasso-selection-help' style='position:absolute;pointer-events:none;background:#cccccc;'></div>\")
```

How to Fix

Do not specify colors in custom CSS but use the standard theme-dependent classes instead.

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-forbidden-window-property](#) [page 165]
[sap-no-navigator](#) [page 166]
[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]
[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]

[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.27 sap-no-hardcoded-url

Use of hardcoded URLs is not allowed.

SAP Fiori guidelines do not allow usage of hardcoded URLs to internal or external systems.

Rule Details

Instead of references to internal systems in your URLs, you should only reference the path to the resource.

Allowed URLs are:

<http://www.w3.org/> , <http://www.sap.com/Protocols/> , <http://www.sap.com/adt> ,
<http://localhost/offline/> , <https://localhost/offline/>

The following patterns are considered warnings:

```
serviceUrl: URI("http://ldcigm3.wdf.sap.corp:50057/sap/opu/odata/sap/  
XXXX/").directory(),
```

```
serviceUrl: "proxy/http/ldcigm3.wdf.sap.corp:50057/sap/opu/odata/sap/XXXX/"
```

The following patterns are not considered warnings:

```
serviceUrl: "/sap/opu/odata/sap/FDMO_PROCESS_RECEIVABLES_SRV/",
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.28 sap-no-history-manipulation

Direct history manipulation is discouraged.

Warning Message

Direct history manipulation, does not work with deep links, use router and navigation events instead

Rule Details

The following patterns are considered warnings:

```
window.history.back();
```

```
history.go(-3);
```

```
var personalHistory = window.history;
personalHistory.back();
```

The following patterns are not considered warnings:

```
myNavBack : function(sRoute, mData) {
  var oHistory = sap.ui.core.routing.History.getInstance();
  var sPreviousHash = oHistory.getPreviousHash();
  //The history contains a previous entry
  if (sPreviousHash !== undefined) {
    window.history.go(-1);
  } else {
    var bReplace = true; // otherwise we go backwards with a forward history
    this.navTo(sRoute, mData, bReplace)
  }
},
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)

[sap-forbidden-window-property \[page 165\]](#)

[sap-no-navigator \[page 166\]](#)

[sap-no-override-rendering \[page 168\]](#)

[sap-no-override-storage-prototype \[page 169\]](#)

[sap-no-proprietary-browser-api \[page 170\]](#)

[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]
[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.29 sap-no-jquery-device-api

Usage of the jQuery device APIs is not allowed.

The jQuery device API is deprecated since SAPUI5 1.20. The respective functions of `sap.ui.Device` should be used instead.

Rule Details

The check looks for any call of `jQuery.device`.

The following patterns are considered warnings:

```
if (jQuery.device.is.android_phone === false) {}  
if ($.device.is.android_phone === false) {}
```

The following patterns are not not considered warnings:

```
if (!sap.ui.Device.system.desktop) {  
    this.getView().byId("factSheetButton").setVisible(false);  
}
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-reload \[page 209\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.3.30 sap-no-localhost

Usage of localhost is not allowed.

Usage of localhost in SAP Fiori applications is often done for debugging or test reasons and should be avoided in productive code.

Rule Details

The check detects the string localhost in any JavaScript function call or expression. The usage of localhost in an offline scenario is allowed, therefore the coding mentioned below will not raise a warning.

The following patterns are considered warnings:

```
if (location.hostname === "localhost"){};
location.host.indexOf("localhost");
```

The following patterns are not considered warnings:

```
return "http://localhost/offline/my_contacts/ContactCollection";
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)

[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.31 sap-no-localstorage

Usage of local storage is not allowed.

Local storage must not be used in a SAP Fiori application

Rule Details

The following patterns are considered warnings:

```
localStorage.setObj(this.SETTINGS_NAME, this.objSettings);
```

Parent topic: [Fiori JavaScript Validator Rules](#) [page 162]

Related Information

[sap-cross-application-navigation](#) [page 163]
[sap-forbidden-window-property](#) [page 165]
[sap-no-navigator](#) [page 166]
[sap-no-override-rendering](#) [page 168]
[sap-no-override-storage-prototype](#) [page 169]
[sap-no-proprietary-browser-api](#) [page 170]
[sap-no-sessionstorage](#) [page 172]
[sap-no-ui5-prop-warning](#) [page 173]

[sap-no-ui5base-prop](#) [page 174]
[sap-timeout-usage](#) [page 176]
[sap-ui5-no-private-prop](#) [page 178]
[sap-usage-basemastercontroller](#) [page 180]
[sap-message-toast](#) [page 182]
[sap-no-absolute-component-path](#) [page 183]
[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-location-reload](#) [page 209]
[sap-no-location-usage](#) [page 211]

7.1.5.3.3.3.32 sap-no-location-reload

Detects location reload.

SAP Fiori guidelines do not allow `location.reload()`.

Warning Message

```
location.reload() is not permitted.
```

Rule Details

This check detects usage of `location.reload()`.

The following patterns are considered warnings:

```
location.reload();  
var mylocation = location; mylocation.reload();
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)
[sap-no-br-on-return \[page 185\]](#)
[sap-no-dom-access \[page 186\]](#)
[sap-no-dom-insertion \[page 188\]](#)
[sap-no-dynamic-style-insertion \[page 190\]](#)
[sap-no-element-creation \[page 191\]](#)
[sap-no-encode-file-service \[page 192\]](#)
[sap-no-exec-command \[page 194\]](#)
[sap-no-global-define \[page 195\]](#)
[sap-no-global-event \[page 197\]](#)
[sap-no-global-selection \[page 198\]](#)
[sap-no-global-variable \[page 200\]](#)
[sap-no-hardcoded-color \[page 201\]](#)
[sap-no-hardcoded-url \[page 202\]](#)
[sap-no-history-manipulation \[page 204\]](#)
[sap-no-jquery-device-api \[page 205\]](#)
[sap-no-localhost \[page 207\]](#)
[sap-no-localstorage \[page 208\]](#)
[sap-no-location-usage \[page 211\]](#)

7.1.5.3.3.33 sap-no-location-usage

Override of location properties and methods is not allowed.

Warning Message

- Usage of `location.assign()`
- Override of `location`

Rule Details

This check detects usage of `window.location.assign()` and override of the `window.location` object as well as any of its properties.

The following patterns are considered warnings:

```
location.assign(data.results[0].url);
window.location.hash = "#foo";
location = this.oNavParams.toOppApp;
location.myProperty = this.oNavParams.toOppApp;
location.href = myHref;
```

Parent topic: [Fiori JavaScript Validator Rules \[page 162\]](#)

Related Information

[sap-cross-application-navigation \[page 163\]](#)
[sap-forbidden-window-property \[page 165\]](#)
[sap-no-navigator \[page 166\]](#)
[sap-no-override-rendering \[page 168\]](#)
[sap-no-override-storage-prototype \[page 169\]](#)
[sap-no-proprietary-browser-api \[page 170\]](#)
[sap-no-sessionstorage \[page 172\]](#)
[sap-no-ui5-prop-warning \[page 173\]](#)
[sap-no-ui5base-prop \[page 174\]](#)
[sap-timeout-usage \[page 176\]](#)
[sap-ui5-no-private-prop \[page 178\]](#)
[sap-usage-basemastercontroller \[page 180\]](#)
[sap-message-toast \[page 182\]](#)
[sap-no-absolute-component-path \[page 183\]](#)

[sap-no-br-on-return](#) [page 185]
[sap-no-dom-access](#) [page 186]
[sap-no-dom-insertion](#) [page 188]
[sap-no-dynamic-style-insertion](#) [page 190]
[sap-no-element-creation](#) [page 191]
[sap-no-encode-file-service](#) [page 192]
[sap-no-exec-command](#) [page 194]
[sap-no-global-define](#) [page 195]
[sap-no-global-event](#) [page 197]
[sap-no-global-selection](#) [page 198]
[sap-no-global-variable](#) [page 200]
[sap-no-hardcoded-color](#) [page 201]
[sap-no-hardcoded-url](#) [page 202]
[sap-no-history-manipulation](#) [page 204]
[sap-no-jquery-device-api](#) [page 205]
[sap-no-localhost](#) [page 207]
[sap-no-localstorage](#) [page 208]
[sap-no-location-reload](#) [page 209]

7.1.5.3.3.4 Set JavaScript Rules for All Users

You can centrally define ESLint validation rules for JavaScript code checking, which then become the default rules for all projects for all users in the SAP BTP account. This capability is very useful when an SAP BTP account administrator wants to enforce a common ESLint validation rules for JavaScript code, checking all applications that are developed within the account.

Context

i Note

You must be an administrator of your SAP BTP account.

Procedure

1. Open SAP BTP cockpit.

i Note

You can access SAP BTP cockpit from SAP Web IDE by selecting **Tools > SAP BTP Cockpit**.

2. Open the SAP Web IDE page for JavaScript rules.
 - a. In the SAP BTP cockpit, go to [Services](#).
 - b. Select the [SAP Web IDE](#) service.
 - c. Select [JavaScript Rules](#).
3. Select [Import](#), a ZIP file, comprised of your ESLint rules files (`eslinttrc` and `eslinttrcext`).

i Note

- To explore an ESLint rule file: locate the files `eslinttrc` and `eslinttrcext` in the file structure of your project, and export them.
- The new rule configuration is selected by default upon refreshing SAP Web IDE.
- Developers can still manually change the uploaded rules. The rules are applied in the following sequence – local rules, account rules, system rules.

7.1.5.3.4 XML Validation

You can configure which XML validator to use in your project.

Context

SAP Web IDE uses a default XML validator to check the syntax of your XML files.

[XML Semantic Validation \[page 214\]](#)

SAP Web IDE performs semantic validation on XML files.

[SAP Fiori XML Validator Rules \[page 215\]](#)

The SAP Fiori XML validator rules are used when building SAP Fiori projects.

Task overview: [Checking Code \[page 155\]](#)

Related Information

[Configure Code Checking \[page 156\]](#)

[Code Checking Annotations \[page 157\]](#)

[JavaScript Validation \[page 159\]](#)

[Validation of neo-app.json Files \[page 239\]](#)

[Validation of manifest.json Files \[page 239\]](#)

[Validation of Stable IDs \[page 240\]](#)

[Validation of i18n Files \[page 241\]](#)

[Using the Problems View \[page 242\]](#)

7.1.5.3.4.1 XML Semantic Validation

SAP Web IDE performs semantic validation on XML files.

SAP Web IDE performs XML semantic validation including deprecation checks using the following rules.

Rule	Description
XML_INVALID_AGGR_NODE	The aggregation node for the specified control is incorrect. Enter the correct node name.
XML_INVALID_CANDIDATE	The candidate for the specified aggregation node is incorrect. Enter a candidate of the correct type.
XML_INVALID_MULTIPLICITY	The specified aggregation is allowed only one child node but it has more than one child. You need to remove the extra nodes.
XML_INVALID_ID	The control ID is incorrect.
XML_DUPLICATE_ID	The control ID must be unique but it is the same as another control ID in the view. Change the ID so that it is unique
XML_INVALID_CSS	The css class is incorrect.
XML_INVALID_ASSOCIATION_VALUE	The Association property value is incorrect.
XML_INVALID_FORMATTER_FUNC	The formatter function name is incorrect.
XML_INVALID_EVENT_FUNC	The event function name is incorrect.
XML_INVALID_PROPERTY_NAME	The property or event name does not exist for the control. Enter the correct property or event name.
XML_INVALID_PROPERTY_VALUE	The value of the specified property is incorrect for the specified mode type.
XML_INVALID_PROPERTY_ENUM_VALUE	The specified property value is incorrect for the specified enum type of the specified property.
XML_DEPRECATED_CONTROL	The specified control is deprecated since the specified version.
XML_DEPRECATED_PROPERTY	The specified property is deprecated since the specified version.
XML_DEPRECATED_EVENT	The specified event is deprecated since the specified version.
XML_DEPRECATED_AGGREGATION	The specified aggregation is deprecated since the specified version.

Parent topic: [XML Validation \[page 213\]](#)

Related Information

[SAP Fiori XML Validator Rules \[page 215\]](#)

7.1.5.3.4.2 SAP Fiori XML Validator Rules

The SAP Fiori XML validator rules are used when building SAP Fiori projects.

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)

Detects buttons with text and icons.

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)

Detects lists and tables with `showNoData` attributes that are not set to `true`.

[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)

Detects duplicate icons in `IconTabBar` tags.

[DG_XML_NO_SINGLE_TAB \[page 220\]](#)

Detects a single `IconTabFilter` tag in `IconTabBar` tags.

[XML_COMMONS_USAGE \[page 222\]](#)

Detects the usage of `sap.ui.commons` objects.

[XML_DEPRECATION \[page 223\]](#)

Checks for deprecated controls.

[XML_DIALOG_IN_VIEW \[page 224\]](#)

Checks `Dialog`, `Popover`, `ResponsivePopover`, and `ActionSheet` tags in views.

[XML_FORM_USAGE \[page 225\]](#)

Checks the usage of the `sap.ui.commons.form` tags.

[XML_ICON_ACCESSIBILITY \[page 226\]](#)

Checks accessibility for icons.

[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)

Checks accessibility for icons.

[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)

Checks accessibility for images.

[XML_LAYOUT_USAGE \[page 230\]](#)

Detects the usage of the `sap.ui.commons.layout` tags.

[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)

Checks usage of absolute `media_src` URLs.

[XML_MISSING_STABLE_ID \[page 232\]](#)

Checks for stable IDs for controls.

[XML_PAGE_ACCESSIBILITY \[page 233\]](#)

Detects missing `title` attributes.

[XML_TABLE_ACCESSIBILITY \[page 234\]](#)

Detects missing `title` attributes.

[XML_TITLE_ACCESSIBILITY \[page 235\]](#)

Detects missing `title` attributes.

[XML_UPLOAD_IN_VIEW \[page 236\]](#)

Checks usage of `FileUpload` and `AddPicture` tags.

[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

Checks your setting of the `serviceRefreshInterval` attribute.

Parent topic: [XML Validation \[page 213\]](#)

Related Information

[XML Semantic Validation \[page 214\]](#)

7.1.5.3.4.2.1 DG_XML_FOOTER_BUTTON_TEXT_ICON

Detects buttons with text and icons.

According to the *SAP Fiori Design Guidelines*, a button in a page's footer bars must not have text and icons at the same time.

Warning Message

A footer button must either have an icon or a text

Rule Details

The rule detects `Button` tags with text and icon attributes in a page's footer bars. All tags with text and icons are found.

The following patterns are considered warnings:

```
<Page title="Page">
  <content></content>
  <footer>
    <Toolbar>
      <Button text="Submit" icon="sap-icon://send" type="Accept"/>
    </Toolbar>
  </footer>
</Page>
```



```
</footer>
</Page>
```

The following patterns are not considered warnings:

```
<Page title="Page">
  <content></content>
  <footer>
    <Toolbar>
      <Button icon="sap-icon://send" type="Accept"/>
    </Toolbar>
  </footer>
```

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.2 DG_XML_LIST_BASE_SHOW_NO_DATA

Detects lists and tables with `showNoData` attributes that are not set to `true`.

According to the *SAP Fiori Design Guidelines*, lists and tables need to display `No Data` text when they do not contain any data.

Warning Message

Attribute `showNoData` must not be set to `false`

Rule Details

The rule checks whether each list and table tag has a `showNoData` attribute. All tags with `showNoData` not set to `true` are found.

The following patterns are considered warnings:

```
<List
    headerText="Products"
    showNoData="yes"
    binding="{/ProductCollection/0}" >
  <StandardListItem
    title="{Name}"
    description="{ProductId}"
    icon="{ProductPicUrl}"
    iconDensityAware="false"
    iconInset="false" />
</List>
<List
    headerText="Products"
    showNoData=""
    binding="{/ProductCollection/0}" >
  <StandardListItem
    title="{Name}"
    description="{ProductId}"
    icon="{ProductPicUrl}"
    iconDensityAware="false"
    iconInset="false" />
</List>
```

The following patterns are not considered warnings:

```
<List
    headerText="Products"
    binding="{/ProductCollection/0}" >
  <StandardListItem
    title="{Name}"
    description="{ProductId}"
    icon="{ProductPicUrl}"
    iconDensityAware="false"
    iconInset="false" />
</List>
```

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)

[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)

[DG_XML_NO_SINGLE_TAB](#) [page 220]
[XML_COMMONS_USAGE](#) [page 222]
[XML_DEPRECATION](#) [page 223]
[XML_DIALOG_IN_VIEW](#) [page 224]
[XML_FORM_USAGE](#) [page 225]
[XML_ICON_ACCESSIBILITY](#) [page 226]
[XML_ICON_BUTTON_ACCESSIBILITY](#) [page 227]
[XML_IMAGE_ACCESSIBILITY](#) [page 229]
[XML_LAYOUT_USAGE](#) [page 230]
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 231]
[XML_MISSING_STABLE_ID](#) [page 232]
[XML_PAGE_ACCESSIBILITY](#) [page 233]
[XML_TABLE_ACCESSIBILITY](#) [page 234]
[XML_TITLE_ACCESSIBILITY](#) [page 235]
[XML_UPLOAD_IN_VIEW](#) [page 236]
[XML_BOOKMARK_PERFORMANCE](#) [page 237]

7.1.5.3.4.2.3 DG_XML_NO_DUPLICATE_ICONS

Detects duplicate icons in `IconTabBar` tags.

According to the *SAP Fiori Design Guidelines*, an `IconTabFilter` tag cannot contain the same icon twice.

Warning Message

A specific icon must not occur twice in a tab bar

Rule Details

The rule checks each `IconTabFilter` tag in all `IconTabBar` tags. All tags with the same icon value are found.

The following patterns are considered warnings:

```
<IconTabBar id="idIconTabBar">
  <items>
    <IconTabFilter
      icon="sap-icon://begin"
      text="Heavy"
      key="Heavy" />
    <IconTabSeparator />
    <IconTabFilter
      icon="sap-icon://begin"
      text="Overweight"
      key="Overweight" />
```

```
</items>  
</IconTabBar>
```

False Positives

The rule can't detect the visibility of an element. So there might be cases when a duplicate icon is not visible. In such a cases, please ignore the warning.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.4 DG_XML_NO_SINGLE_TAB

Detects a single `IconTabFilter` tag in `IconTabBar` tags.

According to the *SAP Fiori Design Guidelines*, an `IconTabBar` cannot contain just a single `IconTabFilter`.

Warning Message

An `IconTabBar` should have more than one `IconTab`

Rule Details

The rule checks each `IconTabBar` and finds all tags with just one `IconTabFilter`.

The following patterns are considered warnings:

```
<IconTabBar
    id="idIconTabBar"
    select="handleIconTabBarSelect"
    class="sapUiResponsiveContentPadding">
  <items>
    <IconTabFilter
      showAll="true"
      count="{/ProductCollectionStats/Counts/Total}"
      text="Products"
      keys="All" />
    <IconTabSeparator />
  </items>
</IconTabBar>
```

False Positives

If there are multiple `IconTabFilter` tags, and some of them are invisible, the rule does not detect a violation if just one `IconTabFilter` is visible.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)

[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)

[XML_COMMONS_USAGE \[page 222\]](#)

[XML_DEPRECATION \[page 223\]](#)

[XML_DIALOG_IN_VIEW \[page 224\]](#)

[XML_FORM_USAGE \[page 225\]](#)

[XML_ICON_ACCESSIBILITY \[page 226\]](#)

[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)

[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)

[XML_LAYOUT_USAGE \[page 230\]](#)

[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)

[XML_MISSING_STABLE_ID \[page 232\]](#)

[XML_PAGE_ACCESSIBILITY \[page 233\]](#)

[XML_TABLE_ACCESSIBILITY \[page 234\]](#)

[XML_TITLE_ACCESSIBILITY \[page 235\]](#)

[XML_UPLOAD_IN_VIEW \[page 236\]](#)

[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.5 XML_COMMONS_USAGE

Detects the usage of `sap.ui.commons` objects.

According to the *SAP Fiori Architectural Guidelines*, controls from `sap.ui.commons` objects are not allowed. Instead, `sap.m` controls should be used.

Warning Message

Usage of `sap.ui.commons` controls is forbidden, please use controls from `sap.m` / `sap.me` or `sap.ca`

Rule Details

The rule checks each tag for the `sap.ui.commons` namespace. Each finding is reported.

The following patterns are considered warnings:

```
<sap.ui.commons:table>growFactor="1"</sap.ui.commons:table>
```

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)

[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)

[DG_XML_NO_SINGLE_TAB \[page 220\]](#)

[XML_DEPRECATION \[page 223\]](#)

[XML_DIALOG_IN_VIEW \[page 224\]](#)

[XML_FORM_USAGE \[page 225\]](#)

[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.6 XML_DEPRECATION

Checks for deprecated controls.

Controls marked as deprecated in the API Reference documentation are no longer intended to be used. They will not get feature updates in the future.

Warning Message

```
A deprecated control is used in the XML view
```

Rule Details

For each control, the check scans for deprecation in the API documentation in the corresponding version loaded from Nexus.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)

[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.7 XML_DIALOG_IN_VIEW

Checks `Dialog`, `Popover`, `ResponsivePopover`, and `ActionSheet` tags in views.

Dialogs should not be declared in a view as they will result in UI artifacts (you can see them inside the footer bar on your detail view).

Warning Message

Dialogs should not be declared in the view but rather in a separate fragment as they can result in UI artifacts

Rule Details

The rule detects the usage of `Dialog`, `Popover`, `ResponsivePopover`, and `ActionSheet` tags in views.

How to Fix

- Declare all your `Dialog` / `Popover` / `ResponsivePopover` / `ActionSheet` tags in separate XML files that will be used for the fragment.
- Instantiate those fragments from the JS controller of the view (as you were probably doing before), for example:.

```
var myPopover = sap.ui.xmlfragment("my.useful.VerySimpleUiPart", this);
```


The second parameter is going to be the controller that will be passed to this fragment, in this case it will reuse the controller of your view. If you were already defining your dialogs in a separate `xmlview`, make the following changes:

- You should no longer instantiate your dialogs with the `sap.ui.xmlview` syntax but with `sap.ui.xmlfragment`.
- If you need to instantiate your controller separately, first check if your controller code can be put inside the calling view's controller; otherwise, you just need to instantiate the controller separately with `sap.ui.controller("controllerName")` and pass it to your `xmlfragment` instantiation.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)

[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)

[DG_XML_NO_SINGLE_TAB \[page 220\]](#)

[XML_COMMONS_USAGE \[page 222\]](#)

[XML_DEPRECATION \[page 223\]](#)

[XML_FORM_USAGE \[page 225\]](#)

[XML_ICON_ACCESSIBILITY \[page 226\]](#)

[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)

[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)

[XML_LAYOUT_USAGE \[page 230\]](#)

[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)

[XML_MISSING_STABLE_ID \[page 232\]](#)

[XML_PAGE_ACCESSIBILITY \[page 233\]](#)

[XML_TABLE_ACCESSIBILITY \[page 234\]](#)

[XML_TITLE_ACCESSIBILITY \[page 235\]](#)

[XML_UPLOAD_IN_VIEW \[page 236\]](#)

[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.8 XML_FORM_USAGE

Checks the usage of the `sap.ui.commons.form` tags.

The `sap.ui.commons.form` object must not be used anymore, according to the *SAP Fiori Architectural Guidelines*.

Warning Message

Usage of `sap.ui.commons.form` is deprecated, please use `sap.ui.layout.form`

Rule Details

The check finds `sap.ui.commons.form` tags in the XML views.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.9 XML_ICON_ACCESSIBILITY

Checks accessibility for icons.

According to the *SAP Fiori Accessibility Guidelines*, icons need to be accessibility enabled.

Warning Message

An icon must define one of the following attributes: `tooltip`, `ariaLabelledBy`, `ariaDescribedBy`, `alt`

Rule Details

The rule checks whether each icon has a `tooltip`, `ariaLabelledBy`, `ariaDescribedBy` or `alt` attribute, and whether these are empty.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.10 XML_ICON_BUTTON_ACCESSIBILITY

Checks accessibility for icons.

According to the *SAP Fiori Accessibility Guidelines*, buttons need to be accessibility enabled.

Warning Message

An icon-only button must define the following attribute: tooltip

Rule Details

The rule checks whether each button has text and tooltip attributes and whether these are empty.

The following patterns are considered warnings:

```
<Button type="Back" press="onPress" />  
  <Button icon="sap-icon://action" press="onPress" />  
  <Button icon="sap-icon://action" press="onPress"  
  ariaLabelledBy="actionButtonLabel"/>
```

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.11 XML_IMAGE_ACCESSIBILITY

Checks accessibility for images.

According to the *SAP Fiori Accessibility Guidelines*, images need to be accessibility enabled.

Warning Message

An image must define one of the following attributes: `tooltip`, `ariaLabeledBy`, `ariaDescribedBy`, `alt`

Rule Details

The rule checks each image for `tooltip`, `ariaLabelledBy`, `ariaDescribedBy` or `alt` attribute. All tags that do not contain any of these attributes with a value will be found.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.12 XML_LAYOUT_USAGE

Detects the usage of the `sap.ui.commons.layout` tags.

According to the *SAP Fiori Architectural Guidelines*, usage of the `sap.ui.commons.layout` tag is not allowed. Instead, the `sap.ui.layout` tag should be used.

Warning Message

```
Usage of sap.ui.commons.layout is deprecated, please use sap.ui.layout
```

Rule Details

The rule checks whether a `sap.ui.commons.layout` tag is used in an XML view.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.13 XML_METADATA_MEDIA_SRC_WITHOUT_FORM ATTER

Checks usage of absolute `media_src` URLs.

If your SAP Fiori application includes attachments or other binary data like images, you should not use self-generated absolute URLs.

Warning Message

Use a formatter to generate absolute `__metadata/media_src` URLs

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)

[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)

[DG_XML_NO_SINGLE_TAB \[page 220\]](#)

[XML_COMMONS_USAGE \[page 222\]](#)

[XML_DEPRECATION \[page 223\]](#)

[XML_DIALOG_IN_VIEW \[page 224\]](#)

[XML_FORM_USAGE \[page 225\]](#)

[XML_ICON_ACCESSIBILITY \[page 226\]](#)

[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)

[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)

[XML_LAYOUT_USAGE \[page 230\]](#)

[XML_MISSING_STABLE_ID \[page 232\]](#)

[XML_PAGE_ACCESSIBILITY \[page 233\]](#)

[XML_TABLE_ACCESSIBILITY \[page 234\]](#)

[XML_TITLE_ACCESSIBILITY \[page 235\]](#)

[XML_UPLOAD_IN_VIEW \[page 236\]](#)

[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.14 XML_MISSING_STABLE_ID

Checks for stable IDs for controls.

If a control has no static ID defined in the XML view, during runtime no stable ID can be guaranteed. This can be a problem, for example, for in-application help or UI testing tools. See the *SAP Fiori Development Guidelines* for more information.

Warning Message

```
Consider setting an ID for this control
```

Rule Details

The check is implemented for the following controls:

sap.m

```
Label, Column, Button, ObjectAttribute, Toolbar, IconTabFilter, IconTabBar, SearchField, Page
```

sap.ui.comp.navpopover

```
SmartLink
```

sap.ui.comp.smartfield

```
SmartField, SmartLabel
```

sap.ui.comp.smartfilterbar

```
controlConfiguration, SmartFilterBar
```

sap.ui.comp.smartform

```
SmartForm
```

sap.ui.comp.smarttable

```
SmartTable
```

sap.ui.comp.smartvariants

```
SmartVariantManagement, SmartVariantManagementUi2
```

sap.ui.core

```
Item
```

sap.ui.layout

```
DynamicSideContent, FixFlex, Grid, GridData, GridIndent, GridPosition, GridSpan, HorizontalLayout, ResponsiveFlowLayout, ResponsiveFlowLayoutData, Splitter, SplitterLayoutData, VerticalLayout
```

sap.ui.layout.form

```
Form, FormContainer, FormElement, FormLayout, GridContainerData, GridElementCells, GridElementData, GridLayout, ResponsiveGridLayout, ResponsiveLayout, SimpleForm, SimpleFormLayout
```


Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.15 XML_PAGE_ACCESSIBILITY

Detects missing `title` attributes.

According to the *SAP Fiori Accessibility Guidelines*, pages need to be accessibility enabled.

Warning Message

A page must define a `title` attribute or a custom header with a `title` element within

Rule Details

The rule detects `Page` tags with missing `title` tags. All tags that do not contain a `title` attribute with a value or a `customHeader` tag with a `Title` element are found.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)

[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)

[DG_XML_NO_SINGLE_TAB \[page 220\]](#)

[XML_COMMONS_USAGE \[page 222\]](#)

[XML_DEPRECATION \[page 223\]](#)

[XML_DIALOG_IN_VIEW \[page 224\]](#)

[XML_FORM_USAGE \[page 225\]](#)

[XML_ICON_ACCESSIBILITY \[page 226\]](#)

[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)

[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)

[XML_LAYOUT_USAGE \[page 230\]](#)

[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)

[XML_MISSING_STABLE_ID \[page 232\]](#)

[XML_TABLE_ACCESSIBILITY \[page 234\]](#)

[XML_TITLE_ACCESSIBILITY \[page 235\]](#)

[XML_UPLOAD_IN_VIEW \[page 236\]](#)

[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.16 XML_TABLE_ACCESSIBILITY

Detects missing `title` attributes.

According to the *SAP Fiori Accessibility Guidelines*, a custom header needs to have a title.

Warning Message

A `table` must define a custom header with a `title` element within

Rule Details

The rule checks each `Table` tag for a `customHeader` tag with a `Title` tag. All tags that do not include a `customHeader` with a `Title` will be reported.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.17 XML_TITLE_ACCESSIBILITY

Detects missing `title` attributes.

According to the *SAP Fiori Accessibility Guidelines*, dialogs and simple forms need to be accessibility enabled.

Warning Message

A dialog or simple form must define the following attribute: `title`

Rule Details

The rule detects `Dialog` and `SimpleForm` tags with missing `title` tags.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.18 XML_UPLOAD_IN_VIEW

Checks usage of `FileUpload` and `AddPicture` tags.

An application might include a UI element that offers file upload functionality (`sap.ca.ui.FileUpload` or `sap.ca.ui.AddPicture`). To make sure that only allowed content can be uploaded, the file content must be checked by a virus scanner before it is stored on the database (details can be found in the security guidelines).

Warning Message

Uploaded files shall be sent to VSI 2.0 before stored on DB

Rule Details

The rule will find every `AddPicture` tag and `FileUpload` tag with the `uploadEnabled` attribute set to `true`.

How to fix

Determine which virus scan profile is used and include this information in the documentation. After sending an email containing the name of the application and the scan profile to `fiori-analysis-plugin@listserv.sap.corp`, an exemption will be created which will suppress this finding in the future.

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 237\]](#)

7.1.5.3.4.2.19 XML_BOOKMARK_PERFORMANCE

Checks your setting of the `serviceRefreshInterval` attribute.

When deciding which interval to use, keep in mind that there might be thousands of users who have the SAP Fiori launchpad open and might display some KPIs. A short refresh interval can create a considerable work load in the back end. Therefore we recommend the following default values depending on the use case:

- Complex calculations are required to calculate the data on the tile: This calculation might take several seconds and auto refresh must not be used. Set the interval to `0`.
- Only a simple query is required: For example, from one central table to determine the number of tasks assigned to a user. Set the interval to `300` (5 minutes).

Warning Message

A value of more than 0 and less than 300 for the property `serviceRefreshInterval` may result in performance limitations.

Rule Details

The following patterns are considered warnings:

```
<core:View xmlns:core="sap.ui.core" xmlns:ui="sap.ca.ui"
xmlns:suite="sap.suite.ui.commons" xmlns:layout="sap.ui.layout"
xmlns:html="http://www.w3.org/1999/xhtml"
xmlns="sap.m" xmlns:footerbar="sap.ushell.ui.footerbar"
controllerName="cus.crm.myaccounts.view.S360">
  <Page id="page" title="{i18n>DETAIL_TITLE}" showNavButton="true">
    <content>
      <layout:Grid class ="sapSuiteUtiHeaderGrid sapSuiteUti
sapCRMmyAccountsHeader" defaultSpan="L6 M6 S12" vSpacing="0" >
        <layout:content>
          ...
          <footerbar:AddBookmarkButton serviceRefreshInterval="20" />
        </layout:content>
      </layout:Grid>
    </content>
  </Page>
</core:View>
```

Parent topic: [SAP Fiori XML Validator Rules \[page 215\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 216\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 217\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 219\]](#)
[DG_XML_NO_SINGLE_TAB \[page 220\]](#)
[XML_COMMONS_USAGE \[page 222\]](#)
[XML_DEPRECATION \[page 223\]](#)
[XML_DIALOG_IN_VIEW \[page 224\]](#)
[XML_FORM_USAGE \[page 225\]](#)
[XML_ICON_ACCESSIBILITY \[page 226\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 227\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 229\]](#)
[XML_LAYOUT_USAGE \[page 230\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 231\]](#)
[XML_MISSING_STABLE_ID \[page 232\]](#)
[XML_PAGE_ACCESSIBILITY \[page 233\]](#)
[XML_TABLE_ACCESSIBILITY \[page 234\]](#)
[XML_TITLE_ACCESSIBILITY \[page 235\]](#)
[XML_UPLOAD_IN_VIEW \[page 236\]](#)

7.1.5.3.5 Validation of neo-app.json Files

A project's neo-app.json file is validated on opening in the code editor.

When you open or make changes to your project's neo-app.json application descriptor file in SAP Web IDE, there is automatic schema validation, and errors are displayed as annotations.

For more information about the required format of the neo-app.json file, see [Application Descriptor File](#).

Parent topic: [Checking Code \[page 155\]](#)

Related Information

[Configure Code Checking \[page 156\]](#)

[Code Checking Annotations \[page 157\]](#)

[JavaScript Validation \[page 159\]](#)

[XML Validation \[page 213\]](#)

[Validation of manifest.json Files \[page 239\]](#)

[Validation of Stable IDs \[page 240\]](#)

[Validation of i18n Files \[page 241\]](#)

[Using the Problems View \[page 242\]](#)

7.1.5.3.6 Validation of manifest.json Files

Application descriptor files (manifest.json) for SAP Fiori projects are validated for syntax and schema issues.

Application descriptor files (manifest.json) are validated for issues in syntax and schema. Detected issues are displayed in the *Problems* view, even for files that are not displayed in the code editor. You can filter the list of problems by using manifest.json in the *Location* or *File* columns.

The following validation is performed:

- Validation of a json file syntax. Code issues are displayed as annotations as for JavaScript file code checking.
- After the confirmation of a valid syntax, a schema validation is performed.
 - Schema errors are indicated by a gutter icon in the line of code in which the error occurs and a description of the error is displayed as an inline annotation.
 - Errors for missing fields are displayed as an annotation at the first line of code.

i Note

Schema validation is performed according to a predefined validator, which is not configurable.

Parent topic: [Checking Code \[page 155\]](#)

Related Information

- [Configure Code Checking \[page 156\]](#)
- [Code Checking Annotations \[page 157\]](#)
- [JavaScript Validation \[page 159\]](#)
- [XML Validation \[page 213\]](#)
- [Validation of neo-app.json Files \[page 239\]](#)
- [Validation of Stable IDs \[page 240\]](#)
- [Validation of i18n Files \[page 241\]](#)
- [Using the Problems View \[page 242\]](#)

7.1.5.3.7 Validation of Stable IDs

Check that stable IDs exist in your `manifest.json` file and your `view.xml` file.

You can personalize object pages when running an application in SAP Fiori launchpad. Key users can also easily make UI changes for all users of an application. Since stable IDs are used to identify the controls that a key user can adapt using the key user adaptation feature, you must use stable IDs for all controls and views of your application. For more information, see [Stable IDs: All You Need to Know](#).

As of version 1.12.0 of the application descriptor schema, the `flexEnabled` property in the `sap.ui5` section of the `manifest.json` indicates that the application supports key user and developer adaptation. If you added the `flexEnabled` property, SAP Web IDE validates your IDs. If it finds unstable IDs, you can either provide stable IDs manually or SAP Web IDE can generate them for you.

1. From the workspace, open your application's `manifest.json` file and make sure that its version is at least 1.12.0.

i Note

If you need to change the version, you must restart SAP Web IDE to get a proper validation.

2. Search for the "sap.ui5" property and add the "flexEnabled": true property.
In the `view.xml` file, element IDs are validated.
In the `manifest.json` file, the following properties are validated:
 - The "id" property in the `sap.ui5 > rootView` section.
 - The "viewId" property in the `sap.ui5 > routing > targets` section.
3. In the Problems view, right-click the missing ID errors to generate stable IDs automatically. You can also provide the missing stable IDs manually in the [Code Editor](#).

For more information, see [SAPUI5 Flexibility Services: Adapting UIs Made Easy](#) and [UI Adaptation at Runtime: Enable Your App](#) in the SAPUI5 Demo Kit.

Parent topic: [Checking Code \[page 155\]](#)

Related Information

[Configure Code Checking \[page 156\]](#)

[Code Checking Annotations \[page 157\]](#)

[JavaScript Validation \[page 159\]](#)

[XML Validation \[page 213\]](#)

[Validation of neo-app.json Files \[page 239\]](#)

[Validation of manifest.json Files \[page 239\]](#)

[Validation of i18n Files \[page 241\]](#)

[Using the Problems View \[page 242\]](#)

7.1.5.3.8 Validation of i18n Files

Your i18n files are validated in the SAP Web IDE code editor according to the SAP Fiori best practices.

The i18n file is used by translators to convert strings to other languages. The information provided must follow a specific pattern and include all the relevant information. For more information, see [Context Information for Translators](#).

i18n strings should be written according to the following pattern:

```
#Text type: Note for translator.
```

Where text type indicates where the string will appear, for example: #XMSG, #XBUT, or #XFLD".

In the code editor, when an i18n file is opened or changed and it contains problematic lines, these will be indicated in the file and in the Problems view.

i Note

If you open the entire project and not the specific file, the errors will not appear in the Problems view.

- Error - An error notification is displayed if you enter a key but do not provide a value.
- Warning - A warning notification is displayed if you enter a key value that already exists in the file.
- Info - A general information notification is displayed if an entry does not have a comment for the translators or has the wrong category.

Parent topic: [Checking Code \[page 155\]](#)

Related Information

[Configure Code Checking \[page 156\]](#)

[Code Checking Annotations \[page 157\]](#)

[JavaScript Validation \[page 159\]](#)

[XML Validation \[page 213\]](#)

[Validation of neo-app.json Files \[page 239\]](#)

[Validation of manifest.json Files \[page 239\]](#)

[Validation of Stable IDs \[page 240\]](#)

[Using the Problems View \[page 242\]](#)

7.1.5.3.9 Using the Problems View

View information about problems in the projects in your workspace.

i Note

This feature is not available in SAP Web IDE personal edition.

The *Problems* view displays information about problems in the projects in your workspace for the following files:

- JavaScript files - syntax errors and ESLint validation errors
- XML files - syntax errors and semantic SAPUI5 errors, including deprecation warnings
- JSON files - syntax errors, manifest and neo-app problems

If you select a project when the *Problems* view is open, SAP Web IDE automatically analyzes the project for problems. Alternatively, you can choose to analyze the entire workspace.

1. In the workspace, select a single project, or select the *Workspace* folder to analyze the entire workspace.
2. In the *Problems* view, select *Analyze and Display* to trigger an analysis of the selected files and display the problems.
3. Sort the list of problems by clicking any column header. Filter the list by clicking on a column header and entering a filter string.

The *Problems* view displays the following information:

- The total error count and the error count for each severity for all the analyzed files. The error count does not change when you filter the display or change the severity selection.
- The scope of the analysis, which can be a selected project or the entire workspace.
- The severity of each problem. You can filter the list according to severity.
- The problem description with a link to more information.
- A link to the file that contains the problem. Click the link to open the file in the workspace at the location of the problem.
- The full path to the folder that contains the file with the problem.
- The problem category.

You can use the AutoFix feature to correct errors and warnings in your code in Java or JavaScript files in the *Problems* view:

1. In the *Problems* view, right-click an error or warning for a fixable problem.
2. Choose the indicated correction in the context menu.
The code is automatically corrected.

The *Problems* view is dynamically updated when you edit, add, and delete files or delete a project within the scope of the analysis.

You can exclude specific files or folders from the analysis:

1. Create a text file with the suffix `eslintignore` and put it under the root project folder.
2. In the file, enter the names of all the files and folders to ignore. Each entry should be on a separate line.
The next time that you trigger an analysis, these files and folders will be excluded from the analysis.

Parent topic: [Checking Code \[page 155\]](#)

Related Information

[Configure Code Checking \[page 156\]](#)

[Code Checking Annotations \[page 157\]](#)

[JavaScript Validation \[page 159\]](#)

[XML Validation \[page 213\]](#)

[Validation of neo-app.json Files \[page 239\]](#)

[Validation of manifest.json Files \[page 239\]](#)

[Validation of Stable IDs \[page 240\]](#)

[Validation of i18n Files \[page 241\]](#)

7.1.5.4 Developing Apps Using SAP Fiori Elements

This section contains information about creating SAP Fiori apps using SAP Fiori elements in SAP Web IDE.

SAP Fiori elements provide designs for UI patterns and predefined templates for commonly used application patterns. You can use SAP Fiori elements to create SAP Fiori applications based on OData services and annotations requiring no JavaScript UI coding. The resulting app uses predefined views and controllers that are provided centrally, so no application-specific view instances are required. The SAPUI5 runtime interprets the metadata and annotations of the underlying OData service and uses the corresponding views while starting the SAP Fiori application.

The predefined views and controllers ensure UI design consistency across similar apps. The metadata-driven development model also significantly reduces the amount of front-end code for each app. This means that the developer can focus on the business logic.

By using the wizard for creating projects in SAP Web IDE, you can generate an SAP Fiori app based on SAP Fiori elements. The following SAP Fiori elements are available:

- [List Report \[page 244\]](#)
You can use the list report template to work with a large list of items. It combines powerful functions for filtering large lists with different ways of displaying the resulting item list.
- [Object Page \[page 244\]](#)
On the object page you can display, edit, and create objects, as well as save drafts. It is suitable for both simple objects and more complex, multi-faceted objects. The object page view offers the best possible support for multiple devices.
- [Analytical List Page \[page 247\]](#)

Analytical list page is an SAP Fiori application for performing detailed analytics. Use this template to build a landing page for your SAP Fiori applications. It helps you to quickly and easily combine transactional and analytical data in the form of chart or table visualizations.

- [Overview Page \[page 247\]](#)

An overview page is a data-driven SAP Fiori app based on SAPUI5 technology for organizing large amounts of information. Information is visualized in a card format, different cards for different types of content, in an attractive and efficient way. The user-friendly experience makes viewing, filtering, and acting upon data quick and simple. While simultaneously presenting the big picture at a glance, business users can focus on the most important tasks enabling faster decision making as well as immediate action.

For more information, see [Developing Apps with SAP Fiori Elements](#).

7.1.5.4.1 Prerequisites

Provides a list of the prerequisites you must meet before using the SAP Fiori Fiori Elements application.

- You have subscribed to SAP Web IDE and completed all configuration steps, including Git settings, destinations, and connection to remote systems.
For more information, see [Getting Started \[page 53\]](#).
- You have all of the OData services and annotations.
For more information, see [Preparing OData Services](#).
- You have defined the backend that you are using as a destination in SAP BTP, so that the OData services are available in SAP Web IDE.
- For more information, see [Destinations](#) and [Cloud Connector](#).

7.1.5.4.2 Create List Report and Object Page Applications

You use SAP Web IDE to create an SAP Fiori app containing a list report and object pages.

Context

As an app developer, you must define a configuration in the SAP Web IDE. The main aspects are the destinations to the back-end metadata, navigation between pages, and page design (as pages may contain several templates).

i Note

This step-by-step procedure is also available as a video: .

Procedure

1. In the SAP Web IDE, from the *File* menu, choose **► New ► Project from Template ►**. The system starts the wizard for new projects.
2. Follow the guided procedure:

Step	Action
1. Template Selection	Select <i>List Report Page</i> and click <i>Next</i> .
2. Basic Information	The following fields are mandatory: <ul style="list-style-type: none">○ <i>Project Name</i>○ <i>Title</i>○ <i>Application Component Hierarchy</i> Choose <i>Next</i> .
3. Data Connection	<ol style="list-style-type: none">1. Choose <i>Service Catalog</i> and select the desired data source from the list.2. Choose a service and then choose <i>Next</i>.
4. Annotation Selection	Select the required annotation file and then choose <i>Next</i> .
5. Template Customization	<ol style="list-style-type: none">1. Under <i>Data Binding</i>, complete the fields like this:<ul style="list-style-type: none">○ <i>OData Collection</i> is a mandatory field○ <i>OData Navigation</i> is where you select the relevant navigation option○ <i>OData Sub Navigation</i> is where you select an additional OData sub-navigation target. You can also decide whether you want your app to include flexible column layout and/or variant management. <i>Page Variant</i> is set as the default with an option to change to <i>No Variant</i> and <i>Control Level Variant</i>.2. Choose <i>Next</i> and <i>Finish</i>.

3. Open your project (already selected in project list).
4. Open the *webapp* folder.
5. Select *Component.js* and choose *Run*.

If you get the message that variants can't be loaded, choose *OK* to continue.

7.1.5.4.3 Create a Worklist Application

You use SAP Web IDE to create an SAP Fiori app containing a worklist.

Context

Procedure

1. In the SAP Web IDE, from the *File* menu, choose ► *New* ► *Project from Template* ►. The system starts the wizard for new projects.
2. Follow the guided procedure:

Step	Action
1. Template Selection	Select <i>Worklist</i> and click <i>Next</i> .
2. Basic Information	The following fields are mandatory: <ul style="list-style-type: none">○ <i>Project Name</i>○ <i>Title</i>○ <i>Application Component Hierarchy</i> Choose <i>Next</i> .
3. Data Connection	<ol style="list-style-type: none">1. Choose <i>Service Catalog</i> and select the desired data source from the list.2. Choose a service and then choose <i>Next</i>.
4. Annotation Selection	Select the required annotation file and then choose <i>Next</i> .
5. Template Customization	<ol style="list-style-type: none">1. Under <i>Data Binding</i>, complete the fields as follows:<ul style="list-style-type: none">○ <i>OData Collection</i> – this is a mandatory field○ <i>OData Navigation</i> – select the relevant navigation option2. Choose <i>Next</i> and <i>Finish</i>.

3. Open your project (already selected in project list).
4. Open the *webapp* folder.
5. Select *Component.js* and choose *Run*.

7.1.5.4.4 Create an Analytical List Page Application

You can create your own analytical list page applications using the Analytical List Page plugin in SAP Web IDE.

Procedure

1. Click **File > New > Project from Template**.
2. Select **Category > SAP Fiori Elements > Analytical List Page**.
3. Enter a *Project Name*, and fill in the *App Descriptor Data*.
4. Select a data source and service from the list.
5. (Optional) Add annotation files.

If you add more than one annotation file, you can determine files based on the order in which they are loaded.

i Note

If you have selected a service that does not have an annotation associated with it, the wizard generates the annotation for you.

6. Select *OData Collection* for data binding. (Optional) Enter the required *App Descriptor Settings*, or you can modify this later in the App Descriptor file.
7. Confirm the project information and choose *Finish*.
8. Build and run your application.
 1. Open your project (already selected in the project list).
 2. Open the *webapp* folder.
 3. Select *Component.js* and choose *Run*.

7.1.5.4.5 Create an Overview Page Application

You can create your own overview page applications using the Overview Page plugin in SAP Web IDE.

Procedure

1. Click **File > New > Project from Template**.
2. Select **Category > SAP Fiori Elements > Overview Page Application**.
3. Enter *Project Name*, and fill in the *App Descriptor Data*.
4. Select a data source and service from the list.
5. (Optional) Add annotation files.

If you add more than one annotation file, you can determine files based on the order in which they are loaded.

6. Fill in the required overview page details, paying attention to the following:
 - The default value of the Datasource Alias is the name of the data service selected in the *Data Connection* step.
 - Users filter content on the overview page according to the fields defined in the selected *EntityType*.
7. Confirm the project information and choose *Finish*.

Results

The wizard creates the project structure in the workspace under a new folder with the project name that you specified.

Related Information

[Add Cards to an Overview Page \[page 248\]](#)

7.1.5.4.5.1 Add Cards to an Overview Page

Add cards to populate the overview page that you created.

Procedure

1. In the *Development* tab, select the overview page project that you created, and choose **File > New > Card**.

Note

If you have created a Multi-Target Application, select the Multi-Target Application project that you created, and choose **File > New > Card**.

2. Select an existing data source, or create a new data source for the card.
3. Select one of the following card types:

Option	Description
List Card	Displays an array of items in a vertical list. A number of list types are available.
Link List Card	Displays an array of items in a vertical list with title, picture, icon, or subtitle.
Table Card	Displays items in a three-column table.

Option	Description
Stack Card	A collection of single-object cards. When opened, users can take action on the individual items in the stack.
Analytic Chart Card	These type of chart cards show data in a variety of formats. For example, they can be cards that display data in a series of data points connected by straight lines, that use bubbles to visualize the data dimension, or in columns or stacked columns to help view multiple measures or dimensions.

i Note

Overview Page lets you configure view switch and KPI header section for List, Table, and Analytic Chart cards. Selecting the checkbox:

- [Select to enable view switch for this card](#) lets you configure multiple views, apply different filtering, and sorting options in the card.
- [Select to add KPI header for this card](#) lets you configure KPI header information in the card. If you are using an SAP Smart Business Modeler Apps for SAP S4HANA, click [Next](#) and choose a KPI annotation to configure KPI information.


For more information about the different card types, see [Cards Used in Overview Pages](#).

4. Different card types require different configuration details. Fill in the required details for the selected card type.
5. Choose [Finish](#) to complete the wizard.
6. Build and run your application.
 - Open your project (already selected in project list).
 - Open the [webapp](#) folder.
 - Select [Component.js](#) and choose [Run](#).

7.1.5.4.6 Using the Extension Wizard

For list reports, object pages, and analytical list pages, you can use the extension wizard in the SAP Web IDE to create app extensions.

Procedure

1. In the SAP Web IDE, select your generated app, and choose [File](#) > [New](#) > [Extension](#) . The system starts the extension wizard.
2. Select your SAP Fiori elements template and follow the guided procedure.

7.1.5.5 Using the Outline Pane for JavaScript Files

The Outline pane helps you to understand the structure of JavaScript files and to navigate through the code.

Context

The Outline pane provides a clear view of the main entities and coding structure of JavaScript files by displaying the full hierarchy of a file's functions and objects.

When you open the Outline pane, it expands to the first level of the hierarchy. Functions and objects are denoted by different icons. You can expand and collapse the nodes to explore nested functions and objects.

- Function nodes display the function name and parameters, if they exist, and function assignment names of anonymous functions assigned to variables. Other unnamed functions are displayed with the label `function()`.
- Object nodes display the object name. Unnamed objects are displayed with the label `object{}`.

When you click a node in the Outline pane, the corresponding line or block is selected in the Code Editor pane, enabling you to navigate quickly to the desired object and function code.

The Outline pane refreshes when:

- You open a JavaScript file
- You move between JavaScript file Code Editor panes
- You save changes to an open JavaScript file

If there is a fatal syntax error in the JavaScript code, the Outline pane displays up to the location of the error in the code.

When you move from a JavaScript file Code Editor pane to a Code Editor pane that does not support the outline feature, the Outline pane is cleared.


Note

- You can view the outline of files that contain a maximum of 130,000 characters.
- When a function is a parameter in a constructor, for example, `var x = new Object(param1, function() {});`, it is not displayed as a separate node in the outline.
- Array content is not displayed in the outline.
- Static variable definition is supported; however, dynamic assignment is not supported. In the following example, `a` appears as an object in the outline:

```
var a = x;  
var x = function() {...};
```

Procedure

To open the Outline pane for a specific JavaScript file:



1. Ensure that the required JavaScript file Code Editor pane is in focus.
2. In the right sidebar, choose  (Outline).
The Outline pane opens displaying the outline expanded to the first level.

7.1.5.6 Creating an HTML5 Application Descriptor File

Create an HTML5 Application Descriptor file in a project that has not been imported or created via the project wizards. This is required for defining cloud connectivity for external resources required by the application, and allows you to run the application properly in the SAP Web IDE.

Procedure

In SAP Web IDE:

1. Select your project folder (for example, *myProject*).
2. From the *File* menu, select  *New* > *HTML5 Application Descriptor* .
3. Enter routes for paths */resources* and */test-resources* with service *sapui5* in your *neo-app.json* file:

```
{
  "path": "/resources",
  "target": {
    "type": "service",
    "name": "sapui5",
    "entryPath": "/resources"
  },
  "description": "SAPUI5 Resources"
},
{
  "path": "/test-resources",
  "target": {
    "type": "service",
    "name": "sapui5",
    "entryPath": "/test-resources"
  },
  "description": "SAPUI5 Test Resources"
}
```

For more information, see [Application Descriptor File](#).

4. Enter routes for your remote systems. For example, you can add an ABAP OData system for the path */sap/opu/odata* and a destination *myRemoteDestination*. For connecting remote systems, see [Connect to ABAP Systems \[page 57\]](#).

```
{
  "path": "/sap/opu/odata",
  "target": {
    "type": "destination",
    "name": "myRemoteDestination",
    "entryPath": "/sap/opu/odata"
  },
  "description": "Target OData system"
}
```

7.1.6 Developing UI Libraries

You can create, build, deploy and reuse UI libraries.

Context

UI libraries enable you to share custom controls between SAPUI5 projects in SAP Web IDE.

UI libraries are created including a sample control and Grunt build capabilities based on the [UI5 Build and Development Tooling](#) .

Procedure

1. Create a new project using the *SAP Fiori Library* template. The library project is added to your workspace.
2. Select the project, right-click, and select **► New ► Control ►** to add a custom control to your project using the wizard.
3. Select the project, right-click, and select **► New ► Control from Existing Library ►** to add a custom control to your project based on an already existing control.
 - a. In the *New Control from Existing Library* wizard, select the repository containing the desired control.
 - b. Expand the library containing the desired control.
 - c. Select the *Include* checkbox for the desired control.
 - d. Provide a name for the control.
 - e. You can run unit tests for the new controls.
4. Create a new SAPUI5 project.

i Note

Optionally, you can use an existing project in your workspace.

5. Right-click on the project and select **► Project ► Add Reference to Library ►**.
6. In the *Add Reference to Library* dialog box, select the repository containing the desired library.
7. Select the checkbox of the libraries you want to refer to.
8. Click *Add*. The SAPUI5 project is updated with a dependency to the selected library. This allows you to use any control included in the reused library.
9. Run the project.

If you are referring to already deployed libraries there are no further steps. If you want to run the libraries that are located in your workspace, do the following:

- a. Select the project.
- b. Right-click and select *Run Configurations*.
- c. Create a new run configuration for your web application.
- d. In the *URL Components* tab, add the `sap-ui-xx-lesssupport` parameter and set its value to `true`.

- e. In the *Advanced Settings* tab, under the *Application Resources* section, select the *Use my workspace first* checkbox.
- f. Save and run your project.

7.1.7 Developing Application Tests

Develop tests for application functionality as you develop your application.

You can develop and run OPA and QUnit tests to test your application before submitting it for a build.

- Many of the templates include a set of tests under the project's `test` folder. You can modify these tests and add new tests.
- You can add new OPA and QUnit tests in a project using the New Test wizard.
- You can use code completion to add code snippets for OPA actions, assertions, and tests, and for QUnit tests and modules.

You can run OPA and QUnit tests using the run configurations that are provided with the template or you can create new run configurations..

For more information about OPA testing, see [One Page Acceptance Tests \(OPA5\)](#)

For more information about QUnit testing, see [QUnit Testing Fundamentals](#)

7.1.7.1 Create Test Structures

You can add the best practice test structure (including sample tests) to your project.

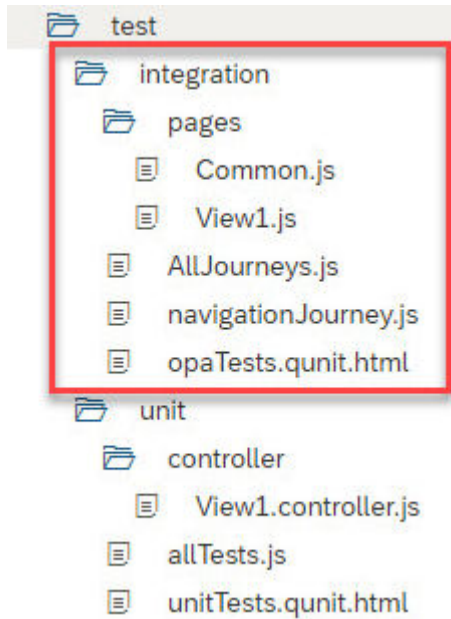
Prerequisites

- The project must have been created using the SAPUI5 Application template.
- The project must not have any pre-existing tests.

Procedure

1. Right-click on the project and select **New > Test Structure**.
Under the `webapp` folder, a new folder called `test` is generated containing two subfolders: `integration` and `unit`.

`integration` **subfolder**



The `integration` subfolder is meant for OPA5 tests. It contains the following files:

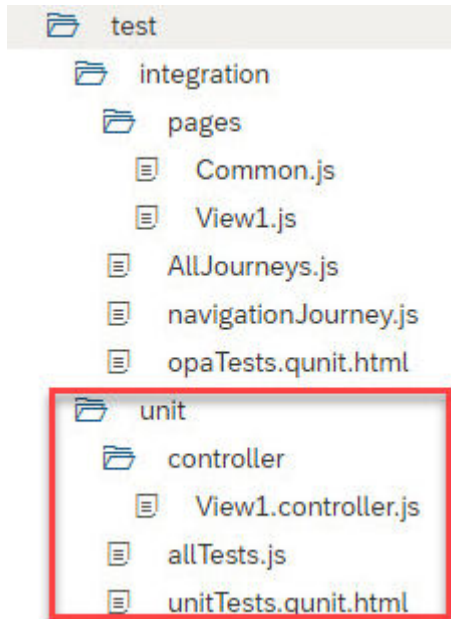
- `Alljourney.js` - contains all the OPA5 test files you want to run.
- `Common.js` - contains the OPA5 tests assertions and actions relevant for all views.
- `<view name>.js` - contains the OPA5 tests assertions and actions relevant for the specific view. The file contains two assertion examples. The `iDoMyAssertion` will fail, and `iShouldSeeTheApp` should pass.

You might want to add additional tests to this view, in which case you can do so in this file, or you might want to add additional tests to a different view, in which case you should create another file with the view name.

- `navigationJourney.js` - contains all test cases for a specific view or use case, for example the navigation journey simulates user interaction with the app.
- `opaTests.qunit.html` - manually starts all integration tests.

For more information, see [Integration Testing with One Page Acceptance Tests \(OPA5\)](#) and [A First OPA Test](#).

`unit` **subfolder**



The unit subfolder is for the unit tests. It contains the following files:

- `allTests.js` - contains all the QUnit test files you want to run.
- `<view name>.controller.js` - contains the QUnits relevant for this specific controller.
The file contains two examples. The `I should test the app controller loads test` will succeed whereas the `I should test any additional controls` will fail.
You might want to add additional tests to this controller, in which case you can do so in this file, or you might want to add additional tests to a different controller, in which case you should create another file with the controller name.
- `unitTests.qunit.html` - manually starts all unit tests.

2. Run integration tests.
 - a. Select `integration > opaTests.qunit.html`.
 - b. Right-click and select `Run > Run as Unit Test`.
3. Run unit tests.
 - a. Select `unit > unitTests.qunit.html`.
 - b. Right-click and select `Run > Run as Unit Test`.

Run QUnit and OPA5 Tests Locally via Grunt

Procedure

1. Export you project.
2. Unzip the project to the desired location in your computer.
3. Open your command shell/terminal and navigate to the extracted project's location.
4. In the command shell/terminal, enter `npm install`.

5. In the command shell/terminal, enter `"npm test"`.

7.1.7.2 Create Tests

You can create tests in your project using a wizard.

Context

You can now create OPA pages, OPA journeys, and QUnit tests in a project using a wizard.

Procedure

1. Choose **File > New** and one of the following:
 - *OPA Page*
 - *OPA Journey*
 - *QUnit Test*

The new test wizard opens.

2. Enter the test file name and, optionally, choose the test container. Click *Next* to progress to the next step.
3. For a new OPA page, in the *SAPUI5 Control Selection* page, choose the required view and then choose the controls that you want to test.

The table displays all the controls that are in the selected view or any of its fragments that are relevant for an OPA test.

4. Click *Next* or *Finish*.

Results

The test is created and added to the corresponding test container file in the project.

7.1.8 Layout Editor

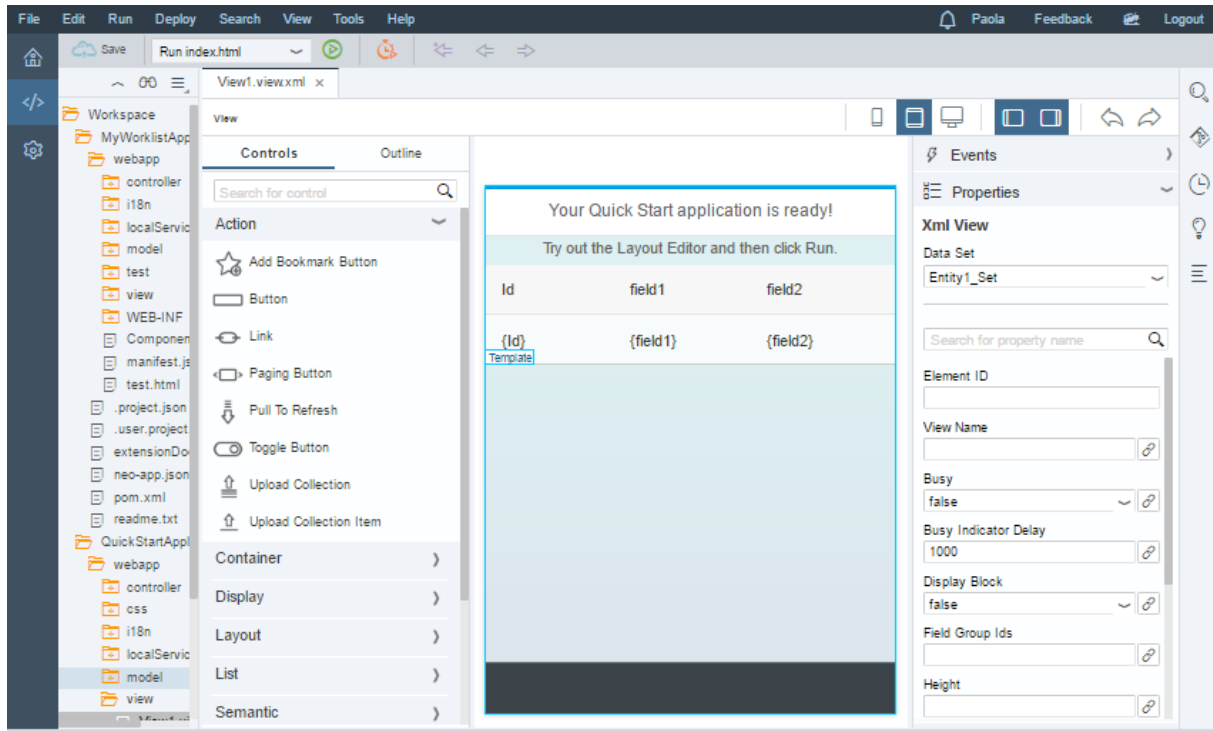
Display the content of an XML view in the layout editor to see it in a way that closely corresponds to how it will appear in your finished application.

i Note

The layout editor is not supported in the Safari browser.

Layout Editor Landscape

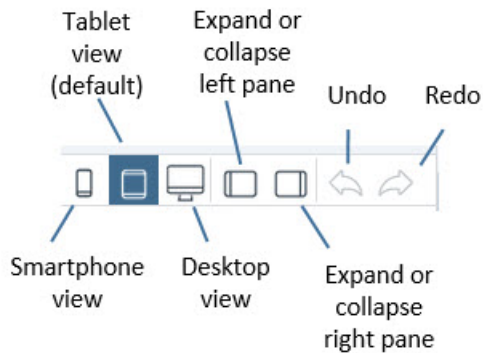
The layout editor is composed of a canvas, a pane on the left that includes the *Controls* and *Outline* tabs, and a pane on the right that includes the *Events* and *Properties* panes.



Toolbar

The buttons on the layout editor toolbar allow you to:

- Change the device format of the canvas to smartphone, tablet, or desktop view.
- Change the application's SAPUI5 version.
- Expand and collapse the panes to the right and left of the canvas.
 - The pane on the left side includes the *Controls* and *Outline* tabs.
 - The pane on the right side includes the *Properties* and *Events* panes.
- Undo and redo actions.



Controls Tab

You can expand or collapse each section by clicking the arrow on each section header. You can also search for controls by entering the control name in the search field at the top of the *Controls* tab. The relevant sections expand to display the controls that match the search criteria.

Note

Make sure to delete the search criteria if you want to expand other sections.

You can drag and drop controls from the *Controls* tab onto the canvas. For more information, see [Add Controls from the Controls Tab \[page 266\]](#).

Controls	Outline
Search for control <input type="text"/>	
Action	>
Container	>
Display	>
Layout	>
List	>
Tile	>
User Input	>

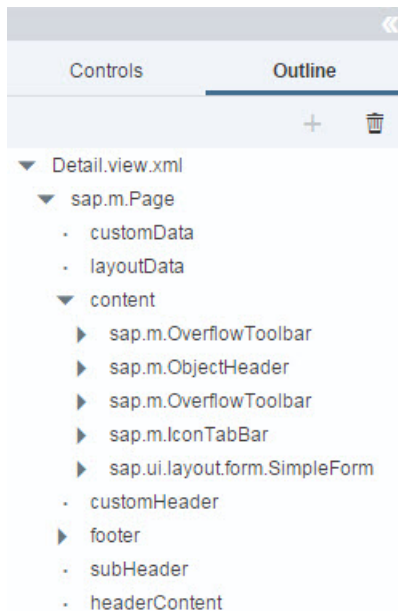
You can find the list of available controls in [SAPUI5 Controls Supported in the Layout Editor \[page 275\]](#).

Outline Tab

Controls that are selected on the *Outline* tab are automatically selected on the canvas and vice versa.

You can use the *Outline* tab to see the hierarchy of controls on the canvas. In addition, you can add and remove controls from the canvas using the *Outline* tab.

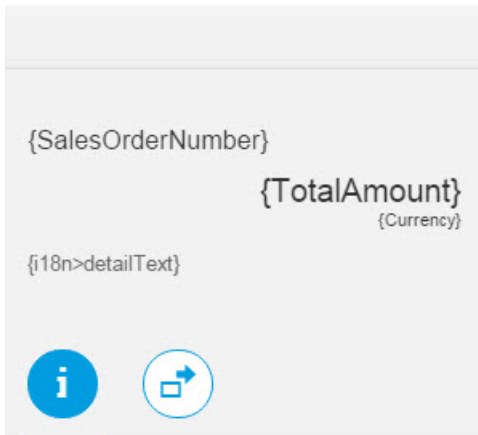
For more information, see [Add Controls from the Outline Tab \[page 263\]](#).



Canvas

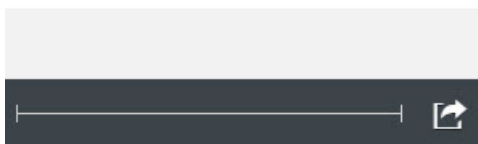
The canvas in the middle of the layout editor area provides a graphical display of the selected XML view.

Click a control on the canvas to select it. Click again to select its parent control. You can keep clicking until you reach the highest control in the hierarchy and then the focus will return to the original control. Click outside the canvas to undo the selection.



NetPriceAmount:
{NetPriceAmount}

TaxAmount:
{TaxAmount}
{OrderDate}

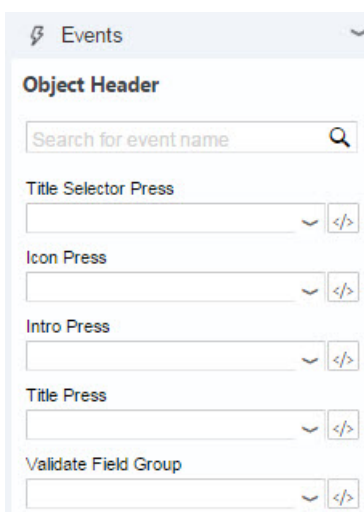


Events and Properties Pane


On the right side of the canvas is a pane that displays the following panes:

Events Pane

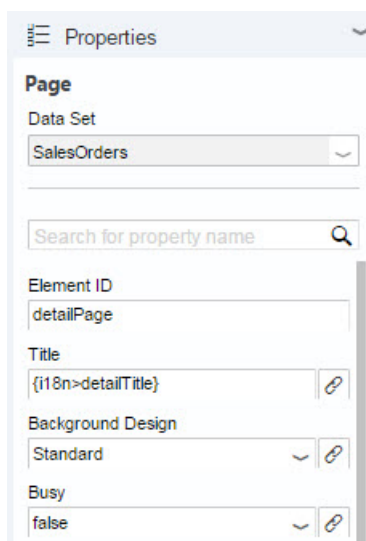
The *Events* pane allows you to select an existing event handler from the controller for an event of the selected control. The `</>` icon next to each event opens the code editor to display the relevant controller in the XML code.



Properties Pane

The *Properties* pane shows the properties of the control that is currently selected in the canvas and allows you to modify its property values. The most commonly used properties for each control are displayed at the top of the list. The  icon next to each property opens the *Data Binding* dialog box.

For more information, see [Binding Data \[page 268\]](#), [Bind Data to a Simple Control \[page 271\]](#), and [Bind Data to an Aggregate-Type Control \[page 271\]](#).



Note

Deprecated properties or aggregations are marked with the label *deprecated* (also in the *Outline* tab). For more information, see [SAP Library for User Interface Add-On 1.0 for SAP NetWeaver](http://help.sap.com/nw-uiaddon) on SAP Help Portal at <http://help.sap.com/nw-uiaddon>. Under *Application Help*, open *SAP Library*, and search for *deprecation*.

[Working with the Layout Editor \[page 262\]](#)

An overview of the steps required to edit a project using the layout editor.

[Creating a Quick Start Application with the Layout Editor \[page 274\]](#)

Quickly create a new application using the layout editor.

[SAPUI5 Controls Supported in the Layout Editor \[page 275\]](#)

Provides a list of SAPUI5 controls that are supported in the layout editor.

[Try It: Build an Application with the Layout Editor \[page 293\]](#)

Get an overview of the features that are available with the layout editor by following this tutorial for building an application.

Related Information

[Working with the Layout Editor \[page 262\]](#)

[SAPUI5 Controls Supported in the Layout Editor \[page 275\]](#)

7.1.8.1 Working with the Layout Editor

An overview of the steps required to edit a project using the layout editor.

Prerequisites

- You have opened SAP Web IDE in Google Chrome or Internet Explorer 11.
- You have created a project using the *Fiori Master-Detail Application* template. In the `view` folder of this project, you can edit the following views:
 - `Detail.view.xml`
 - `Master.view.xml`

Procedure

1. From the context menu of the XML view, choose **► Open with ► Layout Editor ►**.
2. Edit the views of your application as follows:
 - Add controls to your view using the drag and drop functionality.
 - Delete controls from your view.
 - Rearrange controls in your view using the drag and drop functionality.
 - Use the keyboard to navigate within the canvas. Double-click on a control to move the selection to its parent.
 - Bind controls in the layout editor to elements from the OData service.
 - Extract a control to a fragment using the context menu.

[Add Controls from the Outline Tab \[page 263\]](#)

You can add controls to the canvas from the *Outline* tab.

[Delete Controls from the Outline Tab \[page 263\]](#)

You can remove controls from the *Outline* tab.

[Create a New Function \[page 264\]](#)

You can create a new function for a controller from the *Events* pane.

[Create a Navigation Between Views \[page 265\]](#)

You can create a navigation between views using the layout editor.

[Add Controls from the Controls Tab \[page 266\]](#)

Add controls to the canvas by using drag and drop functionality.

[Keyboard Support \[page 267\]](#)

Use the keyboard to move selected controls or navigate within the view that you opened with the layout editor.


[Layout Editor Binding Capabilities \[page 268\]](#)

In the layout editor, you can bind properties of controls or control aggregations to data fields, `i18n` models, and label annotations.

7.1.8.1.1 Add Controls from the Outline Tab

You can add controls to the canvas from the *Outline* tab.

Procedure

1. On the *Outline* tab, select a control to which you want to add another control.
2. At the top of the *Outline* tab, click the *Add*  button, then in the popup menu, select the control you want to add.

The control is added on the *Outline* tab and appears on the canvas.

Note

The information bar at the top of the canvas shows you where you are about to drop the control.

Task overview: [Working with the Layout Editor \[page 262\]](#)

Related Information

[Delete Controls from the Outline Tab \[page 263\]](#)

[Create a New Function \[page 264\]](#)

[Create a Navigation Between Views \[page 265\]](#)

[Add Controls from the Controls Tab \[page 266\]](#)

[Keyboard Support \[page 267\]](#)

[Layout Editor Binding Capabilities \[page 268\]](#)


[Delete Controls from the Outline Tab \[page 263\]](#)

[SAPUI5 Controls Supported in the Layout Editor \[page 275\]](#)

7.1.8.1.2 Delete Controls from the Outline Tab

You can remove controls from the *Outline* tab.

Procedure

1. On the *Outline* tab, select a control that you want to delete.
2. At the top of the *Outline* tab, click the *Delete*  button.

The control is removed from the view.

Task overview: [Working with the Layout Editor \[page 262\]](#)

Related Information

[Add Controls from the Outline Tab \[page 263\]](#)

[Create a New Function \[page 264\]](#)

[Create a Navigation Between Views \[page 265\]](#)

[Add Controls from the Controls Tab \[page 266\]](#)

[Keyboard Support \[page 267\]](#)

[Layout Editor Binding Capabilities \[page 268\]](#)

[Add Controls from the Outline Tab \[page 263\]](#)

[SAPUI5 Controls Supported in the Layout Editor \[page 275\]](#)

7.1.8.1.3 Create a New Function

You can create a new function for a controller from the *Events* pane.

Procedure

1. On the canvas, select a control.
2. In the *Events* pane, open the dropdown list under the event for whose controller you want to create a function.
3. In the dropdown list, select *New Function*.
4. In the *New Function* dialog box, enter a function name and click *OK*.

i Note

The function name you enter must be a valid JavaScript function name.

After entering the name for your new function, the code is created by the layout editor in the code editor.

To go directly to that code, click the *Go to code*  icon next to the new function name in the *Events* pane.

Task overview: [Working with the Layout Editor \[page 262\]](#)

Related Information

- [Add Controls from the Outline Tab \[page 263\]](#)
- [Delete Controls from the Outline Tab \[page 263\]](#)
- [Create a Navigation Between Views \[page 265\]](#)
- [Add Controls from the Controls Tab \[page 266\]](#)
- [Keyboard Support \[page 267\]](#)
- [Layout Editor Binding Capabilities \[page 268\]](#)
- [Layout Editor \[page 256\]](#)



7.1.8.1.4 Create a Navigation Between Views

You can create a navigation between views using the layout editor.

Prerequisites

- You have enabled the *Storyboard* perspective in the *Preferences* perspective, on the *Extensions* page.
- You have created a freestyle project.

Procedure

- In the *Development* perspective, in your Workspace, right-click the view to which you want to add a navigation, then in the context menu, choose **Open With** > *Layout Editor* .
- The view opens in the layout editor.
- From the *Controls* pane, drag a control and drop it onto the view canvas.
- Select the control and in the *Events* pane, in the *Press* field, click the Open Menu  button and select *Navigate To*.
- In the *Navigate To* dialog box, in the *View* dropdown list, select the view to which you want to navigate, choose *OK*, then choose *Save*.

In the *Storyboard* perspective, the navigation now appears as an arrow between the views you configured.

Task overview: [Working with the Layout Editor \[page 262\]](#)

Related Information

- [Add Controls from the Outline Tab \[page 263\]](#)

[Delete Controls from the Outline Tab \[page 263\]](#)
[Create a New Function \[page 264\]](#)
[Add Controls from the Controls Tab \[page 266\]](#)
[Keyboard Support \[page 267\]](#)
[Layout Editor Binding Capabilities \[page 268\]](#)
[Configure a Navigation Between Views \[page 332\]](#)

7.1.8.1.5 Add Controls from the Controls Tab

Add controls to the canvas by using drag and drop functionality.

Select the control that you want to add to the canvas from the *Controls* tab on the left side of the layout editor area and drag it to the drop target on the canvas. A tooltip displays the drop targets as you drag the control around the canvas.

Example

If you want to add an *HBox* container with a *Button* control to your view, do the following:

1. Choose the *Controls* tab.
2. Open the *Container* section by clicking the arrow to the right of the section title.
3. Select the *HBox* control and drag it to the canvas. Drop it at the position where you want it to appear.
4. Open the *Action* section by clicking the arrow to the right of the section title.
5. Select the *Button* control and drag it to the canvas.
6. Drop the *Button* control onto the *HBox* container.

Parent topic: [Working with the Layout Editor \[page 262\]](#)

Related Information

[Add Controls from the Outline Tab \[page 263\]](#)
[Delete Controls from the Outline Tab \[page 263\]](#)
[Create a New Function \[page 264\]](#)
[Create a Navigation Between Views \[page 265\]](#)
[Keyboard Support \[page 267\]](#)
[Layout Editor Binding Capabilities \[page 268\]](#)

7.1.8.1.6 Keyboard Support

Use the keyboard to move selected controls or navigate within the view that you opened with the layout editor.

Selecting Controls

If you selected a control in the canvas of the layout editor, you can move the selection using the arrow keys:

- `UP ARROW`: moves the selection to the parent of the selected control
- `DOWN ARROW`: moves the selection to the child of the selected control
- `LEFT ARROW`: moves the selection to the control that is up/to the right of the selected control (within the same aggregation)
- `RIGHT ARROW`: moves the selection to the control that is down/to the left of the selected control (within the same aggregation)
- `CTRL`+ click: moves the selection to the parent of the selected control

Moving Selected Controls

If you have selected a control in the canvas of the layout editor, you can change its position within the aggregation:

- `SHIFT`+`LEFT ARROW`: moves the control up/to the right
- `SHIFT`+`RIGHT ARROW`: moves the control down/to the left

Changing the Drop Target

When you drag and drop a control from the palette to the canvas or from one position within the canvas to another, you can use the keyboard to define the drop position of the dragged control:

- Use `SHIFT` or `ALT` to define the drop position of the dragged control within an aggregation:
 - `SHIFT`: moves the drop position up/to the right
 - `ALT`: moves the drop position down/to the left

Parent topic: [Working with the Layout Editor \[page 262\]](#)

Related Information

[Add Controls from the Outline Tab \[page 263\]](#)

[Delete Controls from the Outline Tab \[page 263\]](#)

[Create a New Function \[page 264\]](#)

[Create a Navigation Between Views \[page 265\]](#)

[Add Controls from the Controls Tab \[page 266\]](#)

[Layout Editor Binding Capabilities \[page 268\]](#)

7.1.8.1.7 Layout Editor Binding Capabilities

In the layout editor, you can bind properties of controls or control aggregations to data fields, `json` models, and label annotations.

Parent topic: [Working with the Layout Editor \[page 262\]](#)

Related Information

[Add Controls from the Outline Tab \[page 263\]](#)

[Delete Controls from the Outline Tab \[page 263\]](#)

[Create a New Function \[page 264\]](#)

[Create a Navigation Between Views \[page 265\]](#)

[Add Controls from the Controls Tab \[page 266\]](#)

[Keyboard Support \[page 267\]](#)

7.1.8.1.7.1 Binding Data

In the layout editor, you can bind properties of controls or control aggregations to an artifact in the OData service.

Prerequisites

You have defined a data set for the view that you are working on, by doing one of the following:

- If you are opening a view that has no data set defined for it, the *Data Binding* dialog box opens, where you can define a data set.
- Select the view, and define the data set from the dropdown list.

Caution

If you change the data set that is defined for the current view, the existing data bindings might become invalid.

Overview

The following types of bindings are possible:

- Properties of controls
- Aggregations of controls

i Note

To bind properties of models that are not OData models, you must work from the source code files and not from the layout editor. Alternatively, if you do not want to work with the source files in the XML editor, you can enter free text for properties in the *Properties* pane.

i Note

If your application does not consume an OData service, you can add the *OData Service* component to it.

[Define Entity Set \[page 269\]](#)

Before you can bind data to a control, you need to define an entity set.

[Bind Data to a Simple Control \[page 271\]](#)

You can bind data to a simple control.

[Bind Data to an Aggregate-Type Control \[page 271\]](#)

You can bind data to an aggregate-type control, which creates a template.

[Bind to the i18n Model \[page 272\]](#)

You can bind a control property to the i18n model or create a new i18n entry.

[Bind to a Label Annotation \[page 273\]](#)

You can bind a control property to a label annotation.

Related Information

[SAPUI5 API Reference](#)

7.1.8.1.7.1.1 Define Entity Set

Before you can bind data to a control, you need to define an entity set.

Prerequisites

- You have created a project in the layout editor. For more information, see [Working with the Layout Editor \[page 262\]](#).
- You have dragged a control from the *Controls* pane to the canvas. For more information, see [Add Controls from the Controls Tab \[page 266\]](#).

Context

Procedure

1. In the canvas, select the control for which you want to define an entity set.
2. In the *Properties* pane, click the icon.

The *Select Entity Set* dialog box opens.

3. Select and configure one of the entity sets for your control.

i Note

Most users should use either of the first two entity sets listed below.

- *Use ancestor's entity set.*
- *Define entity set and set the selected control as template.*
- *Define dummy entity set for the selected control.*

! Restriction

The "dummy" entity set is only for advanced users.

4. Choose OK.

Task overview: [Binding Data \[page 268\]](#)

Related Information

[Bind Data to a Simple Control \[page 271\]](#)

[Bind Data to an Aggregate-Type Control \[page 271\]](#)


[Bind to the i18n Model \[page 272\]](#)

[Bind to a Label Annotation \[page 273\]](#)

7.1.8.1.7.1.2 Bind Data to a Simple Control

You can bind data to a simple control.

Procedure

1. On the canvas, select the desired control for which you want to define data binding.
2. In the *Properties* pane to the right of the canvas, do one of the following:
 - To the right of the property to which you want to bind data, click the *Binding*  button, and then in the *Data Binding* dialog box:
 1. In the *Data Fields* list, double-click one or more data fields that you want to add to the expression. The data fields are automatically concatenated to the string in the *Expression* box.
 2. Click *OK* or first manually edit the expression string and then click *OK*.
 - In the *Properties* pane, for a field or dropdown list, manually enter the required expression within curly brackets {...} according to the data set that you selected.

Task overview: [Binding Data \[page 268\]](#)

Related Information

[Define Entity Set \[page 269\]](#)

[Bind Data to an Aggregate-Type Control \[page 271\]](#)

[Bind to the i18n Model \[page 272\]](#)

[Bind to a Label Annotation \[page 273\]](#)

7.1.8.1.7.1.3 Bind Data to an Aggregate-Type Control

You can bind data to an aggregate-type control, which creates a template.

Prerequisites

- You have defined an entity set. For more information, see [Define Entity Set](#).

Procedure

1. In the canvas or on the *Outline* tab to the left of the canvas, choose an aggregate-type control that you want to turn into a template, such as a *List Item* control.
2. In the properties pane to the right of the canvas, under the *Data Set* dropdown list, check the *Set as template* checkbox.

This control becomes a template.

3. In the *Confirmation Needed* dialog box, confirm the removal of any existing controls on the same level by clicking *OK*.

Results

The template item is now marked *Template* in the *Outline* tab.

Task overview: [Binding Data \[page 268\]](#)

Related Information

[Define Entity Set \[page 269\]](#)

[Bind Data to a Simple Control \[page 271\]](#)

[Bind to the i18n Model \[page 272\]](#)


[Bind to a Label Annotation \[page 273\]](#)

[Aggregation Binding](#)

7.1.8.1.7.1.4 Bind to the i18n Model

You can bind a control property to the i18n model or create a new i18n entry.

Procedure

1. On the canvas, select the desired control for which you want to define i18n model binding.
2. In the *Properties* pane to the right of the canvas, do one of the following:
 - To the right of the property to which you want to bind data, click the *Binding*  button, and then in the dialog box that appears:
 1. From the drop down list, select *i18n*, double-click one or more entries that you want to add to the expression.

- The entries are automatically concatenated to the string in the *Expression* box.
2. Click *OK* or first manually edit the expression string and then click *OK*.
- In the *Properties* pane, for a field or dropdown list, manually enter the required expression within curly brackets {...}.
 - Click + to add a new entry.

Task overview: [Binding Data \[page 268\]](#)

Related Information

[Define Entity Set \[page 269\]](#)

[Bind Data to a Simple Control \[page 271\]](#)

[Bind Data to an Aggregate-Type Control \[page 271\]](#)

[Bind to a Label Annotation \[page 273\]](#)


7.1.8.1.7.1.5 Bind to a Label Annotation

You can bind a control property to a label annotation.

Context

You can bind a control property to a label annotation that resides in the OData metadata file. Other annotation files are not supported.

Procedure

1. On the canvas, select the desired control for which you want to define a label annotation binding.
2. In the *Properties* pane to the right of the canvas, do one of the following:
 - To the right of the property to which you want to bind data, click the *Binding*  button, and then in the dialog box that appears:
 1. From the drop down list, select *Labels*, double-click one or more annotation that you want to add to the expression.

The annotations are automatically concatenated to the string in the *Expression* box.
 2. Click *OK* or first manually edit the expression string and then click *OK*.
 - In the *Properties* pane, for a field or dropdown list, manually enter the required expression within curly brackets {...}.

Task overview: [Binding Data \[page 268\]](#)

Related Information

[Define Entity Set \[page 269\]](#)

[Bind Data to a Simple Control \[page 271\]](#)

[Bind Data to an Aggregate-Type Control \[page 271\]](#)

[Bind to the i18n Model \[page 272\]](#)

7.1.8.2 Creating a Quick Start Application with the Layout Editor

Quickly create a new application using the layout editor.

Context

You can create a new application using the layout editor directly from the SAP Web IDE Welcome page. After adding controls and binding data to them, you can run the application using the provided mock data.

Procedure

1. On the SAP Web IDE Welcome page, click the [Quick Start with Layout Editor](#) tile.

Alternatively, choose **File > New > Quick Start with Layout Editor**.

2. Provide a name for your new application.

A new application opens in the layout editor.

i Note

This new application will run on mock data and is not for productive use.

3. Add any controls that you require by doing one of the following:
 - Drag and drop them onto the canvas from the [Controls](#) tab on the left side of the canvas.
 - Add them from the [Outline](#) tab on the left side of the canvas.
4. In the properties pane to the left of the canvas, bind data to each control as required.

Related Information

[Layout Editor \[page 256\]](#)

[Add Controls from the Outline Tab \[page 263\]](#)

[Binding Data \[page 268\]](#)

7.1.8.3 SAPUI5 Controls Supported in the Layout Editor

Provides a list of SAPUI5 controls that are supported in the layout editor.

Controls Tab

The SAPUI5 controls listed below can be dragged and dropped from the *Controls* tab onto the canvas.

i Note

The controls on the *Controls* tab are also available from the *Outline* tab. For more information, see [Add Controls from the Outline Tab \[page 263\]](#).

i Note

For more information about SAPUI5 controls, see [UI development toolkit for HTML5 - Demo Kit](#).

SAPUI5 Controls Available on the Controls Tab

SAPUI5 Control Name	Description
Action List Items sap.m.ActionListItem	Button that is used to fire actions when pressed.
Action Select sap.m.ActionSelect	Provides a list of predefined items that allows end users to choose options and additionally trigger some actions.
Add Bookmark Button sap.ushell.ui.footerbar.AddBookmarkButton	Button that is displayed in the application footer. Clicking the button opens a dialog box that allows the user to save the app state, so that the app can be launched in this state directly from the launchpad.
Analytic Map sap.ui.vbm.AnalyticMap	Renders a map based on a GeoJSON source.
App sap.m.App	The root element of an SAPUI5 mobile application. It inherits from the <code>NavContainer</code> control and thus provides its navigation capabilities. <code>App</code> provides certain header tags to the HTML page that are relevant for mobile apps.

SAPUI5 Control Name	Description
Bar sap.m.Bar	Centers a control like a title while having other controls on its left and right.
Breadcrumbs sap.m.Breadcrumbs	Enables users to navigate between items by providing a list of links to previous steps in the user's navigation path. The last three steps can be accessed as links directly, while the remaining links prior to them are available in a drop-down menu.
Busy Indicator sap.ui.core.BusyIndicator	Provides methods to show or hide a waiting animation that covers the whole page and blocks user interaction.
Button sap.m.Button	Allows users to trigger actions.
Calendar sap.ui.unified.Calendar	Basic calendar that is used for <code>DatePickers</code> .
Calendar Legend sap.ui.unified.CalendarLegend	Legend for the <code>Calendar</code> control. Displays special date colors with their corresponding description.
CalendarLegendItem sap.ui.unified.CalendarLegendItem	Item to be displayed in a <code>CalendarLegend</code> .
Carousel sap.m.Carousel	Navigates through a list of controls by swiping right or left.
Check Box sap.m.CheckBox	Allows the user to select one or multiple items from a list.
Component Container sap.ui.core.ComponentContainer	Container that embeds a <code>UIComponent</code> in a control tree.
Control Configuration sap.ui.comp.smartfilterbar.ControlConfiguration	Can be used to add additional configuration for filter fields in the <code>SmartFilterBar</code> control, in order to overwrite the default settings from the OData metadata. For instance, it is possible to change the label, index or control type of a filter field.
Column sap.m.Column	Allows definition of column-specific properties that are applied when rendering a <code>List</code> control.
Column List Item sap.m.ColumnListItem	Used with cell aggregation to create rows for the <code>sap.m.Table</code> control.
Combo Box sap.m.ComboBox	Combines a dropdown list with items and a text input field with a button allowing the user to either type a value directly or choose from a list of predefined items.

SAPUI5 Control Name	Description
Custom Data sap.ui.core.CustomData	Contains a single key/value pair of custom data attached to an Element.
Custom List Item sap.m.CustomListItem	<p>With content aggregation, can be used to customize standard list items that are not provided by SAPUI5. <code>ListItem</code> type is applied to <code>CustomListItem</code> as well.</p> <div style="background-color: #f0f0f0; padding: 10px; border-left: 2px solid #0070c0;"> <p>i Note Content aggregation allows any control. Complex responsive layout controls (such as <code>Table</code> and <code>Form</code>) should not be aggregated as content.</p> </div>
Custom Tile sap.m.CustomTile	Displays application-specific content in the <code>Tile</code> control.
Date Picker sap.m.DatePicker	Date input control with a calendar used as a date picker.
Detail Page sap.m.semantic.DetailPage	An <code>sap.m.semantic.ShareMenuPage</code> control that supports certain semantic buttons that have default semantic-specific properties and are eligible for content aggregation.
Display List Item sap.m.DisplayListItem	Used to represent a label and a value.
DraftIndicator sap.m.DraftIndicator	A draft indicator is <code>sap.m.Label</code> .
Feed Input sap.m.FeedInput	Allows the user to enter text for a new feed entry and then post it.
Feed List Item sap.m.FeedListItem	Provides a set of properties for text, sender information, and time stamp.
Filter Group Item sap.ui.comp.filterbar.FilterGroupItem	Represents a filter belonging to a group other than basic.
Filter Item sap.ui.comp.filterbar.FilterItem	Represents a filter belonging to the basic group.
Flex Item Data sap.m.FlexItemData	Holds layout data for a <code>FlexBox</code> / <code>HBox</code> / <code>VBox</code> .

SAPUI5 Control Name	Description
Flex Box sap.m.FlexBox	Builds the container for a flexible box layout.
Form sap.ui.layout.form.Form	Structured into <code>FormContainer</code> controls, each of which consists of <code>FormElement</code> controls.
Form Container sap.ui.layout.form.FormContainer	Group inside a <code>Form</code> .
Form Element sap.ui.layout.form.FormElement	Row in a <code>FormContainer</code> control.
Fullscreen Page sap.m.semantic.FullscreenPage	An <code>sap.m.semantic.ShareMenuPage</code> control that supports certain semantic buttons that have default semantic-specific properties and are eligible for content aggregation.
Generic Tile sap.m.GenericTile	Displays header, subheader, and a customizable main area in a tile format.
Geo Map sap.ui.vbm.GeoMap	A map control that allows the user to position multiple visual objects on top of a map.
Grid sap.ui.layout.Grid	Layout that positions its child controls in a 12-column flow layout.
Grid Container Data sap.ui.layout.form.GridContainer-Data	The <code>GridLayout</code> -specific layout data for <code>FormContainers</code> .
Grid Data sap.ui.layout.GridData	Grid layout data.
Grid Element Data sap.ui.layout.form.GridElement-Data	The <code>GridLayout</code> -specific layout data for <code>FormElement</code> fields.
Group sap.ui.comp.smartform.Group	Groups are used to group <code>group</code> elements.
Group Configuration sap.ui.comp.smartfilterbar.Group-Configuration	Can be used to add additional configurations for groups in the <code>SmartFilterBar</code> . A group in the <code>SmartFilterBar</code> is a group of filter fields in the advanced search.

SAPUI5 Control Name	Description
GroupElement sap.ui.comp.smartform.GroupElement	A combination of one label and different controls associated to this label.
Group Header List Item sap.m.GroupHeaderListItem	Used to display the title of a group and act as a separator between groups in <code>sap.m.List</code> and <code>sap.m.Table</code> .
HBox sap.m.HBox	Builds the container for a horizontal flexible box layout.
Horizontal Layout sap.ui.layout.HorizontalLayout	Provides support for horizontal alignment of controls.
Icon sap.ui.core.Icon	Uses an embedded font instead of a pixel image.
Icon Tab Bar sap.m.IconTabBar	Represents a collection of tabs with associated content.
Icon Tab Filter sap.m.IconTabFilter	Represents a selectable item inside an <code>Icon Tab Bar</code> control.
Icon Tab Header sap.m.IconTabHeader	Displays a number of <code>Icon Tab Filter</code> and <code>Icon Tab Separator</code> controls.
Icon Tab Separator sap.m.IconTabSeparator	Icon used to separate two <code>Icon Tab Filter</code> controls.
Image sap.m.Image	Wrapper around the <code>IMG</code> tag.
Input sap.m.Input	Allows users to input data.
Input List Item sap.m.InputListItem	List item used for a label and an input field.
Invisible Text sap.ui.core.InvisibleText	Used to bring hidden texts to the UI for screen reader support.
Item sap.ui.core.Item	Control base type.

SAPUI5 Control Name	Description
Label sap.m.Label	Used in SAPUI5 mobile applications to provide label text for other controls.
Layout Data sap.ui.core.LayoutData	A layout data base type.
Link sap.m.Link	Used to trigger actions or to navigate to other applications or web pages.
List sap.m.List	Provides a container for all types of list items.
List Item sap.ui.core.ListItem	Used in lists or list-like controls, such as <code>DropDownBox</code> .
Master Page sap.m.semantic.MasterPage	An <code>sap.m.semantic.SemanticPage</code> control that supports certain semantic buttons that have default semantic-specific properties and are eligible for content aggregation.
Message Strip sap.m.MessageStrip	Allows the embedding of application-related messages in the application.
Nav Container sap.m.NavContainer	Handles hierarchical navigation between <code>Page</code> controls or other fullscreen controls.
Object Attribute sap.m.ObjectAttribute	Displays a text field that can be normal or active.
Object Header sap.m.ObjectHeader	Allows the user to easily identify a special object.
Object Identifier sap.m.ObjectIdentifier	Display control that allows the user to easily identify a specific object.
Object List Item sap.m.ObjectListItem	Display control that provides summary information about an object as an item in a list.
Object Number sap.m.ObjectNumber	Displays number and number unit properties for an object.
Object Status sap.m.ObjectStatus	Status information that can be either text with a value state, or an icon.

SAPUI5 Control Name	Description
Overflow Toolbar sap.m.OverflowToolbar	Container based on <code>sap.m.Toolbar</code> that provides overflow when its content does not fit in the visible area.
Overflow Toolbar Button sap.m.OverflowToolbarButton	Represents an <code>sap.m.Button</code> that shows its text only when in the overflow area of an <code>sap.m.OverflowToolbar</code> .
Page sap.m.Page	Basic container for a mobile application screen.
Paging Button sap.m.PagingButton	Allows users to navigate between items and entities.
Pane Container sap.ui.layout.PaneContainer	An abstraction of <code>Splitter</code> . Could be used as an aggregation of <code>ResponsiveSplitter</code> or other <code>PaneContainers</code> .
Panel sap.m.Panel	Container for controls that has a header and content.
Progress Indicator sap.m.ProgressIndicator	Shows the progress of a process in a graphical way.
Pull To Refresh sap.m.PullToRefresh	Triggers the refresh event.
Radio Button sap.m.RadioButton	Control similar to <code>CheckBox</code> , but it allows the user to choose only one of a predefined set of options.
Radio Button Group sap.m.RadioButtonGroup	Used as a wrapper for a group of <code>sap.m.RadioButton</code> controls, which then can be used as a single UI element.
Rating Indicator sap.m.RatingIndicator	Used to rate content.
Responsive Flow Layout Data sap.ui.layout.ResponsiveFlowLayoutData	This is a <code>LayoutData</code> element that can be added to a control if this control is used within a <code>ResponsiveFlowLayout</code> .
Search Field sap.m.SearchField	Allows users to input a search string.

SAPUI5 Control Name	Description
Segmented Button sap.m.SegmentedButton	Horizontal control made of multiple buttons, which can display a title or an image.
Segmented Button Item sap.m.SegmentedButtonItem	Used for creating buttons for the <code>sap.m.SegmentedButton</code> . It is derived from the <code>sap.ui.core.Item</code> .
Select sap.m.Select	Provides a list of items that allows the user to select an item.
Select List sap.m.SelectList	Displays a list of items that allows the user to select an item.
Semantic Page sap.f.semantic.SemanticPage	An enhanced <code>sap.f.DynamicPage</code> , that contains controls with semantic-specific meaning.
Semantic Page sap.m.semantic.SemanticPage	An enhanced <code>sap.m.Page</code> control that can contain controls with semantic meaning. Content specified in <code>sap.m.semantic.SemanticPage</code> semantic control aggregations are automatically positioned in dedicated sections of the footer or the header of the page, depending on the control's semantics.
Share Menu Page sap.m.semantic.ShareMenuPage	An <code>sap.m.semantic.SemanticPage</code> control that supports a Share menu in the footer.
Shell sap.m.Shell	Can be used as root element of applications. It can contain an <code>App</code> or a <code>SplitApp</code> control. The Shell provides some overarching functionality for the overall application and takes care of visual adaptation, such as a frame around the App, on desktop browser platforms.
Simple Form sap.ui.layout.form.SimpleForm	Provides an API for creating simple forms. Inside a <code>SimpleForm</code> control, a <code>Form</code> control is created along with its <code>FormContainers</code> control and <code>FormElements</code> control, but the complexity in the API is removed.
Slider sap.m.Slider	User interface control that allows the user to adjust values within a specified numerical range.

SAPUI5 Control Name	Description
Smart Chart sap.ui.comp.smartchart.SmartChart	<p>Creates a chart based on OData metadata and the configuration specified. The <code>entitySet</code> property must be specified to use the control. This property is used to fetch fields from OData metadata, from which the chart UI will be generated. It can also be used to fetch the actual chart data.</p> <p>Based on the <code>chartType</code> property, this control will render the corresponding chart.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>i Note</p> <p>Most of the attributes are not dynamic and cannot be changed once the control has been initialized.</p> </div>
Smart Field sap.ui.comp.smartfield.SmartField	<p>A wrapper for other controls. It interprets OData metadata to determine the control that has to be instantiated. The OData entity is derived from the control's binding context. The OData entity's property that is changed or displayed with the control is derived from the control's value property.</p>
Smart Filter Bar sap.ui.comp.smartfilterbar.SmartFilterBar	<p>The <code>SmartFilterBar</code> control uses the OData metadata of an entity in order to create a <code>FilterBar</code>.</p> <p>Whether a field is visible on the <code>FilterBar</code>, supports type-ahead and value help, for example, is automatically determined. When you use control configurations and group configurations it is possible to configure the <code>FilterBar</code> and adapt it according to your needs.</p>
Smart Form sap.ui.comp.smartform.SmartForm	<p>Renders a form (<code>sap.ui.layout.form.Form</code>). When used with the <code>SmartField</code> control the label is taken from the metadata annotation <code>sap:label</code> if not specified directly.</p>
Smart Link sap.ui.comp.navpopover.SmartLink	<p>The <code>SmartLink</code> control uses a semantic object to display <code>NavigationPopover</code> for further navigation steps.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>i Note</p> <p>Navigation targets are determined using <code>CrossApplicationNavigation</code> of the unified shell service.</p> </div>
Smart Table sap.ui.comp.smarttable.SmartTable	<p>Creates a table based on OData metadata and the configuration specified. The <code>entitySet</code> attribute must be specified to use the control. This attribute is used to fetch fields from OData metadata, from which columns will be generated; it can also be used to fetch the actual table data.</p> <p>Based on the <code>tableType</code> property, this control will render a standard, analytical, tree, or responsive table.</p>
Split Pane sap.ui.layout.SplitPane	<p>A container of a single control in a responsive splitter. Could be used as an aggregation of a <code>PaneContainer</code>.</p>

SAPUI5 Control Name	Description
Splitter Layout Data sap.ui.layout.SplitterLayoutData	Holds layout data for the splitter contents. Allowed size values are numeric values ending in "px" and "%" and the special case "auto". (The CSS value "auto" is used internally to recalculate the size of the content dynamically and is not directly set as style property.)
Standard List Item sap.m.StandardListItem	List item that provides the most common use cases, such as image, title, and description.
Standard Tile sap.m.StandardTile	Displayed in the <code>Tile</code> container.
Switch sap.m.Switch	User interface control on mobile devices that is used for switching between binary states.
Table sap.m.Table	Provides a set of sophisticated and convenient functions for responsive table design.
Text sap.m.Text	Used for embedding longer text paragraphs that need text wrapping into your application.
Text Area sap.m.TextArea	Allows multiline text input.
Tile Container sap.m.TileContainer	Container that arranges same-size tiles on carousel pages.
Tile Content sap.m.TileContent	This control is used within the <code>GenericTile</code> control.
Title sap.ui.core.Title	Used to aggregate other controls.
Time Picker sap.m.TimePicker	A single-field input control that enables the users to fill time related input fields.
Toggle Button sap.m.ToggleButton	Control that toggles between pressed and normal state.
Toolbar sap.m.Toolbar	Horizontal container that is usually used to display buttons, labels, selects, and other input controls.

SAPUI5 Control Name	Description
Toolbar Layout Data sap.m.ToolbarLayoutData	Defines layout data for the <code>sap.m.Toolbar</code> items. i Note ToolbarLayoutData should not be used together with <code>sap.m.ToolbarSpacer</code> .
Toolbar Separator sap.m.ToolbarSeparator	Creates a visual separator between toolbar items.
Toolbar Spacer sap.m.ToolbarSpacer	Adds horizontal space between toolbar items.
Upload Collection sap.m.UploadCollection	Allows users to upload single or multiple files.
Upload Collection Item sap.m.UploadCollectionItem	Provides information about uploaded files.
Variant Layout Data sap.ui.core.VariantLayoutData	Allows to add multiple LayoutData to one control in case that an easy switch of layouts (e.g. in a Form) is needed.
VBox sap.m.VBox	Builds the container for a vertical flexible box layout.
Vertical Layout sap.ui.layout.VerticalLayout	Layout in which the content controls are rendered one below the other.
XML View sap.ui.core.mvc.XMLView	A View defined using (P)XML and HTML markup.

Outline Tab

The SAPUI5 controls listed below are available only from the *Outline* tab in the layout editor.

i Note

For more information about SAPUI5 controls, see [UI development toolkit for HTML5 - Demo Kit](#).

SAPUI5 Controls Available on the Outline Tab

SAPUI5 Control Name	Description
<code>sap.m.semantic.AddAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.m.semantic.CancelAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.f.semantic.CloseAction</code>	A semantic-specific button, eligible for the <code>closeAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.
<code>sap.f.semantic.CopyAction</code>	A semantic-specific button, eligible for the <code>copyAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.
<code>sap.f.semantic.DeleteAction</code>	A semantic-specific button, eligible for the <code>deleteAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.
<code>sap.f.semantic.DiscussInJamAction</code>	A semantic-specific button, eligible for the <code>discussInJamAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in the share menu within its title.
<code>sap.m.semantic.DiscussInJamAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> .
<code>sap.f.semantic.EditAction</code>	A semantic-specific button, eligible for the <code>editAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.
<code>sap.m.semantic.EditAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.f.semantic.ExitFullScreenAction</code>	A semantic-specific button, eligible for the <code>exitFullScreenAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.
<code>sap.f.semantic.FavoriteAction</code>	A semantic-specific button, eligible for the <code>favoriteAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.

SAPUI5 Control Name	Description
<code>sap.m.semantic.FavoriteAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.m.semantic.FilterAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.m.semantic.FilterSelect</code>	A <code>sap.m.Select</code> control enhanced with styling according to the semantics of a common "Filter" action. A <code>FilterSelect</code> cannot be used independently but only as aggregation content of a <code>sap.m.semantic.SemanticPage</code> .
<code>sap.f.semantic.FlagAction</code>	A semantic-specific button, eligible for the <code>flagAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.
<code>sap.m.semantic.FlagAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.f.semantic.FooterMainAction</code>	A semantic-specific button, eligible for the <code>footerMainAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its footer.
<code>sap.m.semantic.ForwardAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.f.semantic.FullScreenAction</code>	A semantic-specific button, eligible for the <code>fullScreenAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.
<code>sap.m.semantic.GroupAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.m.semantic.GroupSelect</code>	A <code>sap.m.Select</code> control enhanced with styling according to the semantics of a common "Group" action. A <code>GroupSelect</code> cannot be used independently but only as aggregation content of a <code>sap.m.semantic.SemanticPage</code> .

SAPUI5 Control Name	Description
<code>sap.m.semantic.MainAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.f.semantic.MessagesIndicator</code>	A semantic-specific button, eligible for the <code>messagesIndicator</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its footer.
<code>sap.m.semantic.MessagesIndicator</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.m.semantic.MultiSelectAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.f.semantic.NegativeAction</code>	A semantic-specific button, eligible for the <code>negativeAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its footer.
<code>sap.m.semantic.NegativeAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.m.semantic.OpenInAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.f.semantic.PositiveAction</code>	A semantic-specific button, eligible for the <code>positiveAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its footer.
<code>sap.m.semantic.PositiveAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
<code>sap.f.semantic.PrintAction</code>	A semantic-specific button, eligible for the <code>printAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in the share menu within its title.
<code>sap.m.semantic.PrintAction</code>	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.

SAPUI5 Control Name	Description
sap.m.semantic.SaveAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
sap.f.semantic.SendEmailAction	A semantic-specific button, eligible for the <code>sendEmailAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in the share menu within its title.
sap.m.semantic.SendEmailAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
sap.f.semantic.SendMessageAction	A semantic-specific button, eligible for the <code>sendMessageAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in the share menu within its title.
sap.m.semantic.SendMessageAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> .
sap.f.semantic.ShareInJamAction	A semantic-specific button, eligible for the <code>shareInJamAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in the share menu within its title.
sap.m.semantic.ShareInJamAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
sap.m.semantic.SortAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an <code>sap.m.semantic.SemanticPage</code> control.
sap.f.semantic.TitleMainAction	A semantic-specific button, eligible for the <code>titleMainAction</code> aggregation of the <code>sap.f.semantic.SemanticPage</code> to be placed in its title.
sap.m.Title	Used for header texts and title.
sap.m.Token	Renders a token containing text and an optional <i>Delete</i> icon.
sap.m.Tokenizer	Displays multiple tokens.
sap.suite.ui.commons.BusinessCard	Allows displaying of business card information, including an image, first title (either URL link or text), second title, and multiple text lines.

SAPUI5 Control Name	Description
<code>sap.suite.ui.commons.ComparisonChart</code>	Displays a comparison chart.
<code>sap.suite.ui.commons.ComparisonData</code>	Comparison tile value holder.
<code>sap.suite.ui.commons.CountingNavigationItem</code>	Extends the <code>sap.ui.ux3.NavigationItem</code> control. This control displays the number of items in a corresponding content area. It also provides a rich tooltip that appears and disappears after a certain delay.
<code>sap.suite.ui.commons.DateRangeScroller</code>	Provides a method to scroll through a series of time periods, each of which is represented by a start date and an end date, known as the date range.
<code>sap.suite.ui.commons.DateRangeSliderInternal</code>	Provides the user with a <code>RangeSlider</code> control that is optimized for use with dates.
<code>sap.suite.ui.commons.DeltaMicroChart</code>	Displays a delta of two values as a chart.
<code>sap.suite.ui.commons.DynamicContainer</code>	Displays multiple <code>GenericTile</code> controls as changing slides.
<code>sap.suite.ui.commons.FacetOverview</code>	Used in <code>UnifiedThingInspector</code> to display a preview of facet content.
<code>sap.suite.ui.commons.GenericTile</code>	Tile control that displays a title, description, and customizable main area.
<code>sap.suite.ui.commons.HarveyBallMicroChart</code>	Chart that shows a comparative part to a total.
<code>sap.suite.ui.commons.HarveyBallMicroChartItem</code>	Configuration of a slice on a pie chart.
<code>sap.suite.ui.commons.HeaderCell</code>	Contains four cells (West, North, East, South). It can display one or more controls in different layouts. Each aggregation must contain only one instance of <code>HeaderCellItem</code> .
<code>sap.suite.ui.commons.HeaderCellItem</code>	Object that contains an instance of a control and information about its height. It should be used inside <code>sap.suite.ui.commons.HeaderCell</code> .
<code>sap.suite.ui.commons.HeaderContainer</code>	Container that provides a horizontal layout. On mobile devices, it provides a horizontal scroll. On desktops, it provides scroll left and scroll right buttons. This control supports keyboard navigation. You can use left and right arrow keys to navigate through the inner content. The <code>Home</code> key puts focus on the first control and the <code>End</code> key puts focus on the last control. Use the <code>Enter</code> key or <code>Spacebar</code> key to choose the control.

SAPUI5 Control Name	Description
sap.suite.ui.commons.JamContent	Displays SAP Jam content text, subheader, and numeric value in a tile.
sap.suite.ui.commons.KpiTile	Used in <code>UnifiedThingInspector</code> to display object-related KPIs in a factsheet.
sap.suite.ui.commons.NewsContent	Displays news content text and subheader in a tile.
sap.suite.ui.commons.NoteTaker	Allows creation and storage of notes for further reference.
sap.suite.ui.commons.NoteTakerCard	Allows storage of <code>NoteTaker</code> card header and body text.
sap.suite.ui.commons.NoteTakerFeeder	Allows entering quick notes and note cards.
sap.suite.ui.commons.NumericContent	Numeric content to be used in a tile or other place where it is needed to show numeric values with semantic colors and deviations.
sap.suite.ui.commons.TileContent	Serves as a universal container for different types of content and footer.
sap.ui.commons.ApplicationHeader	Located at the top of an application page and consists of four areas.
sap.ui.commons.Button	Allows users to trigger actions such as save or print. For the button UI, you can define text or an icon, or both.
sap.ui.commons.CheckBox	Provides a box that can be flagged and has a label. A checkbox can either stand alone, or be in a group with other checkboxes.
sap.ui.commons.ColorPicker	Allows the user to choose a color. The color can be defined using HEX, RGB, or HSV values, or a CSS colorname.
sap.ui.commons.FileUploader	Framework that generates an input field and a button with the text <i>Browse...</i>
sap.ui.commons.MenuBar	Represents a user interface area that is the entry point for menus with their menu items.
sap.ui.commons.MenuButton	Common button control that opens a menu when chosen by the user. The control provides an API for configuring the docking position of the menu.
sap.ui.commons.Paginator	Provides navigation between pages within a list of numbered pages.
sap.ui.commons.Panel	Represents a container with scroll functionality that can be used for text and controls.

SAPUI5 Control Name	Description
sap.ui.commons.PasswordField	Text field with masked characters that borrows its properties and methods from the <code>TextField</code> control.
sap.ui.commons.ProgressIndicator	Shows the progress of a process in a graphical way.
sap.ui.commons.RadioButton	Consists of a round element and descriptive text.
sap.ui.commons.RangeSlider	Interactive control that is displayed either as a horizontal or vertical line with two pointers and units of measurement.
sap.ui.commons.RatingIndicator	Allows the user to rate a certain topic.
sap.ui.commons.SegmentedButton	Provides a group of buttons.
sap.ui.commons.Slider	Interactive control that is displayed either as a horizontal or vertical line with a pointer and units of measurement.
sap.ui.commons.Splitter	Allows splitting the screen into two areas.
sap.ui.commons.TextArea	Control for entering or displaying multiple rows of text.
sap.ui.commons.TextField	Renders an input field for text input.
sap.ui.commons.Toolbar	Horizontal row of items where in many cases the single toolbar items are buttons that contain icons.
sap.ui.commons.Tree	Simple tree for displaying an item in a hierarchical way.
sap.ui.commons.TreeNode	Tree node element.
sap.ui.layout.FixFlex	Builds the container for a layout with a fixed and a flexible part.
sap.ui.layout.form.FormLayout	Base layout used to render a <code>Form</code> control.
sap.ui.layout.form.GridLayout	Renders a <code>Form</code> control using an HTML table-based grid.
sap.ui.layout.form.ResponsiveGridLayout	Renders a <code>Form</code> control using a responsive grid.
sap.ui.layout.form.ResponsiveLayout	Renders a <code>Form</code> control with a responsive layout.
sap.ui.layout.ResponsiveSplitter	A responsive splitter which divides the application into several areas.
sap.ui.unified.Currency	Text view that displays currency values and aligns them at the separator.
sap.ui.unified.FileUploader	Framework that generates an input field and a button with the text <i>Browse</i>

SAPUI5 Control Name	Description
sap.ui.unified.FileUploaderParameter	Represents a parameter for the <code>FileUploader</code> , which is rendered as a hidden input field.
sap.ui.unified.Menu	Interactive element that provides a choice of different actions to the user.
sap.ui.unified.MenuItem	Standard item used inside a menu. Represents an action that can be selected by a user in the menu or that can be used as a submenu that organizes the actions hierarchically.
sap.ui.unified.ShellOverlay	Opened in front of an <code>sap.ui.unified.Shell</code> control.
sap.ui.unified.SplitContainer	Provides a main content and a secondary content area.
sap.ui.ux3.ExactArea	Consists of two sections: a toolbar and a content area where arbitrary controls can be added.
sap.ui.ux3.FeedChunk	Unit that is embedded, standalone or multiple, into a <code>Feed</code> control.
sap.ui.ux3.Feeder	Lean common feed, or a comment feed, with a text commit function.
sap.uxap.ObjectPageHeader	Static part of an <code>Object</code> page header.
sap.uxap.ObjectPageHeaderContent	Dynamic part of an <code>Object</code> page header.

7.1.8.4 Try It: Build an Application with the Layout Editor

Get an overview of the features that are available with the layout editor by following this tutorial for building an application.

Related Information

[Prerequisites \[page 294\]](#)

[Create an OData Model File \[page 294\]](#)

[Create a Project for Your New Application \[page 295\]](#)

[Add Controls to Your New Application \[page 297\]](#)

7.1.8.4.1 Prerequisites

Prerequisite steps that you must complete before following the steps in the tutorial topic "Add Controls to Your New Application."

1. You have created an OData model file according to the instructions in the topic [Create an OData Model File \[page 294\]](#).
2. You have created a new project from a template according to the instructions in the topic [Create a Project for Your New Application \[page 295\]](#).

Related Information

[Create an OData Model File \[page 294\]](#)

[Create a Project for Your New Application \[page 295\]](#)

[Add Controls to Your New Application \[page 297\]](#)

7.1.8.4.1.1 Create an OData Model File

This task is a prerequisite for the tutorial on building an application with the layout editor.

Procedure

1. Open a new text file.
2. Copy and paste the XML code provided below into the text file.

```
<edmx:Edmx Version="1.0" xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" xmlns:sap="http://www.sap.com/Protocols/SAPData">
<edmx:DataServices xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" m:DataServiceVersion="2.0">
<Schema xmlns="http://schemas.microsoft.com/ado/2008/09/edm"
Namespace="MySalesOrders">
<EntityType Name="SalesOrder">
<Key>
<PropertyRef Name="SalesOrderNumber" />
</Key>
<Property Name="SalesOrderNumber" Type="Edm.String" sap:label="Sales Order
Number" Nullable="false" MaxLength="10" />
<Property Name="TotalAmount" Type="Edm.Decimal" sap:label="Total Amount"
Precision="16" Scale="3" sap:unit="Currency" MaxLength="10" />
<Property Name="Currency" Type="Edm.String" sap:label="Currency"
MaxLength="5" sap:semantics="currency-code" />
<Property Name="CustomerID" Type="Edm.String" sap:label="Customer ID"
MaxLength="10" />
<Property Name="CustomerName" Type="Edm.String" sap:label="Customer Name"
MaxLength="35" />
<Property Name="NetPriceAmount" Type="Edm.Decimal" sap:label="Net Price
Amount" Precision="16" Scale="3" sap:unit="Currency" />
<Property Name="TaxAmount" Type="Edm.Decimal" sap:label="Tax Amount"
Precision="16" Scale="3" sap:unit="Currency" />

```

```

    <Property Name="OrderDate" Type="Edm.String" sap:label="Order Date" />
    <Property Name="RequestedDate" Type="Edm.String" sap:label="Requested
Date" />
    <Property Name="Status" Type="Edm.String" MaxLength="1"
sap:label="Status" />
    <Property Name="SalesOrganization" Type="Edm.String" MaxLength="4"
sap:label="Sales Organization" />
    <Property Name="SalesOrganizationName" Type="Edm.String" MaxLength="20"
sap:label="SalesOrganizationName" sap:creatable="false" sap:updatable="false"
sap:sortable="false" sap:filterable="false" />
    <Property Name="DistributionChannel" Type="Edm.String" MaxLength="4"
sap:label="Distribution Channel" />
    <Property Name="Division" Type="Edm.String" MaxLength="2"
sap:label="Division" />
    <Property Name="DistributionChannelName" Type="Edm.String" MaxLength="20"
sap:label="DistributionChannelName" sap:creatable="false"
sap:updatable="false" sap:sortable="false" sap:filterable="false" />
    <Property Name="DivisionName" Type="Edm.String" MaxLength="20"
sap:label="DviisionName" sap:creatable="false" sap:updatable="false"
sap:sortable="false" sap:filterable="false" />
  </EntityType>
<EntityContainer Name="MySalesOrders_Entities"
m:IsDefaultEntityContainer="false">
  <EntitySet Name="SalesOrders" EntityType="MySalesOrders.SalesOrder"
sap:searchable="true" sap:requires-filter="true" />
</EntityContainer>
</Schema>
</edmx:DataServices>
</edmx:Edmx>

```

3. Save the file to your computer with the file name `SalesOrderService_metadata.xml`.

7.1.8.4.1.2 Create a Project for Your New Application

This task is a prerequisite for the tutorial on building an application with the layout editor.

Context

Create a new project for a sales order tracking application using a template.

Procedure

1. Open SAP Web IDE in the Google Chrome browser.
2. Open a new project from a template by using one of the following options:
 - On the SAP Web IDE Welcome page, choose [New Project from Template](#).
 - In the *File* menu, choose **New** **>** [Project from Template](#).

The [New Project](#) wizard opens.

3. In the [Template Selection](#) wizard step, select the [SAP Fiori Master Detail Application](#) tile and then choose the [Next](#) button.

- In the *Basic Information* wizard step, enter a project name. Choose the *Next* button.

i Note

The project name must start with a letter or an underscore and may contain alphanumeric characters, periods, and underscores. It may not end with a period.

- In the *Data Connection* wizard step, select the *File System* source. Then choose the *Browse* button and navigate to the file `SalesOrderService_metadata.xml` that you created in the topic [Create an OData Model File \[page 294\]](#). Choose the *Next* button.
- In the *Template Customization* wizard step, enter or select the mapping data in the fields and dropdown lists according to the tables below.

Application Settings Section

Field or Dropdown List	Value
<i>Title</i>	Sales Orders
<i>Project Namespace</i>	<project_namespace>
<i>OData Collection</i>	<i>SalesOrders</i>
<i>Item Title</i>	<i>SalesOrderNumber</i>
<i>Numeric Attribute</i>	<i>TotalAmount</i>
<i>Unit of Measure</i>	<i>Currency</i>

i Note

The namespace must start with a letter or an underscore and may contain alphanumeric characters, periods, and underscores. It may not end with a period.

Detail Section

Field or Dropdown List	Value
<i>Title</i>	Sales Order
<i>Detail Text</i>	Sales Order Details
<i>Status Attribute</i>	Leave it blank
<i>Attribute 1</i>	<i>NetPriceAmount</i>
<i>Attribute 2</i>	<i>TaxAmount</i>
<i>Attribute 3</i>	<i>OrderDate</i>

- Choose the *Next* button.
- In the *Confirmation* wizard step, choose the *Finish* button.

Your new project is now created in your workspace.

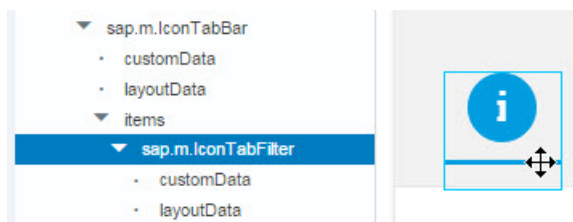
7.1.8.4.2 Add Controls to Your New Application



Steps for adding controls to your new sales order track *Open Withing* application.

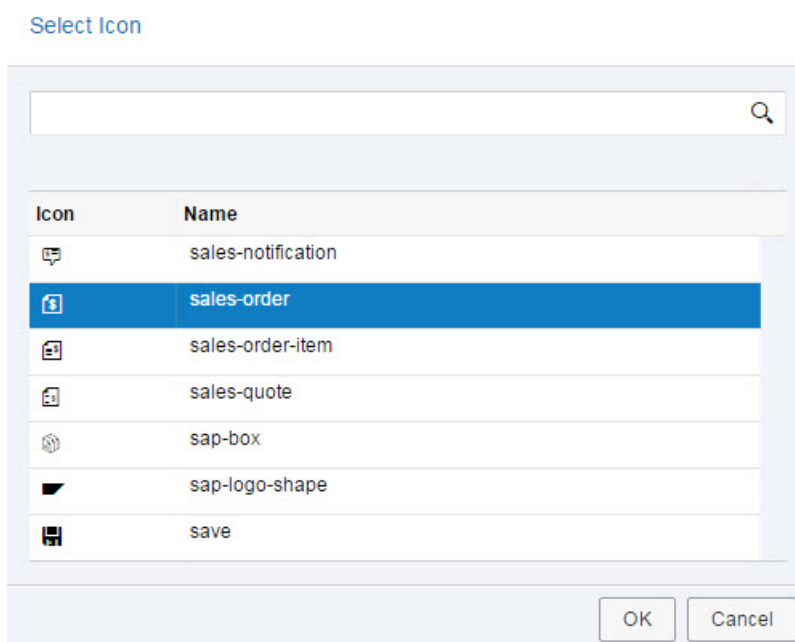
Procedure






1. In the workspace, expand the folder with name of the project that you created in the topic [Create a Project for Your New Application \[page 295\]](#), then expand the `view` folder and right-click the `Detail.view.xml` view.
2. From the context menu, choose **Layout Editor**.
The content of the XML view is displayed on the canvas in a way that corresponds to how it will appear in your finished application.
3. Change the icons of the *Icon Tab Filter* controls:

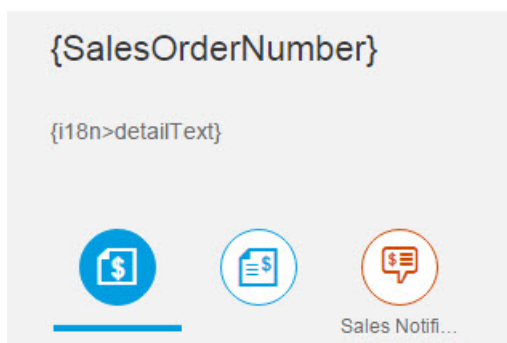
- a. On the canvas, select the first *Icon Tab Filter*  control on the left side of the *Icon Tab Bar* control. In the *Outline* tab on the left side of the canvas, `sap.m.IconTabFilter` is selected.



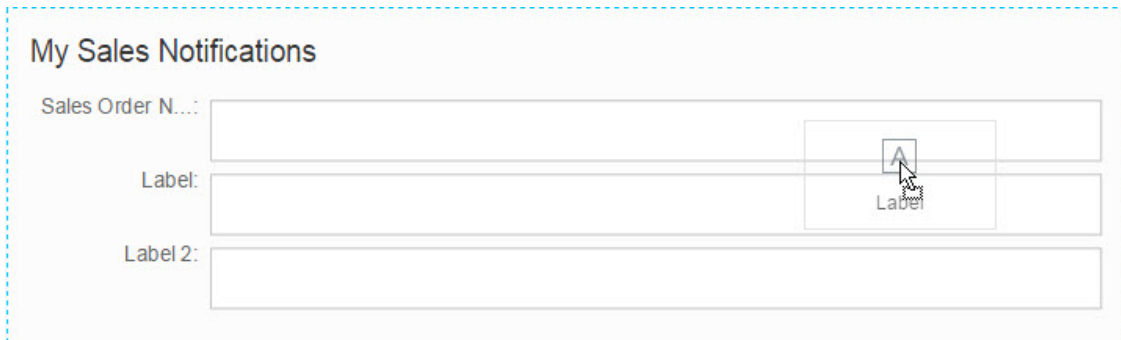
- b. In the *Properties* pane to the right of the canvas, in the *Icon* field, choose the *Select Icon*  button to open the *Select Icon* dialog box. Scroll down and select the *sales-order*  icon and then choose *OK*.



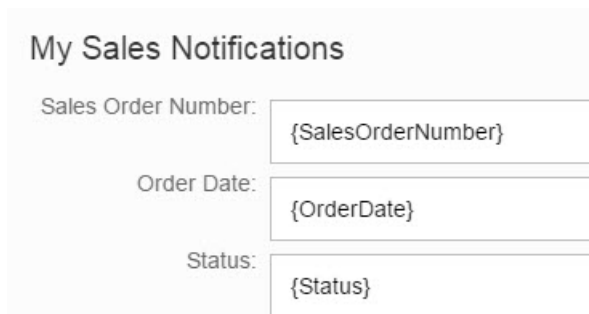
- c. On the canvas, select the second *Icon Tab Filter*  control.
 - d. In the *Properties* pane, in the *Icon* field, choose the  button to open the *Select Icon* dialog box. Scroll down and select the *sales-order-item*  icon and then choose *OK*.
4. Add a new *Icon Tab Filter* control to the view:
- a. On the *Controls* tab to the left of the canvas, expand the *Container* section or use the search field to search for the *Icon Tab Filter* control.
 - b. From the *Controls* tab, drag the *Icon Tab Filter* control to the canvas and drop it on the *Icon Tab Bar* control.
 - c. Change its  icon to the *sales-notification*  icon in the same way as you changed the icons in step 3.
 - d. In the *Properties* pane, in the *Count* field, clear the value by deleting it.
 - e. In the *Properties* pane, change the value in the *Text* field to *Sales Notifications*.




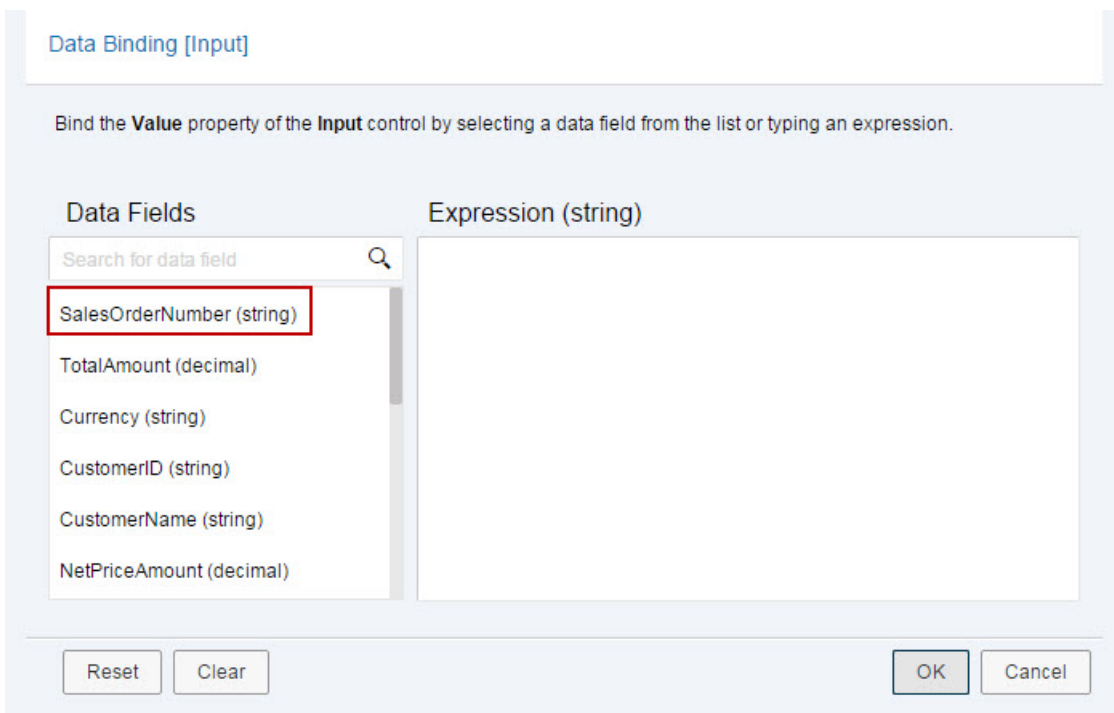
5. Add a *Simple Form* control to the new sales notification *Icon Tab Filter* control:
 - a. On the *Controls* tab to the left of the canvas, expand the *Layout* section.
 - b. In the canvas, select the *Sales Notification Icon Tab Filter*.
 - c. From the *Controls* tab on the left, drag the *Simple Form* control to the canvas and drop it on the space below the *Icon Tab Filter* control.
6. Change the properties of the new *Simple Form* control:
 - a. In the canvas, select the title in the *Simple Form* control.
 - b. In the *Properties* pane, change the value of the *Text* property to **My Sales Notifications**.
 - c. In the canvas, in the *Simple Form* control, select *Label 1*.
 - d. In the *Properties* pane, change the value of the *Text* property to **Sales Order Number**.
 - e. On the *Controls* tab to the left of the canvas, search for the *Label* control.
 - f. Drag a new label and drop it above the second input field.





- g. In the canvas, in the *Simple Form* control, select the label that you just created and then in the properties pane to the right of the canvas, change its text to **Order Date**.
- h. Change the text of the *Label 2* control of the third input field in the same way to **Status**.

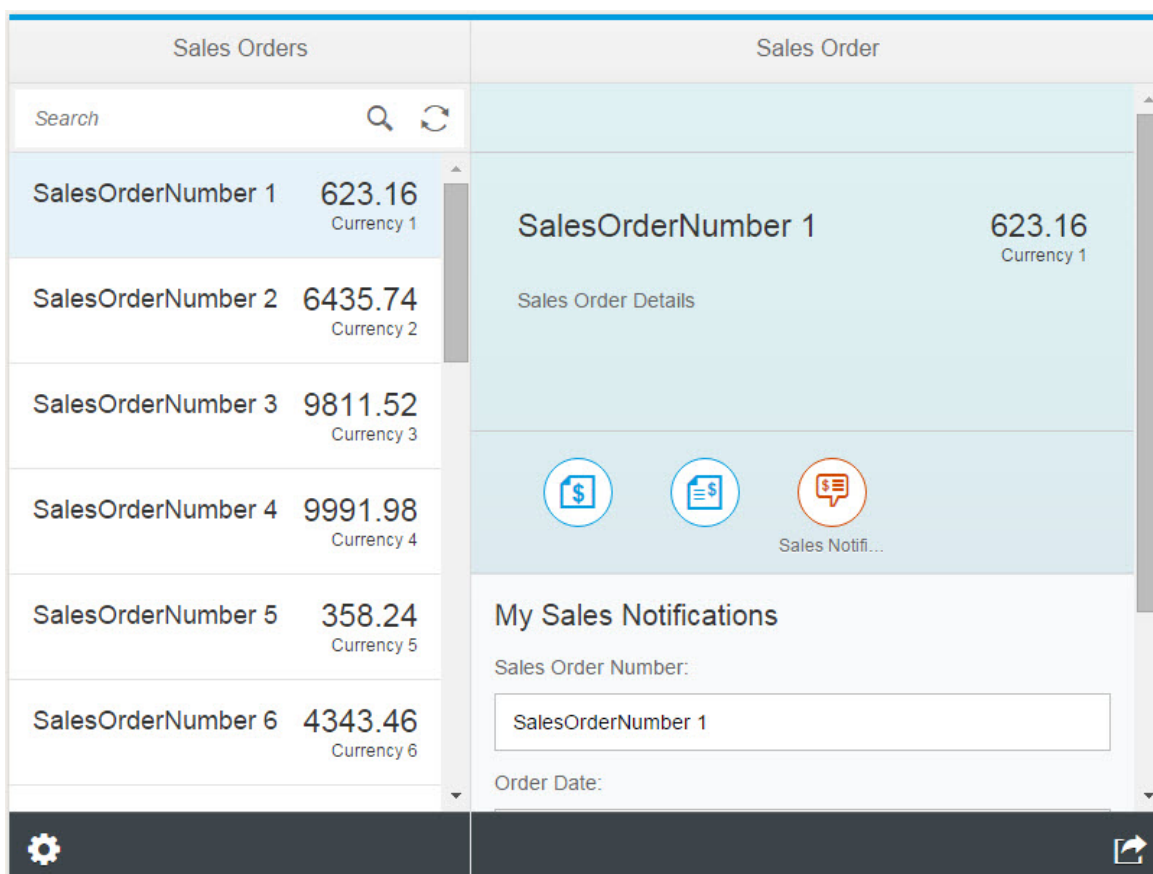


7. Bind the *Value* property of the input fields to elements from the OData service that you created:
 - a. In the canvas, select the *Sales Order Number* field in the *SimpleForm* control.
 - b. In the *Properties* pane, click the *Data Binding*  button next to the *Value* property.
 - c. In the *Data Binding [Input]* dialog box, select *SalesOrderNumber* and choose *OK*.



- d. In the canvas, select the *Order Date* field in the *Simple Form* control.
 - e. In the *Properties* pane, click the *Data Binding*  button next to the *Value* property.
 - f. In the *Data Binding* dialog box, select *OrderDate* and choose *OK*.
 - g. Select the *Status* field and bind it to *Status* in the same way as the previous two input fields.
8. Save your changes.
Your changes are saved for both the layout editor and the code editor.
 9. Run your application to test the result:
 - a. In your project folder, right-click the `index.html` file.
 - b. From the context menu, choose **Run** > *Run with Mock Data* .

The application preview starts in a new browser tab and the application loads with the mock data, based on the service structure that you created.



7.1.9 Annotation Modeler

The annotation modeler provides an intuitive user interface for annotating OData services. It enables you to build and enhance the user interface of model-based applications in the cloud by adding and editing UI annotations of the local annotation file.

You can also override annotations from other sources, such as metadata, by editing them locally in your SAP Web IDE project.

To run and use annotation modeler with your annotation file, some prerequisites need to be fulfilled. See [Prerequisites for Working with Annotation Modeler \[page 305\]](#).

i Note

Annotation modeler can be used with different SAP Web IDE projects:

- You can use annotation modeler with a [List Report Page](#) project to build applications based on OData services and annotations that require no JavaScript UI coding. See [Developing Apps with SAP Fiori Elements](#).
- You can use annotation modeler with an [Overview Page Application](#) project to define the annotation terms that can be referenced in cards as part of their properties. See [Overview Pages: Create Interactive Overviews of a Subject Area](#).

→ Tip

If you cannot define annotations in the backend, use annotation modeler to develop and enhance applications based on Fiori Elements whenever possible.

Use annotation modeler with mock data if you develop your service locally in a SAP Web IDE project or if you want to try out how annotations affect the UI. See [Configure Annotation Modeler to Use Mock Data \[page 109\]](#).

Use SAPUI5 Visual Editor to do flex changes for freestyle apps, app variants and personalization as well as for any adjustments that are not possible by using annotations. See [SAPUI5 Visual Editor Usage](#).

For more information about annotation modeler, watch the following video:

[Getting started with the SAP Web IDE annotation modeler](#) 🎥

Related Information

[Architecture \[page 301\]](#)

[User Interface \[page 303\]](#)

[Prerequisites for Working with Annotation Modeler \[page 305\]](#)

[Adding Local Annotation Files to the Project \[page 307\]](#)

[Working with Annotation Modeler \[page 309\]](#)

7.1.9.1 Architecture

Infrastructure

The annotation modeler is embedded in the SAP Web IDE infrastructure and can be used directly in SAP Web IDE. Simply enable the annotation modeler feature (extension), see [Enable SAP Web IDE Extensions \[page 510\]](#).

Supported OData Vocabularies

The annotation modeler is based on the [official OASIS vocabularies](#) (OData version 4.0) and specific SAP vocabularies that are stored as local copies in SAP Web IDE annotation modeler.

The annotation modeler recognizes annotation files containing the terms from these vocabularies and displays them in a tabular view. You can also add, edit, and delete annotations that are defined in these vocabularies and that apply to the levels listed below.

The annotation modeler supports only standard vocabularies. The following vocabularies are supported:

- **OData.org:**
 - [Aggregation](#)
 - [Authorization](#)
 - [Capabilities](#)
 - [Core](#)
 - [Measures](#)
 - [Validation](#)
- **SAP:**
 - [Analytics](#)
 - [CodeList](#)
 - [Common](#)
 - [Communication](#)
 - [PersonalData](#)
 - [UI](#)

i Note

The annotation modeler might not support all vocabulary terms listed in the mentioned OData vocabularies.

→ Tip

To find out more about how to prepare UI annotations for SAP Fiori elements, see [SAPUI5 Developer Guide: Preparing UI Annotations](#).

Scope

The annotation modeler supports applying annotations to the following targets:

- Actions
- Action Imports
- Entity Container
- Entity Set
- Entity Type
- Entity Type Property

Note

Exception: Entity type properties of complex type are not supported.

- Function
- Function Import
- Function Import Parameters

The annotation modeler also allows you to annotate the collection records and property values of the annotations.

Limitations

- The annotation modeler only supports the services of OData version 2.0.

7.1.9.2 User Interface

The screenshot displays the SAP Web IDE Annotation Modeler interface. At the top, there is a tab labeled 'annotationNavigation.xml'. Below the tab, the 'Annotation Structure' section shows the OData Data Source as 'mainService'. A search bar is present on the right. The main area is a table with columns: Node, Edit Qualifier, Key Information, Expression Type, Value, and Actions. The table is organized into a tree structure under 'Entity Types'. The 'Local Annotations' section is expanded, showing 'UI.DataFieldForAnnotation' with a target of '@UI.Chart#SpecificationWeightBulletChart'. Below this, a 'Target' row shows the annotation path 'Navigation' and the target '@UI.Chart#SpecificationWeightBulletChart'. Another 'Target' row shows the annotation path 'to_Supplier' and the target '@Communication.Contact'. The 'External Annotations' section is also expanded, showing 'UI.Chart#SpecificationWeightBulletC' with a chart type of 'Donut' and measures of 'Weight'. The 'MeasureAttributes' section is expanded, showing the 'Item' property path 'Weight'.

Node	Edit Qualifier	Key Information	Expression Type	Value	Actions
Entity Types					
STTA_C_MP_ProductType					
Local Annotations		Source: annotationNavigation			+
UI.LineItem#Products					+
UI.DataFieldForAnnotation		Target: (@UI.Chart#SpecificationWeightBulletChart)			+
Target			AnnotationPath	Navigation @UI.Chart#SpecificationWeightBulletChart	+
UI.DataFieldForAnnotation		Target: (to_Supplier#@Communication.Contact)			+
Target			AnnotationPath	to_Supplier @Communication.Contact	+
Local Annotations		Source: annotationVariant			
External Annotations		Source: STTA_PROD_MAN_ANNO_MDL			
UI.Chart#BulletChartQualifierC		ChartType: Donut; Measures: Width			
UI.Chart#SpecificationWeightBulletC		ChartType: Donut; Measures: Weight			
Title			String	Product Weight Specification	
Description			String	No navigation with qualifier	
ChartType			EnumMember	Donut	
Measures					
Item			PropertyPath	Weight	
MeasureAttributes					

Here, you perform the following actions:

Edit annotations in your local annotation file

- Annotation modeler provides an overview of the annotations that are available in your project for the given data source of the OData type. You can see the name of this data source at the top of the page, details are displayed in tooltips.
If the project is configured to use mock data, this is indicated next to the screen title. Here, you can also select the target that you want to annotate.
In the table, the annotations are displayed under the targets they apply to. Annotations that are applied to the same target are grouped in nodes according to their annotation origin. Details of the annotation origin are displayed as tooltips of the *Key Information* column.

i Note

The sequence of the nodes under each target depends on the sequence of their nomination in the `manifest.json` file. This sequence also defines which annotation overrides another one. See [Edit External Annotations \[page 321\]](#).

- You can edit *Local Annotations* from the currently open file in annotation modeler. See [Edit Annotations \[page 310\]](#). Annotations originating from other local files or external sources relevant to the same data source of OData type are non-editable and cannot be changed directly. Annotations from other data sources are not displayed in the tree and can be viewed and edited in their respective annotation files.
- You can search through all annotations defined for the currently selected targets to locate annotations that correspond to the UI elements that you want to change. This also includes the annotations defined in sources other than the current file. See [Search in Annotations \[page 318\]](#).
- You can reset annotation modeler to default view by choosing the *Reset to default view* button. This clears the search and collapses all the nodes that group annotations by origin. All your annotation changes are saved.
- You can change the sequence of records within the collections. See [Change the Order of Annotations and Collection Items \[page 318\]](#).

Define properties in your local annotation file

- By default, the mandatory properties and the annotated optional properties are displayed. Mandatory properties are marked with an asterisk (*). You can add more properties from the *Actions* column to the annotation.

→ Tip

Annotation modeler provides vocabulary descriptions of terms and properties in tooltips.

Tooltips are shown only for terms and properties that have descriptions in the OData vocabularies.

- The validation run highlights all mandatory and optional properties that do not have values added. The error tooltip states that the value for the property is missing.
- If you do not enter an ID if there are more than one facet ID fields of a given hierarchy level, the entry field is highlighted yellow and the warning tooltip states that the ID is missing.

i Note

At the beginning of a row, annotation modeler displays status icons for errors or warnings for an annotation entity or its child nodes. If the errors or warnings are caused by invalid or missing properties, the relevant property fields are highlighted in red or yellow. Highlighted fields and entries are accompanied by a tooltip that provides more information about the cause of the error or warning and how to fix it. See [Warnings and Errors in Annotation Modeler \[page 323\]](#).

- The internationalization details for language-dependent texts, such as labels, are also displayed with the properties. You can display and update the information using the *i18n Details* icon. See [Create Internationalized Labels \[page 319\]](#).
- You can concatenate several values for all terms containing the *Value* property. See [Concatenate Values \[page 320\]](#).
- You can insert parameters in any part of a URL. See [Use Parameters in URLs \[page 321\]](#).

Add new annotations

You can add the terms you want to annotate to your local annotation file. See [Add Annotation Terms, Items, and Records to the Local Annotation File \[page 313\]](#).

Override external annotations by cloning them to the local annotation file

External annotations and annotations from local annotation files, other than the annotation file that is currently open, are read-only and cannot be edited directly. To override an annotation, you can clone it to the local annotation file that is currently open in annotation modeler. Annotations can only be cloned to a local annotation file that uses the same OData data source as the external annotation file that includes the original annotation. When you clone the annotation, the clone appears in the annotation file that is currently open and you can edit it there. In this case, the original annotation is marked as overridden (strikethrough) and its local clone is used to display the data on the UI. See [Edit External Annotations \[page 321\]](#).

→ Tip

Use the [Annotation Modeler](#) and [Code Editor](#) tabs to switch between these editors. This option is only available if you select annotation modeler as your default editor for annotation files.

For more information about how to set default editors, see [Setting User Preferences \[page 70\]](#).

7.1.9.3 Prerequisites for Working with Annotation Modeler

To adapt the UI of your application with annotation modeler, your SAP Web IDE project needs to meet the following criteria:

OData Service

- You have prepared an OData service that can be addressed locally or via a remote destination; this service may or may not contain OData annotations already.
- The metadata of the OData service must include one or multiple [edm:Schema](#) definitions within the [edm:DataServices](#) element.
According to the [OData CSDL](#), your metadata file must contain a single EntityContainer.

⚠ Caution

The namespace of the OData service should not contain / (slashes). The OData specification requires namespaces to consist of one or more SimpleIdentifiers separated by dots. Slashes are not supported.

A SimpleIdentifier must start with a letter or underscore, followed by a maximum of 127 letters, underscores and digits.

i Note

Make sure that the backend system is available and that the path to it is defined correctly.

If the backend system is not available, the metadata is loaded from the copy stored in the project.

If you want to use the local metadata (for example, when you develop the service locally), use the mock data. See [Configure Annotation Modeler to Use Mock Data \[page 109\]](#). Please note that using mock data in

a productive setting, for example, when the remote system is temporarily unavailable, may lead to inconsistencies in the UI of your app. If your app uses a local copy of metadata, the app's metadata and annotations will not be synced automatically with the metadata on the backend system.

Local Annotation File

- Your project contains a valid annotation XML file that includes the `/edmx:DataServices/Schema` node.

→ Tip

If your project does not contain an `annotation.xml` file, you can create a new one or import the file from the server. See [Create New Annotation Files \[page 308\]](#) and [Import Existing Annotation Files \[page 308\]](#).

Manifest.json File

- Your project contains a `manifest.json` file.

i Note

All paths used in annotation modeler are relative to the location of the `manifest.json` file.

- The object `dataSources/mainService/uri` in the `manifest.json` file must contain a URL to your project's OData service. Annotation modeler uses this information to retrieve the service metadata.

i Note

`Sap.ui.generic.app.pages[0].entitySet` is an **optional** setting in the `manifest.json` file. It is used to automatically select an entity type in a combobox in the [Select Targets](#) dialog when you open annotation modeler. If it is not present in the `manifest.json` file, you can select the desired entity type manually.

→ Tip

For an overview of how to set the attributes in the `manifest.json` file, see [Descriptor for Applications, Components, and Libraries](#).

Neo-app.json File

If the URI of your data source service points to a remote destination (that is, it starts with a `/`), a route definition must exist in the `neo-app.json` file of your project with type `destination` and a `path` value that matches at least the first part of the URI.

Component.js File

- Your project contains a `component.js` file.
The `component.js` file of your project references the `manifest.json` file as follows:

```
sap.ui.core.UIComponent.extend("sap.ui.sample.Component", {
  metadata: {
    "manifest": "json",
  }
});
```

When the above-mentioned criteria are met, annotation modeler can be used to open and edit the annotation file.

Related Information

[Working with Annotation Modeler \[page 309\]](#)

7.1.9.4 Adding Local Annotation Files to the Project

Annotation modeler lets you add or modify UI annotations in the local annotation file. To add annotation files to your project, you can do one of the following:

- Import an existing annotation file. See [Import Existing Annotation Files \[page 308\]](#).
- Create a new annotation file. See [Create New Annotation Files \[page 308\]](#).
Afterwards, you can start working with annotations in your newly created annotation file.

→ Tip

If your project supports multiple OData services, (for example, the Overview Page Application project), you can use annotation modeler to edit annotation files originating from different sources. For each OData service you want to annotate, you need to create a separate annotation file. You can use one of the options above to create a new annotation file for each additional OData service.

Related Information

[Import Existing Annotation Files \[page 308\]](#)

[Create New Annotation Files \[page 308\]](#)

7.1.9.4.1 Import Existing Annotation Files

You can import existing annotations from the server.

Procedure

1. In your SAP Web IDE workspace, navigate to the project folder to which you want to import the annotation file.
2. Right-click the annotations folder of your project folder and choose **► Import ► Annotation File ▾** in the context menu or click the same path in the *File* menu.
3. Select a system that contains annotations, select a matching OData service, and select an annotation file.

The annotation file is imported to the folder and is available for editing if the prerequisites are met. See [Prerequisites for Working with Annotation Modeler \[page 305\]](#).

i Note

- If needed, you can rename the annotation file.
- You cannot import a file that contains no annotations.

7.1.9.4.2 Create New Annotation Files

You can create annotation files from scratch using annotation modeler.

Procedure

1. Select the folder in which you want to create the annotation file.
2. Do one of the following:
 - Right-click the annotations folder of your project folder and choose **► New ► Annotation File ▾**.
 - Choose **► File ► New ► Annotation File ▾**.
3. In the dialog box, enter a file name and select an OData service.

An empty annotation file is created, containing headers and namespaces.

7.1.9.5 Working with Annotation Modeler

You can adjust the data displayed on the UI of your application by editing your project's local annotation files in annotation modeler.

Context

To start working with annotation modeler, proceed as follows:

Procedure

1. Create a project in SAP Web IDE.
2. Check whether the [Prerequisites for Working with Annotation Modeler \[page 305\]](#) have been met.

i Note

If you work with a *List Report Page* or an *Overview Page Application* project, the prerequisites are fulfilled automatically during the project creation. See [Developing Apps with SAP Fiori Elements](#).

- If the prerequisites have been met, you can open and edit the annotation files in annotation modeler. See [Open Local Annotation Files in Annotation Modeler \[page 310\]](#).
- You can preview the annotated data if your project supports running the application from the workspace. See [Run Applications from the Workspace \[page 392\]](#).

Related Information

[Open Local Annotation Files in Annotation Modeler \[page 310\]](#)

[Edit Annotations \[page 310\]](#)

[Validation of the Annotation File \[page 323\]](#)

7.1.9.5.1 Open Local Annotation Files in Annotation Modeler

You can open local annotation files in annotation modeler.

Annotation Modeler as the Default Editor for Annotation Files

If you set annotation modeler as the default editor for annotations, you can open local annotation files in annotation modeler by double-clicking them.

→ Tip

If you select annotation modeler as your default editor for annotations, you can switch between the code editor and annotation modeler at the bottom of the page. This option is not available if the code editor is the default editor for annotations.

For more information about how to set default editors, see [Setting User Preferences \[page 70\]](#).

Open Annotation Modeler from the Context Menu of the Annotation File

1. Right-click the local annotation file of your project.
2. Choose *Open Annotation Modeler*.

7.1.9.5.2 Edit Annotations

Context

You can adjust the data displayed on the UI of your application by editing your project's local annotation files in annotation modeler.

Items from external annotation files and from local annotation files other than the annotation file that is currently open are read-only. For more information about editing read-only elements, see [Edit External Annotations \[page 321\]](#).

⚠ Caution

Some smart controls may have specific prerequisites for defining their properties with annotations. As annotation modeler is a generic annotation editor, it does not validate against control-related restrictions or prerequisites.

Please check the documentation of the respective controls before you start annotating: [UI development toolkit for HTML5 - Demo Kit](#).

i Note

Annotation modeler does not support entity type properties of complex type.

Procedure

1. Find the term you want to edit and expand it. Make sure it is located under the *Local Annotations* node of the desired target. To display additional targets, see [Select Targets \[page 312\]](#).

By default, the mandatory and the already defined optional properties are displayed under this term.

2. In the *Actions* column, choose the *Add* icon and select the annotation properties to be defined.
3. Use the *Expression Type* and *Value* columns to enter *Property* values.
 - In the *Expression Type* column, choose from the expression types that are available in the vocabulary. For language-dependent texts that appear in the application, such as labels, choose *String (i18n)*.
 - In the *Value* column, you can either type in a new value or select a value that already exists in the project. For the values of expression type *AnnotationPath*, you can then use the *Navigate to annotation* icon to navigate to the referenced annotation. In this case, additional targets may be added to the view.

i Note

When annotating function imports or function import parameters, you can only type in the values of the path expression type. Annotation modeler does not provide any input help for function import or function import parameter paths.

For expressions of type *EnumMember*, if available in the vocabulary, descriptions for the values are shown as tooltips.

4. Use the *Edit Qualifier* column to uniquely define annotation terms. The *Edit Qualifier* icon is displayed only if qualifiers are applicable for the annotation.

i Note

If an annotation term appears in the local annotation file more than once, you have to enter a qualifier or an ID (for facets) for all but one of the annotation terms so that they are uniquely identified.

- Make sure that the entered **ID** is unique among all annotations.
- Make sure that the **qualifier** is unique for the annotation term.

5. Use the *Actions* column to add subnodes and to delete nodes.
6. Save your entries.
 - When you save, annotation modeler validates the entire annotation file.
 - Run your application to see the results with the preview function of SAP Web IDE.

Related Information

[Add Annotation Terms, Items, and Records to the Local Annotation File \[page 313\]](#)

[Change the Order of Annotations and Collection Items \[page 318\]](#)

[Create Internationalized Labels \[page 319\]](#)

[Concatenate Values \[page 320\]](#)

[Use Parameters in URLs \[page 321\]](#)

[Edit External Annotations \[page 321\]](#)

7.1.9.5.2.1 Select Targets

Context

To add new or update the existing annotations, you select the targets these annotations apply to.

For the supported annotation levels, see [Architecture \[page 301\]](#), section *Scope*.

Procedure

1. Choose the *Select Targets* button.

The top-level target types are displayed in alphabetical order. The entity types for targets that are already included in the annotation structure are automatically selected.

i Note

The list of targets has a hierarchical structure similar to that of the metadata. Not all nodes are visible at all times.

- To view and select the entity type properties, expand the following nodes: `Entity Types/<entity type name>/Properties`.
- To view and select the entity sets, expand the following nodes: `Entity Containers/<entity container name>/Entity Sets`.
- To view and select the function imports, expand the following nodes: `Entity Containers/<entity container name>/Function Imports`.
- To view and select the function import parameters, expand the following nodes: `Entity Containers/<entity container name>/Function Imports/<function import name>/Parameter`.

2. You can search the tree by entering a search term.

The tree is filtered and expanded to show every target matching the search. If you clear the search term, the original state of the tree is recovered.

i Note

You can search again to find additional targets to select. All your previous selections remain and become visible as soon as you clear the search.

3. Select or deselect the respective checkboxes.
4. Choose *OK*.

Results

The newly selected targets are displayed in the annotation structure. Once you have defined at least one annotation for each target, they are added to the annotation file as well.

7.1.9.5.2.2 Add Annotation Terms, Items, and Records to the Local Annotation File

Procedure

1. Select the targets your annotation should apply to. For more information, see [Select Targets \[page 312\]](#).
2. To add annotations to the selected target, proceed as follows:
 - a. Choose the *Add* icon in the *Actions* column of the underlying *Local Annotations* node level.
 - b. Choose the annotation you want to add. The annotation is inserted above the others and is selected for editing.
3. For annotation terms that are collections of primitive types, you can add items. For example, *UI.Contacts* can contain several items. To add and annotate an item, do the following:
 - a. In the row of the annotation term representing a collection of primitive types, in the *Actions* column, choose the *Add* icon.
 - b. In the *Add to* dialog box, choose *Item*. Repeat these steps for each item you want to add.
 - a. Annotate the items by choosing the *Add* icon on the item node.
4. For annotation terms that are collections of complex types, you can add records. For example, you can add records of complex type *UI.DataFieldAbstract* to *UI.LineItem*. To add and annotate a record, do the following:
 - a. In the row of the annotation term, in the *Actions* column, choose the *Add* icon.
 - b. In the *Add to* dialog box, choose the record you want to add (e.g. *DataField*). Repeat these steps for each record you want to add.
 - c. Annotate the records by choosing the *Add* icon on the record node.

Related Information

[Example: Add a Field Group \(UI.ReferenceFacet\) for Technical Data to an Object Page \[page 314\]](#)

7.1.9.5.2.1 Example: Add a Field Group (UI.ReferenceFacet) for Technical Data to an Object Page

This section explains how to change the UI of your model-based application based on the sample task of adding a field group for technical data to an object page.

Context

In this sample project, we are working with a *List Report Page* project that uses an *Object Page* template.

Our goal is to add a new field group for the technical data of a product to the *General Information* section, and then edit the properties of this new field group. The new *Technical Data* field group should be displayed next to the existing *Product Information* field group.

i Note

In this example, the field group that includes the technical data already exists in the backend. We will create a new *UI.ReferenceFacet* and reference to the target *UI.FieldGroup*.

The screenshot shows the SAP Object Page for a '10" Portable DVD player' (HT-2001). The page is divided into several sections: 'General Information', 'Sales Data', and 'Price'. The 'General Information' section contains details about the product, including its category, sub-category, and supplier. A large green rectangular box highlights a specific area within the 'General Information' section, likely indicating the location where the new 'Technical Data' field group will be added. Below the 'Sales Data' section, there is a bar chart titled 'Delivery Year / Delivery Month' showing sales data over time.

→ Tip

In applications using the object page template, the following applies:

- *UI.CollectionFacets* create main sections in the contact area of your application's UI (in this example: **Product Information**).
- *UI.ReferenceFacets* within a *UI.CollectionFacet* display the subsections that present the content of the referenced terms (in this example: the field group **General Information**).
If *UI.ReferenceFacet* is not annotated as a child node of a *UI.CollectionFacet*, but appears on the same level in the annotation model. *UI.ReferenceFacet* creates its own main section that presents the content of the referenced terms.

For more information about the Object page view, see [Object Page View](#).

Procedure

1. Open your project's annotation file with annotation modeler. See [Open Local Annotation Files in Annotation Modeler \[page 310\]](#).
2. In annotation modeler, choose the *Select Target* dialog and select the entity type related to the object page you want to update. For more information, see [Select Targets \[page 312\]](#).
3. Under the selected entity type, expand the *Local Annotations* node.
4. Expand the *UI.Facets* and the *UI.CollectionFacet* nodes to which you want to add a new *UI.ReferenceFacet*.

To ensure that you add the new annotation at the right place, check the *Key Information* column.

The entry in the *Label* property field of the *UI.CollectionFacet* represents the main section heading on the UI of your application. In this example, the main section header is **Product Information**.

5. Add a new *UI.ReferenceFacet* annotation.
 - a. In the table row of the *Facets* node, choose the *Add* icon.

A dialog box opens in which you can select all annotations and records that can be added as child nodes of the *UI.CollectionFacet*.

- b. In the dialog box, select *ReferenceFacet* and choose *OK*.

A new *UI.ReferenceFacet* annotation is added under the *Facets* node. The mandatory properties of the new *UI.ReferenceFacet* are displayed in the table below the new *UI.ReferenceFacet*. In this example, for the *UI.ReferenceFacet*, *Target* is mandatory.

Annotation Structure
 OData Data Source: mainService

Select Targets

Node	Edit Qualifier	Key Information	Expression Type	Value	Actions
Entity Types					
ANNOFD_C_PD_PRODUCTType					
Local Annotations		Source: annotation3			+
UI.Facets					+
UI.CollectionFacet		ID: GeneralInfo; Label: General Information			+
Label			String	General Information	+
ID			String	GeneralInfo	+
Facets					+
UI.ReferenceFacet		ID: SecondFacet; Label: Second Facet; Target: (to_ProductSalesData@...			+
UI.ReferenceFacet		Target: (@UI.FieldGroup#TechnicalData)			+
Target			AnnotationPath	Navigation @UI.FieldGroup#Techni...	+
External Annotations		Source: ANNOFD_PROD_MAN_ANNQ_MDL			
External Annotations		Source: mainService			

Annotation Modeler Code Editor

6. Before defining the property values, add the optional properties you require for your annotation.
 - a. In the *UI.ReferenceFacet* row, choose the *Add* icon.

A dialog box opens in which you select all properties that apply for the annotation.

- b. Select *Label* and *ID* and choose *OK*.

The *Label* and *ID* rows are displayed under *UI.ReferenceFacet*.

7. Define the heading for the new field group in the *Label* row. In this example, we enter **Technical Data**.
 - a. In the *Expression Type* column, make sure that *String (i18n)* is selected.
 - b. In the *Value* column, enter *Technical Data*.
8. Define a unique value for the new *UI.ReferenceFacet* in the *ID* row. In this example, we apply the **TechData** ID.
 - a. In the *Expression Type* column, make sure that *String* is selected.
 - b. In the *Value* column, enter *TechData*.

Each *UI.ReferenceFacet* requires a unique ID to ensure that it is displayed correctly on the application UI. This means that only one *UI.ReferenceFacet* in the annotation model can have an empty ID field, each additional *UI.ReferenceFacet* requires a unique value for the property *ID*.

If you create two or more *UI.ReferenceFacets* with the same ID, annotation modeler issues an error when you try to save your project or when you switch to another annotation.

9. Assign the *UI.FieldGroup* as content for *UI.ReferenceFacet*. In this example, we reference the *UI.FieldGroup* with the qualifier *TechnicalData* from the OData entity type *SEPMRA_C_PD_Product*.
 - a. In the *Target* row, in the *Annotation* field, select *UI.FieldGroup#TechnicalData*.

The term following @ defines the annotation type. The term following # defines the qualifier. This means that *@UI.FieldGroup#TechnicalData* references the *UI.FieldGroup* annotation with the qualifier *TechnicalData*.

Note

In the backend of this sample project, a *UI.FieldGroup* that contains all necessary technical data already exists. We reference this *UI.FieldGroup* in our new *UI.ReferenceFacet* to display the fields of this field group on the UI of the application. You can see how the referenced annotation is configured by

choosing the *Navigate to annotation* icon. In this example, the *External Annotation* node expands and *@UI.FieldGroup#TechnicalData* is selected to draw your attention.

- If you leave the *Navigation* field empty, you can reference annotations only from the currently selected OData entity type.
- The content targets that are available in the *Annotation* dropdown menu depend on the selected *Navigation*.
To reference an annotation from another OData entity type, select the entity type in the *Navigation* field before assigning the content target in the *Annotation* field. You can select only OData entity types that have a one-to-one relationship with the entity type that is currently selected.

10. When you have finished editing, save your project.

11. Run your application to see the changes you've made.

In this example, we have added a subsection to our main section *Product Information* that displays the technical data of the sample product:

The screenshot displays the SAP Fiori user interface for a product page. At the top, the SAP logo and navigation icons are visible. The main header shows the product name '10" Portable DVD player' and its ID 'HT-2001'. Below this, there are three tabs: 'PRODUCT INFORMATION', 'SALES DATA', and 'PRICE'. The 'PRODUCT INFORMATION' tab is active, showing 'General Information' and 'Technical Data'. The 'Technical Data' section is highlighted with a green box and lists the following details: Base Unit: EA, Product Height: 3.000 CM, Product Width: 30.000 CM, and Product Depth: 18.000 CM. Below the technical data, there is a 'Sales Data' section with a bar chart titled 'Delivery Year / Delivery Month' showing sales data for various months and years.

→ Tip

To find out more about how facets annotations are used in applications based on the ObjectPage template, see [Sections](#).

7.1.9.5.2.3 Search in Annotations

To quickly locate annotations that correspond to UI elements, search through the annotation modeler stack of your selected target.

Context

When extending applications, you want to locate the relevant annotations quickly.

For example, you want to edit a *Technical Data* section on the application UI.

Therefore, in annotation modeler, you can search for matching texts in the following columns:

- *Node*
- *Edit Qualifier*
- *Value*

Annotation modeler searches through the entire annotation stack for the given OData data source, that is, in both local and external annotations.

Procedure

1. Choose the *Select Target* dialog and select the targets you want to search through. For more information, see [Select Targets \[page 312\]](#).
2. In the search field, enter the text you want to search for, and press .

For example, enter **Technical Data**, and press .

The number of matches is displayed next to the search field.

In the table, the first search result is selected automatically. Further search results are highlighted.

3. Edit the node as required.

7.1.9.5.2.4 Change the Order of Annotations and Collection Items

Context

In annotation modeler, you can change the sequence of collection items. For example, you can change the sequence of *UI.DataFields* within a *UI.LineItem* to change the order of the columns in your application.

You can also re-order local annotations, for example, to structure your annotations as required. The changed order of the annotations is reflected in the code of the XML file and can also be seen in the code editor. The order of the items on the UI is not affected.

Procedure

1. In a local annotation node, select the row you want to move.
2. To move the row one position up or down, choose *Move up* or *Move down*.
 - With each click, the row moves one position.
 - The row remains selected, so you can click several times to move the row multiple positions.

7.1.9.5.2.5 Create Internationalized Labels

Prerequisites

For language-dependent texts that appear in the application, such as labels, the *String (i18n)* expression type is defined in the vocabulary for applying internationalization (i18n).

Context

When defining texts as *String (i18n)* expression type, the text is automatically added to the `i18n.properties` file of your project and prepared for translation into other languages.

In addition, the internationalization icon is displayed in the *Value* column and you can edit the values for internationalization.

Procedure

1. Choose the *String (i18n)* expression type and enter a text in the *Value* column.

The i18n key and other text properties are automatically generated and stored in the `i18n.properties` file of your project.
2. To change the automatically generated text properties (except for the text key), in the *Value* column, choose the *i18n Details* icon.

The *i18n Details* dialog box opens.
3. Edit the i18n details as required. The available text types are read from the `i18n.properties` file that is defined in the `manifest.json` file.

4. Change the text key, if required, in the `i18n.properties` file.

Related Information

[Translatable Texts](#)

7.1.9.5.2.6 Concatenate Values

To create complex expressions, you can concatenate several properties of type *Value*. You can use delimiters to separate concatenated values.

Context

Concatenated paths and values are displayed in the *Key Information* column.

Procedure

1. In the *Value* row, in the *Actions* column, choose the *Add* icon.
The *Add To Value* dialog box opens.
2. Choose *Component* and choose *OK*.
The previous value is inserted as first component. The component is inserted as second component.
3. Edit the *Expression Type* and *Value* columns as required.
4. To concatenate more values, add another component.
5. To set delimiters, add another component and enter the delimiter string. Use the *String (i18n)* expression type for delimiters that are internationalized. Use the *String* expression type for delimiters such as space or semicolon+space.

Related Information

[Binding Paths: Accessing Properties in Hierarchically Structured Models](#)

7.1.9.5.2.7 Use Parameters in URLs

To make a URL more flexible, you can insert parameters in any part of a URL.

Context

You can use parameters in properties that can express URLs, for example, in *IconUrl*.

For each parameter you insert, you can then define flexible targets.

Procedure

1. In the property row, enter the URL as *String* expression type. For each parameter, use curly braces {}, for example, `www. {SLD} test. {TLD}`.
2. Press .
- For each parameter, a new node is created.
3. For each parameter, in the *Value* column, enter the target of the URL parameter.

7.1.9.5.2.8 Edit External Annotations

Annotations defined in the backend or in other external sources cannot be edited directly but can be overridden by cloning them to local annotations. When you clone an annotation from the backend to the local file, the clone is inserted in the currently open annotation file where you can edit it.

Prerequisites

Verify that the `manifest.json` file contains the *dataSources* object with an array of one or more *ODataAnnotation* types. See [Prerequisites for Working with Annotation Modeler \[page 305\]](#).

i Note

Each of these objects has the *uri* and *localUri* properties. When you open the annotation file in annotation modeler, it loads all annotations that are defined in the *uri*. If the connection to the back end is not available, annotation modeler shows an error. For more information, see [Troubleshooting \[page 325\]](#).

Context

Annotations can only be cloned to the currently open local annotation file that uses the same OData data source as the external annotation file that includes the original annotation.

Procedure

1. Select the targets for which you want to edit external annotations. For more information, see [Select Targets \[page 312\]](#).
2. Expand the *External Annotations* node under the target and navigate to the external annotation you want to override.
3. From the *Actions* column, choose *Clone for overriding*.

Annotation modeler clones the external annotation from the remote annotation source to the local annotation file that is currently open in annotation modeler. The cloned term and its properties are selected. Now you can edit the clone in the local annotation, see [Edit Annotations \[page 310\]](#).

As each of the sources now contain a different value for the same term, you have to be aware of which of the values will affect the UI:

- In the *External Annotation* section, an overridden annotation and its child nodes are displayed in gray and struck through and marked as *(overridden)*.

i Note

Individual collection records, such as *UI.DataField* of *UI.LineItem* can only be cloned with the whole collection. In this case, choose the *Clone for overriding* icon for the parent term to clone the record together with the whole collection. As soon as the collection is cloned, all its records are marked as overridden. In your local annotation file, you can then delete the collection records that you do not need, or you can change their properties as required.

- If you change the qualifier of a cloned annotation, it is treated like a new one and is used along with the initial one. For example, if you clone the *UI.LineItem* annotation and change its qualifier, both *UI.LineItem* entities are active.
- When you delete a clone of an external annotation from the local annotations, the external annotation is no longer overridden and affects the UI.
- If your project supports previewing the changes by using the *Run* function in SAP Web IDE, you can also run the component and check which annotation affects the UI.

i Note

Terms from different annotation files originating from the same OData data source override each other in the same sequence as specified in the `manifest.json` file. The last annotation specified in the `manifest.json` file is the active one.

7.1.9.5.3 Validation of the Annotation File

Annotation modeler checks your `annotation.xml` file for inconsistencies in vocabulary specifications and other issues that can cause an unstable UI.

i Note

- When you add new annotations to your `annotation.xml` file, annotation modeler validates their property values as soon as you leave the field after editing.
- To validate the entire `annotation.xml` file, save your project.

Related Information

[Warnings and Errors in Annotation Modeler \[page 323\]](#)

[Check Error Descriptions in the Problems View \[page 324\]](#)

7.1.9.5.3.1 Warnings and Errors in Annotation Modeler

Annotation modeler validates the local annotation file based on the vocabulary definitions and OData standards. Error and warning messages notify you about incompatibilities during the development process and support you in resolving them.

- When you save a project, the local annotation file is validated.
- When you leave a field, the field is validated.
- If the validation process discovers an error, such as an empty value for mandatory property or wrong input format, error and warning statuses are updated: the fields are highlighted in red, and the nodes that contain them are marked with red vertical bars.
- Errors and warnings are propagated to the parent elements in the tree structure. If there are errors and warnings in the child nodes, the parent node is marked with a red vertical bar.

→ Tip

To obtain an overview of all errors and warnings in your local annotation file, open the *Problems* view in SAP Web IDE.

7.1.9.5.3.2 Check Error Descriptions in the Problems View

In the *Problems* view, you can check your `annotations.xml` file for issues resulting from validation checks run by the annotation modeler. This includes inconsistencies regarding vocabulary specifications and other issues that can cause an unstable UI.

Procedure

1. To check the `annotations.xml` file for issues specific to the annotation modeler, open the `annotations.xml` file in the annotation modeler.

i Note

The *Problems* view displays issues specific to the annotation modeler only if the `annotations.xml` file is opened in the annotation modeler.

2. Open the *Problems* view. See [Using the Problems View \[page 242\]](#).

To navigate to errors in annotations of another entity type, switch to the entity type before clicking on the navigation link in the *Problems* view.

7.1.9.6 Troubleshooting

Error Message	Issue Description and Solution
<p>OData metadata cannot be loaded from destination <destination name>.</p> <p>This error can appear for several reasons:</p> <ul style="list-style-type: none">• The OData Service URI and the local URI are missing or defined incorrectly. Please check the OData Service URI and the local URI in the manifest.json file (app descriptor) of this project.• The destination is defined incorrectly in SAP BTP Cockpit.	<p>The annotation file cannot be loaded because the OData metadata file cannot be loaded from the backend or the local copy.</p> <div data-bbox="821 571 1396 862"><p>i Note</p><p>Annotation modeler loads metadata based on the URI entry defined in the <code>manifest.json</code> file. This is usually a URI pointing to a backend destination. The local URI setting is used when the remote metadata is not accessible or when the annotation modeler is configured to use the mock data. If none of these settings is correctly defined, annotation modeler cannot be loaded.</p></div> <ul style="list-style-type: none">• Check the URI and local URI for the data sources of type OData in the <code>manifest.json</code> file (app descriptor).• Check the destination that is referred to by the <code>neo-app.json</code> entry. The name of the destination must be the same as defined in SAP BTP Cockpit. Also, make sure that the <code>WebIDEUsage</code> property contains either <code>odata_abap</code> or <code>odata_xs</code>, but not both. You can also use destinations of <code>WebIDEUsage odata_gen</code>. If your destination addresses a dedicated service with its full URL, you must specify the additional <code>WebIDEAdditionalData</code> property with the <code>full_url</code> value. <p>For more information, see Prerequisites for Working with Annotation Modeler [page 305] and Connect to ABAP Systems [page 57].</p>

Error Message

Issue Description and Solution

The file `<file name>` cannot be edited, because it is not registered in the `manifest.json` file (app descriptor).

- **The local annotation file cannot be opened.**

The annotation file cannot be loaded because it is not registered in the app descriptor (`manifest.json` file) of this project. This happens if the annotation file has not been added to the project in the standard way (using the project wizard, [New Annotation File Settings](#) or [Import Annotation File](#)). Please define the annotation file for an existing data source in the `manifest.json` file.

See [Modifying the Application Descriptor Configuration \[page 121\]](#) for further instructions.

- **The external annotation file cannot be opened.**

External annotation files such as the back-end annotations of the OData service cannot be opened in annotation modeler because they are not supposed to be used productively. They are copies of the back-end annotations of the OData service. Changes in this file are not loaded when you run the application, because the application uses the external version of this local copy.

To modify external annotations, clone them to your local annotation file to edit them. For more information, see [Edit External Annotations \[page 321\]](#).

If your project does not have a local annotation file, create a new local annotation file. For more information, see [Create New Annotation Files \[page 308\]](#).

The annotation xml file cannot be loaded because it contains xml format errors. Open the Problems view to view and fix the errors.

The xml file is invalid due to xml format errors and cannot be opened in annotation modeler. Open the *Problems* view to view and fix the errors. Then, reopen the annotation file in annotation modeler.

For more information, see [Warnings and Errors in Annotation Modeler \[page 323\]](#).

Invalid file: DataServices definition is missing.

Check that the annotation xml file includes an `/edmx:DataServices/Schema` node.

You can create a valid file by creating an annotation file.

For more information, see [Create New Annotation Files \[page 308\]](#).

There is no OData data source defined in the `manifest.json`. Please define an OData data source.

Check the relevant entry of the `dataSources` setting of type OData in the `manifest.json` file (app descriptor).

Error Message	Issue Description and Solution
<p>OData metadata cannot be loaded from destination <destination name>. The destination is not correctly maintained at the SAP BTP.</p>	<p>Check the destination that is referred to by the <code>neo-app.json</code> entry. The name of the destination must be the same as defined in SAP BTP cockpit. Also make sure that the <code>WebIDEUsage</code> property contains either <code>odata_abap</code> or <code>odata_xs</code>, but not both.</p> <p>You can also use destinations of <code>WebIDEUsage</code> <code>odata_gen</code>. If your destination addresses a dedicated service with its full URL, you must specify the additional <code>WebIDEAdditionalData</code> property with the <code>full_url</code> value.</p>
<p>Connection to system <system name> failed</p>	<p>This error can appear for several reasons:</p> <ul style="list-style-type: none"> • The login data is incorrect: Make sure that you have entered the correct credentials for accessing the back-end system. • The connection to the backend is not available: Make sure that the back-end system is available. • The destination configured in SAP BTP is incorrect: Check the destination configuration. You can also use destinations of <code>WebIDEUsage</code> <code>odata_gen</code>. If your destination addresses a dedicated service with its full URL, you must specify the additional <code>WebIDEAdditionalData</code> property with the <code>full_url</code> value. <p>For more information, see Prerequisites for Working with Annotation Modeler [page 305] and Connect to ABAP Systems [page 57].</p>

7.1.10 Storyboard

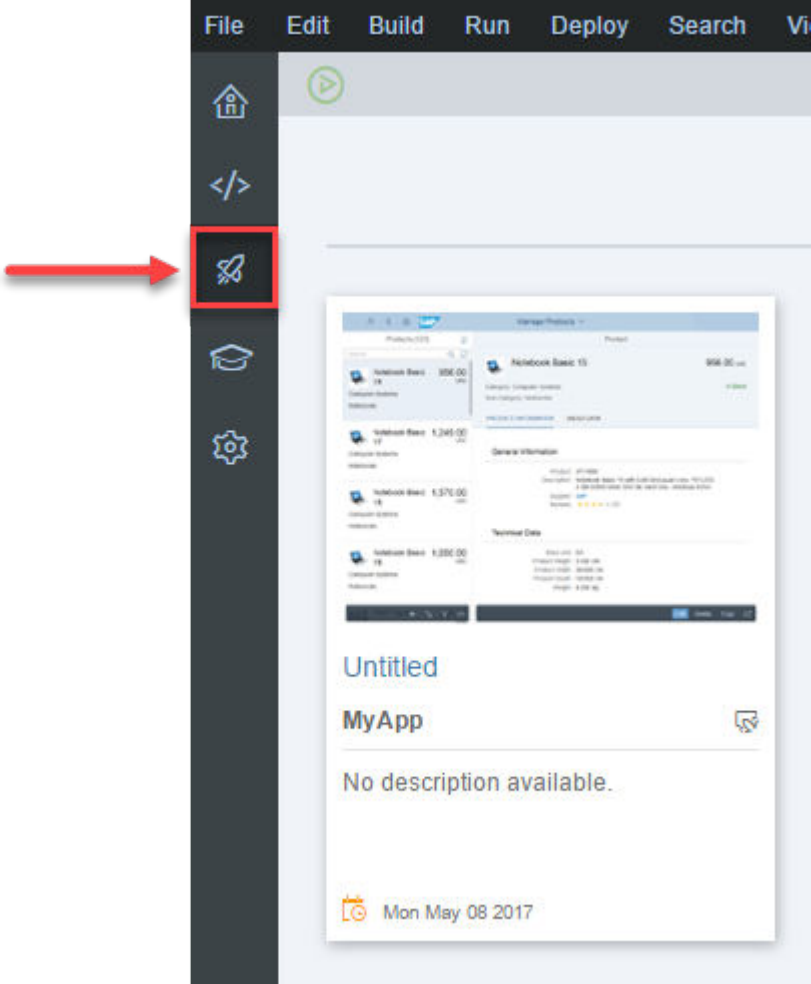
The storyboard provides a visual representation of the application's UI including its pages (views), navigations, and the services and entities that it uses.

Prerequisites

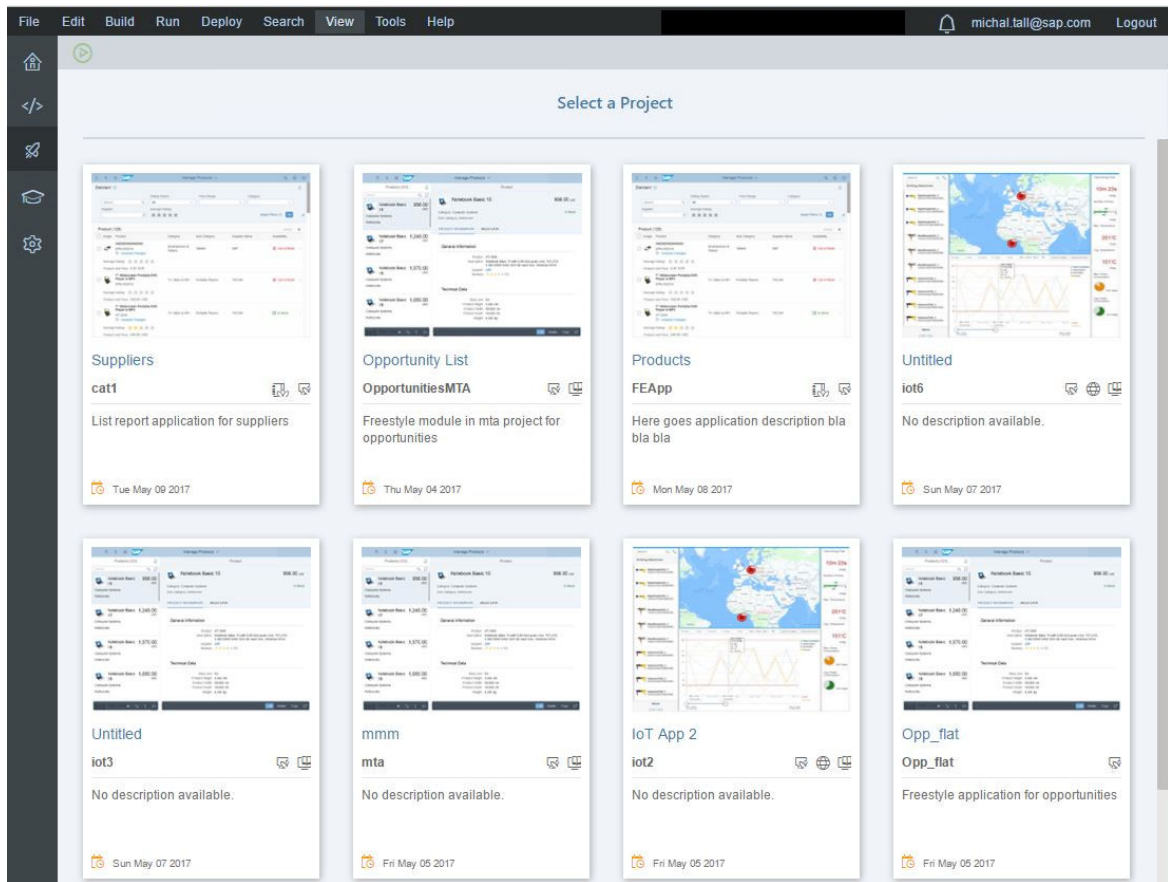
- You have enabled the *Storyboard* perspective in the *Preferences* perspective, on the *Features (Extensions)* page.
- You have created a freestyle or SAP Fiori Elements List Report project.

Procedure

- 1. Open the *Storyboard* perspective from the left sidebar.



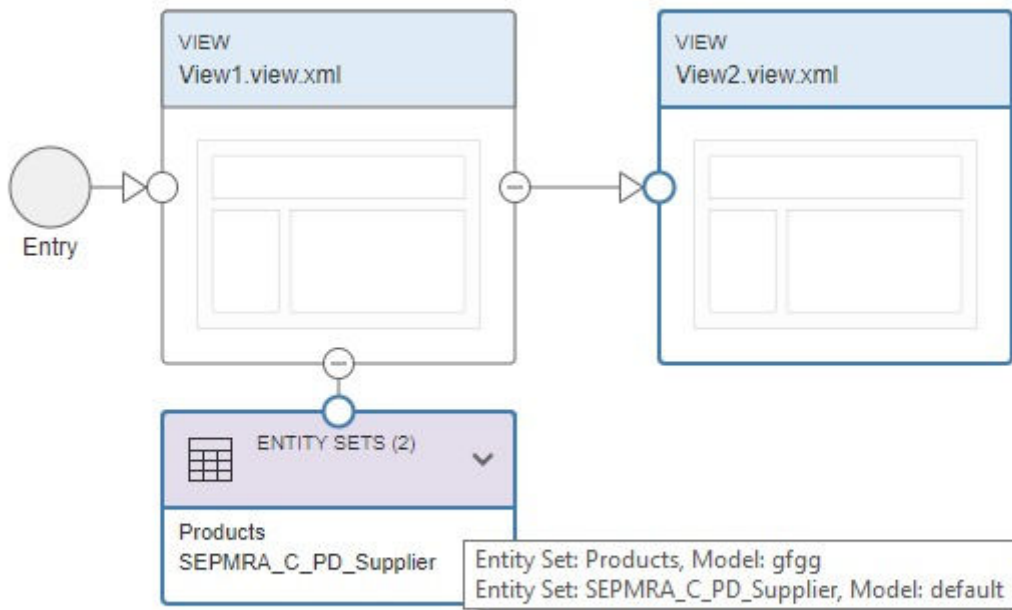
A preview of all the projects supported by the storyboard in your workspace is displayed.



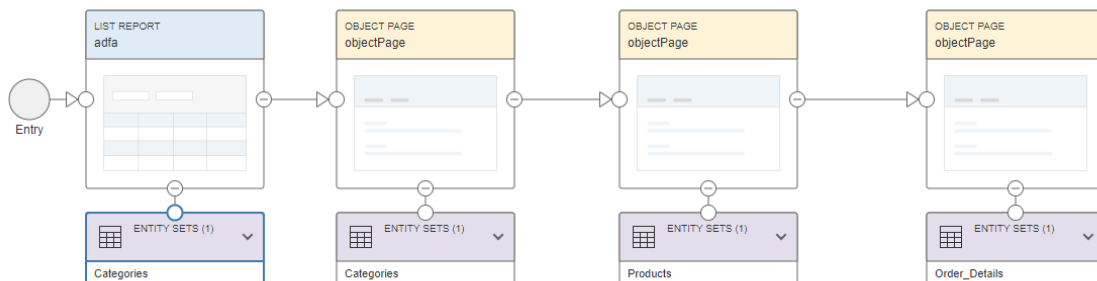
2. Select the desired project.

The *Storyboard* tab opens, showing the project flow with the navigations and respective entity sets. When you hover your mouse over an entity set, the model origin appears in a tooltip.

For **freestyle** projects, views and entity sets are displayed as shown below.



For **SAP Fiori Elements List Report** projects, pages and entity sets are displayed as shown below.



7.1.10.1 Add View to Freestyle Project

You can add a new view to a freestyle project in the *Storyboard* perspective.

Prerequisites

- You have enabled the *Storyboard* perspective in the *Preferences* perspective, on the *Features* tab.


- You have created a freestyle project.

Procedure

1. In the *Storyboard* perspective, choose *New View*.
2. In the *Template Customization* wizard, in the *View Name* field, enter a name for the new view.
3. Choose *Next*, then choose *Finish*.

Results


The new view appears in the *Storyboard* perspective, and in the *Development* perspective, a view and controller are created in your workspace and the manifest file is updated accordingly.

You can now edit the view in the layout editor by clicking the *Edit*  button, or you can go to the *Design* tab. For more information, see [.Layout Editor \[page 256\]](#).


7.1.10.2 Add Page to SAP Fiori Elements Project

You can add a new child page to an SAP Fiori Elements List Report type project in the storyboard.

Procedure

1. To add a new child page in the storyboard, click the *Add Page*  icon.
2. In the *Add Object Page* window, fill in the relevant fields and click *OK*.

The new child page now appears in the storyboard, including a navigation to the parent page. In addition, in the *Development* perspective, a page and controller are created in your workspace and the manifest file is updated accordingly.

You can now edit the new page in the SAPUI5 Visual Editor by clicking the *Edit*  button, or by going to the *Design* tab. For more information, see [SAPUI5 Visual Editor Usage](#).

7.1.10.3 Configure a Navigation Between Views

In the *Storyboard* perspective, configure a navigation between views of a freestyle project.

Prerequisites

- You have created a freestyle project in the layout editor that has at least two views. For more information, see [Add View to Freestyle Project \[page 330\]](#).

Context

You can configure a navigation between two views in the *Storyboard* perspective and determine whether to propagate the binding context from the start view into the target view.

Procedure

1. In the *Storyboard* perspective, select the view where you want to start the navigation and then click the *Configure Navigation* button.
2. In the *Configure Navigation* dialog box, do the following.
 - a. In the *Control* dropdown list, select the control that will trigger the navigation event.
 - b. In the *Event* dropdown list, select the event to attach to the control.
 - c. In the *Navigate To* dropdown list, select the target view.
 - d. To propagate the data source that is bound to the control to the target view, select the *Propagate context binding* checkbox. Alternatively, you can bind the target view to the required data source in the layout editor.

For more information, see [Binding Data \[page 268\]](#).

3. Save your configuration.

The navigation is now configured and you can run it as required.

Related Information

[Create a Navigation Between Views \[page 265\]](#)

7.1.10.4 Delete View

You can delete a view from the *Storyboard* perspective.

Prerequisites

- You have created one or more views in your project.

Procedure

1. In the *Storyboard* perspective, select the view you want to delete.
2. Press `Delete` on your keyboard and then choose *Yes* in the *Confirmation Needed* dialog box.

Results

The view is removed from the *Storyboard* perspective and the view and its controller are deleted from the Workspace and the manifest file is updated accordingly.

7.1.11 Extending an SAP Fiori Application with SAPUI5 Visual Editor

Developers can change, adapt, and extend the user interface of SAPUI5 applications.

See the following topics for more information:

- [Adaptation Project \(for SAP Fiori Elements-Based and Freestyle Applications\) \[page 333\]](#)
- [Extending SAP Fiori Elements-Based Applications \[page 355\]](#)

7.1.11.1 Adaptation Project (for SAP Fiori Elements-Based and Freestyle Applications)

An adaptation project lets you create an app variant for an existing SAP Fiori elements-based application or freestyle application, on S/4 HANA on-premise ABAP system and provides extension capabilities for UI5 controls. You can create an app variant that includes the changes that you make to the source application and to the variant itself.

Extensions or changes are made via SAPUI5 flexibility possibilities and are modification-free. This modification-free approach allows you to reference the source application and its artifacts instead of modifying the source

artifacts itself. An adaptation project behaves the same as any other SAP Web IDE application project – start as a preview, adapt using SAPUI5 Visual Editor, and deploy from SAP Web IDE. The artifacts on an adaptation project represent only the changes that you make and not the entire application that you reuse.

The app variant refers to the original application, but contains a separate set of changes created in the adaptation project. Also an application ID is defined for the variant and needs separate registration in Fiori Launchpad. The option to create app variants based on existing apps allows you to keep both instances running and keeps the original application untouched. You can configure both application instances as different tiles on FLP. You can work with both applications or assign the app variant to a different set of users.

Safe Mode and Advanced Mode

Safe mode allows an app variant to stay fully compatible with future upgrades of the source application. When an app variant is in safe mode, you can only make semantic changes to the UI elements. Semantic changes are the changes that you make during the application runtime. Safe mode also means fewer freestyle capabilities. When you launch the application using SAPUI5 Visual Editor in safe mode, you see only the *Canvas* and the *Outline* pane. The *Properties* pane is disabled but still visible.

In Advanced mode, you can change the control properties too.

When you create an adaptation project, by default, safe mode is enabled. You can opt out of safe mode while you are creating an adaptation project or while adapting the UI.

→ Tip

You cannot return to safe mode after switching to Advanced mode.

A typical workflow for creating an app variant includes the following steps:

1. [Create an Adaptation Project \[page 335\]](#).
2. [Adapt the UI \[page 336\]](#) of the app variant.
3. [Preview the App Variant \[page 352\]](#).
4. [Deploy the App Variant \[page 353\]](#) to ABAP repository for consumption.

→ Remember

Do not create an adaptation project from an app variant or base application residing on the same layered repository on the ABAP platform. Changes from a source application on the same layered repository cannot be copied to an adaptation project.

See [Known Issues \[page 361\]](#) for a list of open issues.

7.1.11.1.1 Create an Adaptation Project

Create an adaptation project from an existing oData v2 SAP Fiori elements-based and SAPUI5 freestyle applications available in the on-premise ABAP systems.

Prerequisites

Currently adaptation projects are supported only on on-premise ABAP systems with the minimum required version of software component being SAP_UI 7.53 SP00. Have in mind that, if you have SAP_UI 7.53 SP00 installed, you need to apply the SAP note [2615176](#), and implement the manual correction instructions to enable the system. If you are on SP01 or higher, no actions are needed.

The minimum SAPUI5 version must be 1.62.

i Note

Have in mind that you can't create adaptation project for an app variant that has already been created and deployed with an adaptation project.

Adaptation project does not support as basis application:

- Fiori Elements applications using OData V4
- Applications without manifest file
- Scaffolding applications
- Applications that require a mandatory parameter to start

Procedure

1. Choose **File > New > Adaptation Project**.
2. Enter a name for the project and a title for the app variant, then click **Next**.
By default, **New App Variant** is selected to ensure that you don't change the source application.
3. Choose a system and then select the SAP Fiori elements-based application (LR, ALP, OVP, or OP) you want to adapt.
In the **Source** list, **SAPUI5 ABAP Repository** is the default source.
4. Choose if you want to use the **Safe Mode**.
5. Click **Next** or **Finish** to create an adaptation project.

7.1.11.1.2 Adapt the UI

Make changes to the UI elements of the adaptation project.

Prerequisites

[Create an Adaptation Project \[page 335\]](#).

Procedure

1. In your workspace, select *SAPUI5 Visual Editor* from the context menu.

SAPUI5 Visual Editor launches and the application loads in the editor for you to make the changes when you switch to Edit mode from Preview mode.

If an Overview Page application doesn't launch in the editor, follow the steps mentioned in [Known Issues \[page 361\]](#).

2. Navigate to the page containing the UI element you want to change.
3. From the editor header, select *Edit*.

See [SAPUI5 Visual Editor Usage \[page 355\]](#) or [UI Editing Options \[page 358\]](#) to know more about adapting the UI elements.

In safe mode, any changes you make to the app variant are read-only. Also, you can only make semantic changes and cannot make any control property change.

4. To switch to the Advanced mode, click *Safe Mode* in the editor header. In the pop-up, disable the *Safe Mode* check box. Click *Apply*.

7.1.11.1.2.1 Create Control Variants (Views)

As a control developer you can create control variants within the Visual Editor.

Prerequisites

The minimum SAPUI5 version must be 1.73.

Make sure that you have enabled your application for UI adaptation and have the `sap.ui.fl.variants.VariantManagement` control as part of the page (XML view) that is going to be adapted. For more information, see [SAPUI5 Flexibility: Enable Your App for UI Adaptation](#) and the `sap.ui.fl.variants.VariantManagement` [API Reference](#).

Context

The SAPUI5 Visual Editor allows you to create, load, change, and save control variants. Typically, you would create variants of a control to give different options for the end user for better usability in special use cases when they might need (or not) a specific part of control.

i Note

You can define variants for specific controls on the user interface. Have in mind that variants are also called "views", usually in a more nontechnical context, for example, on the user interface.

Procedure

1. Open the *SAPUI5 Visual Editor*.

In your workspace, choose *SAPUI5 Visual Editor* from the context menu. When loaded, switch to Edit mode, to be able to make changes.

2. Adapt variants.

For managing of the variants, you will be working with the `sap.ui.fl.variants.VariantManagement`'s control variant selector. Have in mind that the selector is situated in the page, as it is defined in the XML view. Right click the selector to open the popup, where you can perform the following:

Create new variant.

Choose *Duplicate* and enter a name. This creates your new variant being a copy of the current one, and switches to it as your current variant.

When you make a semantic change and save it, a change file is created in your project. The change is associated with your currently selected variant and has the `variantReference` property pointing to the ID of this variant.

Switch between your variants.

Choose *Switch Views* and select the variant you want. The selected variant is grayed out. You can also do this in navigation mode, by clicking the arrow and selecting the view.

Rename the variants.

Double-click on the variant name. Another way to do this is to right-click on the view selector and choose *Rename*.

Manage the variants.

Choose *Manage Views* to open the dialog, where you can:

- Define a default variant, which is displayed when the app is started.
- Rename and delete variants. This action depends on your authorizations. As a rule of thumb, you can always rename and delete variants that you have created. The standard variant can never be deleted.
- Add or remove favorite views for end users.

7.1.11.1.2.2 Add Fragments to an Extension Point or Aggregation

You have the possibility to add fragments to an extension point or to an aggregation.

Content

- [Add Fragments to an Aggregation \[page 338\]](#)
- [Add Fragments to an Extension Point \[page 339\]](#)

Add Fragments to an Aggregation

Prerequisites

- You are using SAPUI5 version 1.56 or above.
- You are in Advanced mode of SAPUI5 Visual Editor.

Procedure

1. In your workspace, from the project context menu, choose *SAPUI5 Visual Editor*.
The application opens in the canvas in preview mode.
2. From the editor header, choose *Edit*.
3. On the UI, select a control (such as a smart-filter bar or an overflow toolbar), and from its context menu, choose *Add Fragment*.

In the dialog box that opens, the default target aggregation and last index are selected. A list displays the fragments available for the selected target aggregation. You can also create fragments. Choose the target aggregation and the index from the list where you want to add the fragment. You cannot reuse the same fragment multiple times.

i Note

The index is disabled for **controlConfiguration** aggregation in the control Smart Filter Bar. This aggregation currently does not support positioning at a specific index.

4. To create a fragment:
 1. Choose [Create New](#).
The default target aggregation is the one you selected in the previous step.
 2. Enter a name for the fragment and choose [Create](#).
A fragment.xml file is created in the folder: ► [Your project name](#) ► [webapp](#) ► [changes](#) ► [fragments](#) ► and opens in the editor.
 3. Define the fragment. Save and close the .xml file. For more information, see [SAPUI5 documentation](#).
An associated addXML.change file is created for every fragment in the folder: ► [Your project name](#) ► [webapp](#) ► [changes](#) ►. This change file contains the reference to the fragment.xml file, selected target aggregation, and index.
5. To choose an existing fragment from the list:
 1. Select a fragment from the list and choose [Add](#).
 2. Double click the `fragment.xml` file from the workspace to open it in the editor.
 3. Edit the code to modify the properties of the fragment. Save and close the .xml file.
6. Navigate to the canvas to see the changes that you made.
After adding fragments, you'll be prompted to reload the SAPUI5 Visual Editor to see your changes.

i Note

Have in mind that you should add definition of the namespace of the controls you are going to use inside of the fragment. Also, you should use stable and unique IDs for the controls you define.

Example:

```
<core:FragmentDefinition xmlns:core='sap.ui.core' xmlns:uxap='sap.uxap'>
  <uxap:ObjectPageSection id="sample.Id"
  title="Title"><uxap:ObjectPageSection>
</core:FragmentDefinition>
```

Also, each control might have its own specific rules for its definition and they should be followed in order for everything to work fine.

→ Remember

You can delete the fragments that you create. For a seamless experience, first delete the change files associated to a fragment and then delete the fragment. If you don't delete them, you might not be able to add further fragments to the adaptation project.

Add Fragments to an Extension Point

Prerequisites

- You are using SAPUI5 version 1.78 or above.
- You are using SAP_UI version 7.55 or above..
- You are using freestyle application that has extension points for basis of your adaptation project.
- You are in the Advanced mode of SAPUI5 Visual Editor.

Procedure

1. In your workspace, from the project context menu, choose *SAPUI5 Visual Editor*.
The application opens in the canvas in preview mode.
2. From the editor header, choose *Edit*.
3. On the left side of the Visual Editor you can see the outline tree. If you expand it enough you will be able to see which element is a possible extension point. Then you can select the element in the tree and you will see its parent marked and highlighted in the visual part of the editor. You can then right click on that highlighted element and choose *Add fragment at extension point*. Alternatively, you can right click on the tree element inside the tree and also select *Add fragment at extension point*.
In the dialog box that opens, the default extension point is selected. A list displays the fragments available for the selected extension point. You can also create fragments. Choose the target extension point and the index from the list where you want to add the fragment. You cannot reuse the same fragment multiple times.
4. To create a fragment:
 1. Choose *Create New*.
The default extension point is the one you selected in the previous step.
 2. Enter a name for the fragment and choose *Create*.
A fragment.xml file is created in the folder: `▶ Your project name ▶ webapp ▶ changes ▶ fragments ▶` and opens in the editor.
 3. Define the fragment. Save and close the .xml file. For more information, see [SAPUI5 documentation](#).
An associated addXML.change file is created for every fragment in the folder: `▶ Your project name ▶ webapp ▶ changes ▶`. This change file contains the reference to the fragment.xml file, selected target aggregation, and index.
5. To choose an existing fragment from the list:
 1. Select a fragment from the list and choose *Add*.
 2. Double click the `fragment.xml` file from the workspace to open it in the editor.
 3. Edit the code to modify the properties of the fragment. Save and close the .xml file.
6. Navigate to the canvas to see the changes that you made.
After adding fragments, you'll be prompted to reload the SAPUI5 Visual Editor to see your changes.

Note

Have in mind that you should add definition of the namespace of the controls you are going to use inside of the fragment. Also, you should use stable and unique IDs for the controls you define.

Example:

```
<core:FragmentDefinition xmlns:core='sap.ui.core' xmlns:uxap='sap.uxap' >
  <uxap:ObjectPageSection id="sample.Id"
  title="Title"><uxap:ObjectPageSection>
</core:FragmentDefinition>
```

Also, each control might have its own specific rules for its definition and they should be followed in order for everything to work fine.

→ Remember

You can delete the fragments that you create. For a seamless experience, first delete the change files associated to a fragment and then delete the fragment. If you don't delete them, you might not be able to add further fragments to the adaptation project.

Related Information

[Controller Extensions \[page 341\]](#)

7.1.11.1.2.3 Controller Extensions

Controller extensions allow you to add functionality to a fragment that you add to the SAP Fiori elements-based application. You can also override the functionality of the base methods inherited from the source application and lifecycle methods such as `onInit`, `onBeforeRendering`, `onAfterRendering`, and `onExit`.

Context

Controller extensions allow you to enhance the functionality of a controller. You can create a controller extension specific to the view, for example, one controller extension for the list report view and one controller extension for the object page view. Additionally a controller extension can be delivered with an Adaptation Project and is dynamically added to an existing controller. Controller extensions that an Adaptation Project delivers are added to the reserved ".extension" namespace of the controller to avoid name clashes with existing functionality.

Controller extensions let the developer to add new methods and override methods. These override methods are optional callback methods that override the existing methods using a special override member. For more information on defining an extension, see [SAPUI5 Documentation](#).

→ Remember

You can add a controller extension only if:

- SAPUI5 Visual Editor supports the project.
- You use an SAP Fiori elements-based application (LR, OVP, and ALP).

Procedure

1. In your workspace, from the project context menu, choose [SAPUI5 Visual Editor](#).
The application opens in the canvas in preview mode.
2. From the Editor Header, select [Edit](#).
3. On the canvas, select a view or element in the view you want to extend and choose [Extend with Controller](#) from the project context menu.
4. In the dialog box that appears, provide a name for the controller extension.

The controller extension file (.js) is created in the folder, **► Your project name ► webapp ► changes ► coding ►**. The .js file opens in the editor.

When you create a controller extension file, an associated **controllerExtension.change** file having reference to the .js file is created in the folder, **► Your project name ► webapp ► changes ►**.

For list report, object page, and over view page applications, in addition to the lifecycle methods, this .js file contains new methods that come from the templates provided by Fiori elements. These methods can be used to override the base methods in the applications. For analytical list page applications, the .js file contains only the lifecycle methods. Reuse the methods to write extensions.

i Note

Have in mind that if you need to add your own separate custom business logic files, they have to be located in the folder `changes/coding`, the same folder that is used for the controller extension files. You should not create additional folders for such additional files on upper levels, because it will invalidate the project structure and you will not be able to deploy.

5. Define a method in the controller extension file for the new fragment or override any lifecycle methods in the override section. Save the file.

The lifecycle methods are defined within the override section. Make sure that you define the new methods outside the override section.

The controller extension file contains a built-in debugger which is supported from SAPUI5 version 1.60 onwards.

6. To assign event handlers from .xml fragment to the controller extension, manually associate the methods with the respective fragments. In the .xml file for the fragment, prefix the method with `.extension.<controller extension namespace>.` and save the file.
7. Navigate to the canvas. You are prompted to reload the project for the controller changes to be applied.

A loading indicator appears with the message, *Loading SAPUI5 Visual Editor with Controller Extension changes*.

7.1.11.1.2.4 Internationalization

Adaptation project supports internationalization (i18n), which enables you to provide easily translatable text for applications. The topic describes the ways in which you provide text for internationalization. You can define i18n key-value pairs and change the value for existing i18n keys.

Context

Internationalization refers to the process of designing and developing a software application and its content so that it can be adapted to multiple languages without making much engineering changes. When you create an Adaptation project, SAPUI5 Visual Editor creates an *i18n.properties* file for every page of the application. Define a key-value pair in the *i18n.properties* file that you can refer to while providing text for the UI controls.

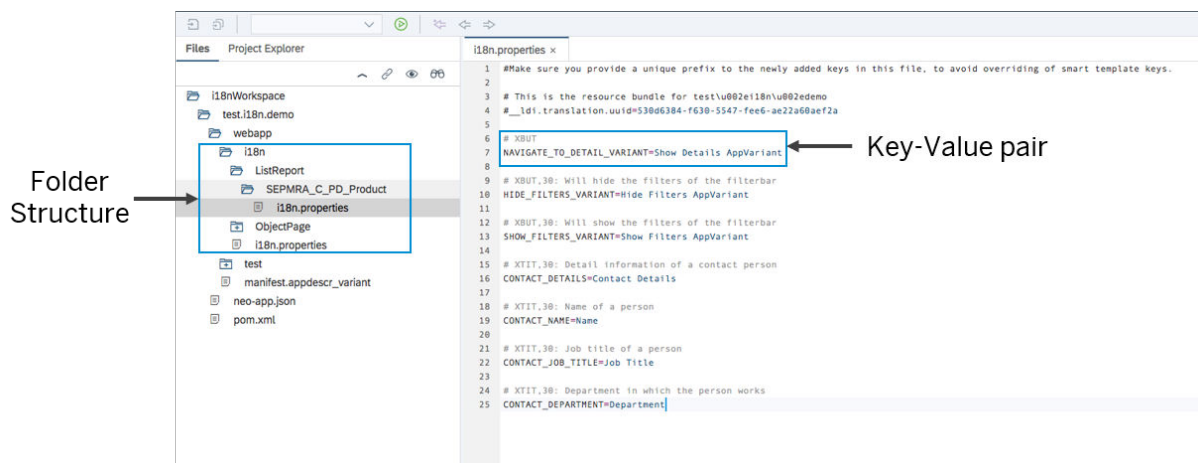
Folder Structure

In your workspace, under the Adaptation project, the folder structure looks like:

- Project name
 - webapp
 - i18n
 - page 1
 - collection
 - i18n.properties
 - page 2
 - collection
 - i18n.properties

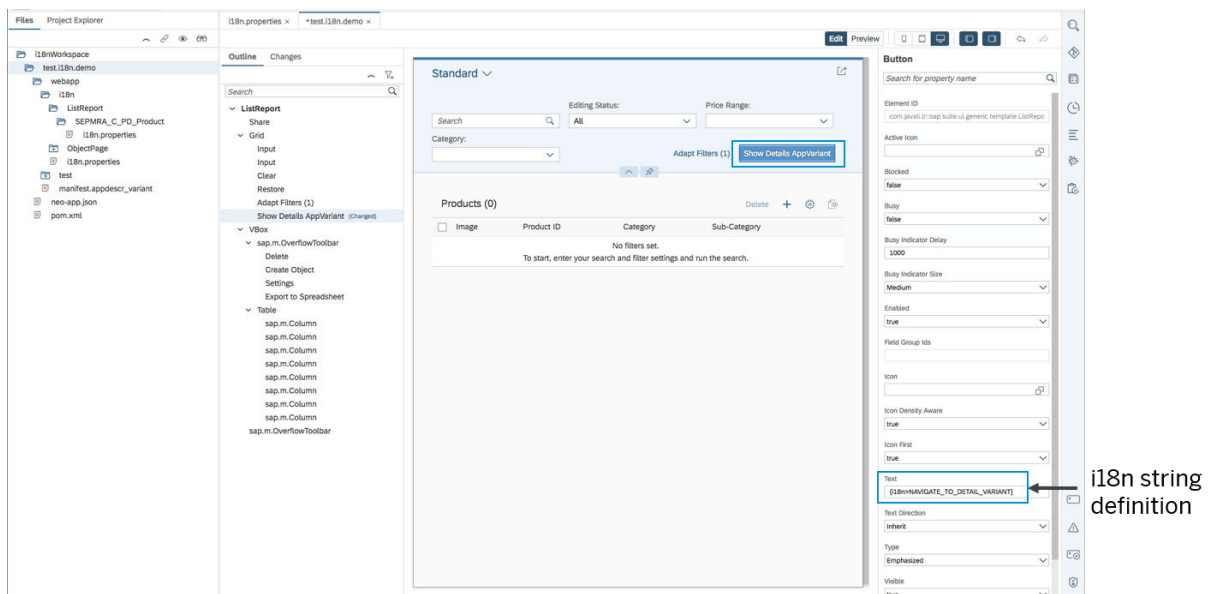
Define Key-Value Pair

Define your key-value pairs in the *i18n.properties* file for each page (list report or object page).



Refer to the keys while you edit the project using SAPUI5 Visual Editor.

In the *Properties* pane, refer to the i18n key for a SAPUI5 control instead of typing-in the text.



Run the project as a web application and check whether the i18n value associated to the key appears on the UI control.

Similarly, you can change the text of existing UI controls by overriding the associated i18n key-value pairs in the source application.

7.1.11.1.3 Adding App Descriptor Changes

Adaptation project gives you the opportunity to add also the following types of changes to your application variant.

Contents

- [Replacing OData Service \[page 345\]](#)
- [Adding Annotation Files \[page 346\]](#)
- [Adding OData Service and New UI5 Model \[page 348\]](#)

7.1.11.1.3.1 Replacing OData Service

This guide describes how to replace OData service in an adaptation project without tooling support.

Context

It is required that new OData service must be compatible to the OData service of the original app. For V2 service using annotations also the annotation data source needs to be changed and the new annotation must also be compatible to the annotation of the original app.

Procedure

1. Create your project and disable Safe Mode from within the Visual Editor.
2. Determine the name of the data source of the base app for which the URI should be replaced.

Check the manifest.json of the base app and find the name under sap.app/dataSources.

3. Open the file webapp/manifest.appdescr_variant and add the following to the content array:

webapp/manifest.appdescr_variant to change URI

```
{
    "fileName": "manifest",
    "layer": "<App Variant Layer>",
    "fileType": "appdescr_variant",
    "reference": "<Base App Id>",
    "id": "<App Variant Id>",
    "namespace": "<...>",
    "content": [
        {
            "changeType": "appdescr_app_changeDataSource",
            "content": {
                "dataSourceId": "<Target Data Source Name>",
                "entityPropertyChange": [{
                    "propertyPath": "uri",
                    "operation": "UPDATE",
                    "propertyValue": "<New Uri>"
                }]
            }
        },
        ...
    ]
}
```

For V2 service using annotations also the annotation data source URI needs to be changed.

In addition, you can also exchange settings/maxAge.webapp/manifest.appdescr_variant to change URI and settings/maxAge

```
{
    "fileName": "manifest",
    "layer": "<App Variant Layer>",
    "fileType": "appdescr_variant",
    "reference": "<Base App Id>",
```

```

    "id": "<App Variant Id>",
    "namespace": "<...>",
    "content": [
      {
        "changeType": "appdescr_app_changeDataSource",
        "content": {
          "dataSourceId": "<Target Data Source Name>",
          "entityPropertyChange": [{
            "propertyPath" : "uri",
            "operation" : "UPDATE",
            "propertyValue" : "<New Uri>"
          },
          {
            "propertyPath" : "settings/maxAge",
            "operation" : "UPSERT",
            "propertyValue" : <New maxAge value as number>
          }
        ]
      },
      {
        "propertyPath" : "settings/maxAge",
        "operation" : "UPSERT",
        "propertyValue" : <New maxAge value as number>
      }
    ],
    ...
  ]
}

```

4. Run the preview.

7.1.11.1.3.2 Adding Annotation Files

This guide describes how to add annotation files to an adaptation project without tooling support.

Prerequisites

You have the ABAP Platform 2020 SPO0 installed.

Procedure

1. Create your project.
2. Determine the name of the data source of the base app.
Check the manifest.json of the base app and find the name under sap.app/dataSources.
3. Create a new folder named annotations below webapp/changes. This naming is required.
4. Create a new xml file. For example myAnno.xml and put in your annotation code.

webapp/changes/annotations/myAnno.xml

```

<edmx:Edmx xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx"
Version="4.0">
  <edmx:Reference Uri="/sap/opu/odata/iwfnd/
CATALOGSERVICE;v=0002/$metadata">
    <edmx:Include Alias="Metadata"
Namespace="CATALOGSERVICE"/>
  </edmx:Reference>

```

```

        <edm:Reference Uri="https://oasis-tcs.github.io/
odata-vocabularies/vocabularies/Org.OData.Core.V1.
xml">
        <edm:Include Alias="Core"
Namespace="Org.OData.Core.V1"/>
    </edm:Reference>
    <edm:DataServices>
    <Schema xmlns="http://docs.oasis-open.org/odata/ns/
edm" Namespace="sap.app.var.id.myAnno.
CATALOGSERVICE">
    <Annotations Target="Metadata.Catalog">
    <Annotation Term="Core.Description"
String="{i18n&gt;BASE_ANNOTATION_DESCRIPTION}"/>
    </Annotations>
    </Schema>
    </edm:DataServices>
    </edm:Edmx>

```

Adjust the namespace as needed. For text bindings you might make use of texts which have been provided by the adaptation project.

5. Open the file `webapp/manifest.appdescr_variant` and add the following to the content array:

`webapp/manifest.appdescr_variant`

```

{
    "fileName": "manifest",
    "layer": "VENDOR",
    "fileType": "appdescr_variant",
    "reference": "<Base App Id>",
    "id": "sap.app.var.id",
    "namespace": "<...>",
    "content": [
        {
            "changeType": "appdescr_app_addAnnotationsToOData",
            "content": {
                "dataSourceId": "<Target Data Source Name>",
                "annotations": ["sap.myAnno"],
                "annotationsInsertPosition": "END",
                "dataSource": {
                    "sap.myAnno": {
                        "uri": "changes/annotations/myAnno.xml",
                        "type": "ODataAnnotation"
                    }
                }
            }
        },
        ...
    ]
}

```

6. Run the preview.

7.1.11.1.3.3 Adding OData Service and New UI5 Model

This guide describes how to add OData service with the option to also add new UI5 model.

Context

To add V2 service using annotations also the new annotation data source needs to be added.

Procedure

1. Create your project and disable the Safe Mode of the Visual Editor.
2. Open the file `webapp/manifest.appdescr_variant` and add one of the following to the content array, depending on your needs:

Including new UI5 model	Including new annotations	Code
Yes	Yes	<p>webapp/ manifest.appdescr_variant to add new OData service and ODataAnnotation with new UI5 Model</p> <pre> { "fileName": "manifest", "layer": "<App Variant Layer>", "fileType": "appdescr_variant", "reference": "<Base App Id>", "id": "<App Variant Id>", "namespace": "<...>", "content": [{ "changeType": "appdescr_ui5_addNewMod el", "content": { "dataSource": { "<New OData Data Source Name>": { "uri": "<New OData Data Source URI>", "type": "OData", "settings": { "odataVersion": "2.0", "annotations": [<New ODataAnnotation Data Source Name>"] } }, "<New ODataAnnotation Data Source Name>": { "uri": "<New ODataAnnotation Data Source URI>", "type": "ODataAnnotation", "settings": {} } } } }] } </pre>

```
"model": {  
  "<New UI5 Model  
  Name>": {  
    "dataSource": "<New  
    OData Data Source  
    Name>",  
    "settings": {}  
  }  
},  
  ...  
]  
}
```

Including new UI5 model	Including new annotations	Code
Yes	No	webapp/ manifest.appdescr_variant to add new OData service with new UI5 Model

```

{
  "fileName":
  "manifest",
  "layer": "<App
Variant Layer>",
  "fileType":
  "appdescr_variant",
  "reference":
  "<Base App Id>",
  "id": "<App
Variant Id>",
  "namespace":
  "<...>",
  "content": [
    {
      "changeType":
      "appdescr_ui5_addNewMod
el",
      "content":
      {
        "dataSource": {
          "<New OData Data
Source Name>": {
            "uri": "<New OData
Data Source URI>",
            "type": "OData",
            "settings": {
              "odataVersion":
              "2.0"
            }
          }
        },
        "model": {
          "<New UI5 Model
Name>": {
            "dataSource": "<New
OData Data Source
Name>",
            "settings": {}
          }
        }
      }
    },
    ...
  ]
}

```

Including new UI5 model	Including new annotations	Code
		}

It is required that <New OData Data Source Name>, <New ODataAnnotation Data Source Name> and <New UI5 Model Name> start with `customer`. e.g. `customer.newdatasource`, `customer.newanno` and `customer.newmodel`.

3. Run the preview.

7.1.11.1.4 Preview the App Variant

Preview any changes that you've made to the app variant before you deploy to ABAP repository.

Context

You can preview the app variant in either of the following ways:

- Directly launch the app variant in your browser – to see the runtime behavior of the app variant.
- Launch the app variant via Fiori Launchpad – to see the app variant as a tile where you can see the application title. You can also navigate back and forth to the FLP home and app variant multiple times.

SAP Web IDE stores the way you've selected to preview the app variant, and you don't select this option again.

Procedure

1. In your workspace, from the adaptation project context menu, select **Run > Run as > Web Application**.
2. In the pop-up that appears, select one of the following options, then click **OK**. Both files show the corresponding file path adjacent.
 - **Adaptation_index.html** – to launch the app variant.
 - **flpSandbox.html** – to see the app variant as a tile.
3. If you want to change the preview type, follow these steps:
 - a. In your workspace, for the adaption project, select **Run > Run Configurations** from the context menu.

You can either delete the previous configuration or add a configuration.

- b. In the **Web Application** list, you see the previous configuration that you chose. Click the configuration to edit, duplicate, or delete it.

If you delete the configuration, SAP Web IDE prompts you to choose a launch option the next time you preview the app variant.

- c. In the **Web Application** list, click **+** icon to add a configuration. Choose **Web Application** and enter a name and file name for the second configuration that you want to add. Click **OK**.

When you add a second configuration, you can choose between the two options the next time you preview the app variant.

7.1.11.1.5 Deploy the App Variant

Deploy the app variant to the layered repository on the ABAP platform.

Prerequisites

This prerequisite is only valid for on premise customer developers who deploy their adaptation projects to SAP NW AS ABAP. In order to deploy, your user should be assigned to the role `SAP_UI_FLEX_DEVELOPER`, if SAP_UI version is 7.55 and above, or to the role `S_DEVELOP`, if SAP_UI version is below 7.55. Also, if a cloud connector is in use, POST and PUT requests should be allowed for this path: `/sap/bc/lrep/dta_folder/`.

Context

After you have created the app variant, you deploy it to your ABAP system. Since the app variant is a new semantic app with a new ID, you also need to configure a new target mapping for the app variant to create a new tile on the SAP Fiori Launchpad.

Procedure

1. In your workspace, from the adaptation project context menu, select **Deploy** **Deploy to SAP NW Application Server ABAP**.
2. Confirm the *System* and *Namespace* selection by choosing *Next*.

If you have already deployed the adaptation project in the past, steps 3 and 4 are skipped automatically. The deployment uses the same package/transport that you selected for the earlier deployment.

→ Remember

App variant works as expected only if you deploy it to the same system where the source application resides.

3. Select a package.
 - a. For an existing package, choose *Browse*, search for an existing package and choose *OK*. Then, choose *Next* and continue with step 4.
 - b. For a local object, choose *\$TMP*. Then, choose *Next* and continue with step 5.
4. If you have selected an existing package from the list, select or create a transport and choose *Next*.

5. Choose *Finish*.

The UI changes you made in the adaptation project are applied to the layered repository in the selected SAP NetWeaver Application Server ABAP for your application. To add the new app variant to the SAP Fiori launchpad, you need to configure target mappings in the SAP Fiori launchpad designer. For more information, see [Configuring Target Mappings](#).

7.1.11.1.6 Delete the App Variant

If you no longer need an app variant that you created via an adaptation project, you can delete it from the back-end system as described here.

Prerequisites

You have deployed an app variant, which you created via an adaptation project, to your ABAP system.

Context

⚠ Caution

If a key user has created further app variants based on the app variant you just deleted, these key user app variants will be deleted, as well.

To delete the deployed app variant from the layered repository on the ABAP platform, proceed as follows:

Procedure

1. In your workspace, open the `manifest.appdescr_variant` file in your adaptation project and copy the value of the `id` element.
2. In the respective back-end system, start transaction SE38 and execute report `/UI5/DEL_ADAPTATION_PROJECT`.
3. Paste the copied ID of the app variant that you want to delete, or use the F4 help to find it.
4. Choose *Execute* (F8).
5. If prompted, select or create a transport request.
6. In the overview of the files that will be deleted, choose *Execute* (Shift+F1) again.
7. The app variant is deleted from all systems as soon as you release the transport request.

7.1.11.2 Extending SAP Fiori Elements-Based Applications

Developers can extend the capabilities of an existing SAP Fiori elements-based application by adding new functionality and extending the existing functionality using the SAPUI5 Visual Editor.

Currently, SAPUI5 Visual Editor supports List Report and Object Page, Analytical List Page, and Overview Page applications.

i Note

Currently extending SAP Fiori elements-based applications is possible only on S/4 HANA on-premise ABAP systems with the minimum required version of software component being SAP_UI 7.53 SP00. Have in mind that, if you have SAP_UI 7.53 SP00 installed, you need to apply the SAP note [2615176](#), and implement the manual correction instructions to enable the system. If you are on SP01 or higher, no actions are needed.

After you've created an adaptation project, you can extend the application capabilities by doing the following:

- Create fragments.
- Add existing fragments.
- Enhance the existing fragments by modifying the code in the fragment.xml file using the code editor.
- Create a **controller extension** per view and attach to an event.

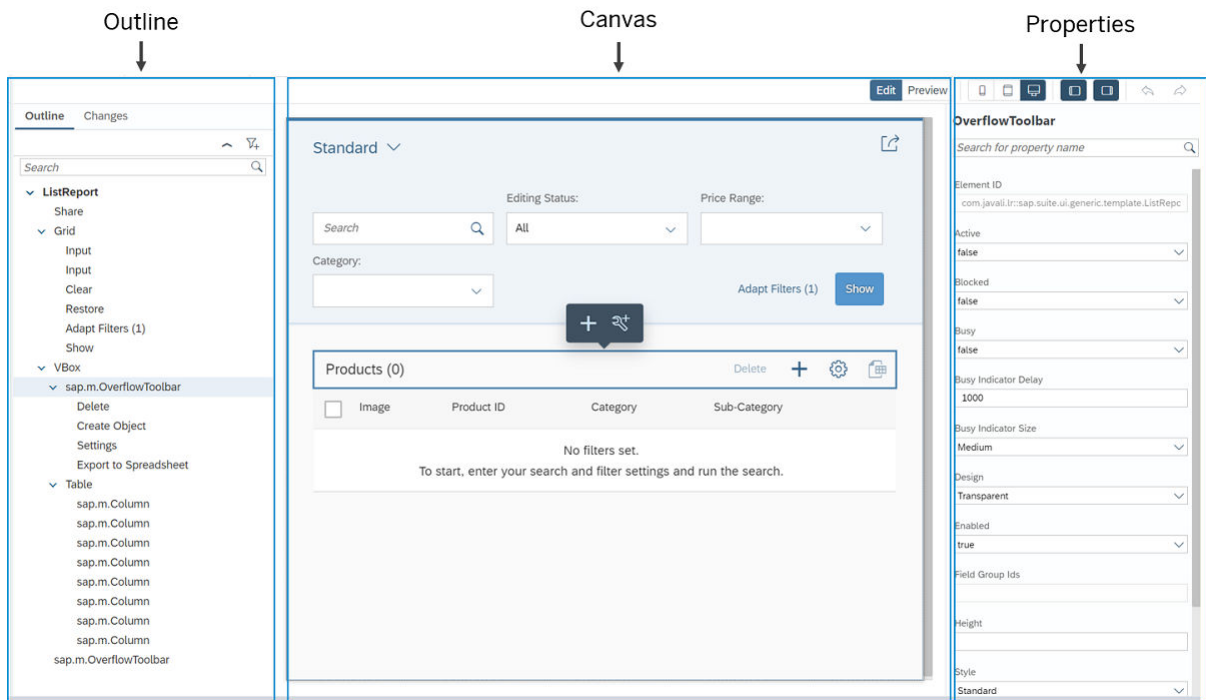
7.1.11.2.1 SAPUI5 Visual Editor Usage

SAPUI5 Visual Editor allows developers to change, adapt, and extend the user interface of SAPUI5 applications.

SAPUI5 Visual Editor is a design-time editor in SAP Web IDE providing capabilities such as creating and modifying SAP Fiori elements-based applications, adapting control properties, and extending existing applications without modification.

The SAPUI5 Visual Editor provides an intuitive user interface to modify SAP Fiori element applications. For example, you can add, remove, or move fields and groups. You can also view all properties of the controls in the application and change the configurable properties.

The editor has an Outline pane, a Canvas (application preview), and a Properties pane.



The buttons on the SAPUI5 Visual Editor toolbar allow you to:

- Navigate the application using the *Preview* mode.
- Change the application using the *Edit* mode. In this mode, if you click a UI element in the Canvas, the element is selected and highlighted in the Outline pane and vice versa. You can deselect the UI element by clicking it again in the Canvas. The Properties pane displays the properties of the UI element.

i Note

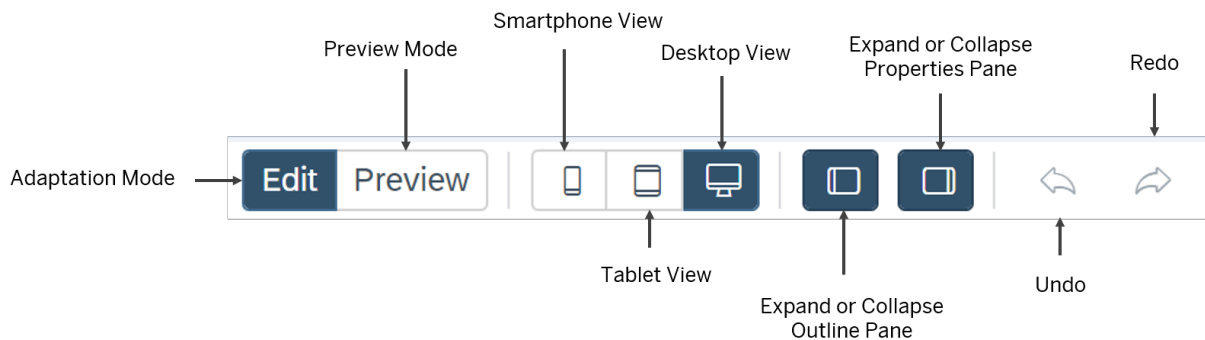
If you switch between modes, your changes are saved to the workspace.

- Change the device format of the canvas to smartphone, tablet, or desktop view.

i Note

If you switch between device formats, your changes are saved to the workspace.

- Expand and collapse the panes to the right and left of the canvas.



Change Applications with the SAPUI5 Visual Editor

You can change an application that is based on the supported SAP Fiori elements.

Prerequisites

- You must have an application that is based on supported SAP Fiori elements. You can create such an application in the *New Project from Template* wizard. In the *Annotation Selection* step, make sure that you have an annotation file. For more information, see [Create Projects from a Template \[page 92\]](#).
- From the project's context menu, select **Project Settings** > **Project Types** and make sure the project type *SAPUI5 Visual Editor* is selected.

Procedure

1. From the project's context menu, select *SAPUI5 Visual Editor*. The application opens up in the canvas in preview mode.
2. Navigate to the page containing the UI element you want to change.
3. From the menu bar, select *Edit* to make changes.
4. Select the UI element that you want to adapt. The control is selected in the application runtime and in the *Outline* pane.

The *Outline* pane displays a filtered list of controls. To see the complete list of controls available in the view,

click  .

The *Properties* pane displays the properties of the selected control.

i Note

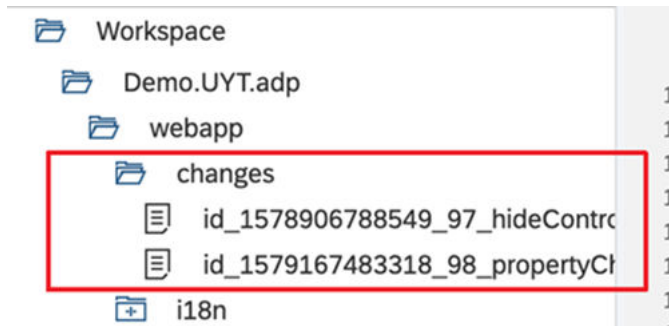
SAPUI5 Visual Editor can be used to adapt or change properties of the supported controls with stable IDs and for making property changes to non-smart controls.

5. To make a change, choose an option from the UI element's context menu, or adapt its property in the *Properties* pane.

i Note

- Not all properties can be changed. Only properties that have been enabled for editing may be changed.
- You can undo the change using the *Undo/Redo* buttons as long as you do not save, change the mode, or change the device.

6. In the *Outline* pane, open the *Changes* tab to see your changes. Alternatively, expand your project folder to see your changes within the `changes` folder that was created under the `webapp` folder. You can remove the changes by deleting them from the *Changes* tab or by directly deleting the files from the workspace.



i Note

Delete the changes one-by-one, starting with the latest change. For example, if you combine two fields and rename them, delete the renaming change first and then delete the change that combined the fields. If you do it the other way, the project might have inconsistencies.

7. Run the application to preview the changes.
8. To incorporate these changes in the application in your SAPUI5 ABAP repository, deploy the application as described in [Deploy Applications to the SAPUI5 ABAP Repository \[page 414\]](#).

7.1.11.2.1.1 UI Editing Options

You can make various changes to your application using the SAPUI5 Visual Editor.

Change properties

1. Select the UI element you want to change.
2. Change the element's properties as needed.

i Note

Not all properties can be changed.

Add new fields

1. Hover over or select a group or a field and choose *Add Field* from the context menu.
2. Select the fields from the list of available fields that you want to add to the UI.
You can also search for field labels and tooltips, or sort the fields in alphabetical order.
3. To apply your adaptations, choose *OK*.

Add a new group

1. Hover over or select a group or select the form it's contained in and choose *Add Group* from the context menu. The default group title is *New Group*, and you can rename it to whatever you want.
2. To apply your adaptations, press **ENTER** or select another element.

Add sections to an object page

1. Hover over or select a section and choose *Add Section* from the context menu.

i Note

If all available sections are already on the object page, you cannot use this function and it's grayed out in the context menu.

2. Select the sections from the list of available sections that you want to add to the UI.
You can also search for sections or sort them in alphabetical order.
3. To apply your adaptations, choose *OK*.

Rename fields and groups

1. Double-click a field or group. You can also hover over or select it and choose *Rename Field* or *Rename Group* from the context menu.
2. Rename the field label or group title.
3. To apply your adaptations, press `ENTER`; to quit, press `ESC`.

-
1. Drag a field, group, or section.
 2. Drop the field, group, or section on its new location.
A space appears where you can drop it. You can drop a field above or below any of the highlighted fields or in any group marked with a dashed box; you can drop a group or section on any of the highlighted groups or sections.

Cut and paste fields and groups

1. Hover over or select a field or group and choose *Cut* from the context menu. The cut field or group gets highlighted. Also, the groups where you can paste the cut field or the forms where you can paste the cut group get highlighted using dashed boxes. Move fields, groups, and object page sections.
2. To paste a cut field, hover over or select a highlighted group or a field in a highlighted group and choose *Paste*. To paste a cut group, hover over or select a group in the highlighted forms and choose *Paste* from the context menu.

i Note

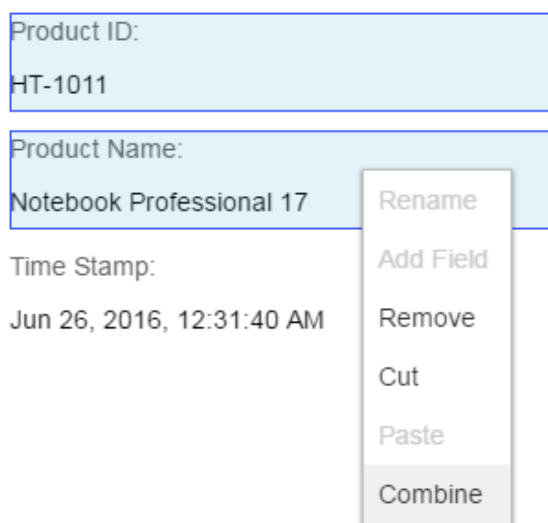
To remove the highlighting and exit pasting, press `ESC`.

Combine fields

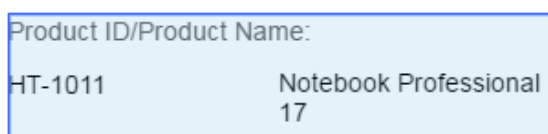
You can combine up to three fields so that they're displayed in a single line.

1. Select a field.
2. Press and hold **CTRL**. Move fields, groups, and object page while selecting the other fields you want to combine with this field.
3. Choose *Combine* from the context menu of one of the selected fields where you want the combined fields to be displayed.

Product Info



Product Info



Split combined fields

1. Hover over or select the combined fields.
2. Choose *Split* from the context menu.

Remove fields, groups, or object page sections

1. Hover over or select the field, group, or section that you want to remove from the UI.
 2. Choose *Remove Field*, *Remove Group*, or *Remove Section* from the context menu or press **DEL**.
The fields and sections are only removed from the UI, not permanently deleted. They're still available in the list of available fields or sections, and you can add them again at any point. You cannot remove mandatory fields (also those contained in groups) by accident as the system will ask you to confirm.
-

7.1.11.2 Customize Overview Page Cards

You can add, edit, and clone overview page cards in the adaptation project.

Prerequisites

- To add a new card, use SAPUI5 version 1.64 or above.
- To edit and clone a card, use SAPUI5 version 1.67 or above.

Context

Procedure

Right-click on a card and select one of the following:

- Add new card: You can add a new card with existing or new data sources.
- Edit card
- Clone card

i Note

If you want to delete a newly added or a cloned card, save the card first, close SAP UI5 Visual Editor, and then return to delete the card. You can also choose to hide the new or cloned card.

Custom Parameters

You can set custom parameters while editing cards. To set custom parameters in a newly added card, save it first and then edit the card.

For more information, see [Extending Delivered Apps Using Adaptation Extensions](#).

7.1.11.3 Known Issues

Following are the known issues when working with SAPUI5 Visual Editor.

- **Issue:** i18n changes not visible when you launch AdaptationIndex.html though the changes are visible in the editor.
Workaround: None. The i18n changes work fine in the editor and can be checked there.
- **Issue:** When you create an Overview Page application as an HTML5 module inside a multitarget application project, the SAPUI5 Resources path "/webapp/resources" cannot be found. Hence, you cannot edit the application using SAPUI5 Visual Editor.
Workaround: In the `manifest.appdescr` file, update the SAPUI5 resources path.
- **Issue:** Add Fragment and Extend Controller dialog boxes do not open when you select a node from the outline pane and then use the context menu of that control in the canvas.

Workaround: To add the fragment or extend the controller, select the node from the canvas and then use the context menu of the control.

- **Issue:** When you create a fragment that an aggregation doesn't support, the aggregation changes to the default aggregation and the fragment show up in the canvas.
Workaround: None.
- **Issue:** When you select an index to position a fragment in the SmartFilterBar/SmartTable/SmartChart, the fragment is not added to the dedicated index.
Workaround: None.
- **Issue:** Delete a fragment from the workspace; then create a fragment with the same name and content as in the deleted fragment, the control does not show up in the canvas or workspace.
Workaround: Delete the change files associated with the fragment.
- **Issue:** After adding a column to a smart table, you might not be able to click the settings icon or add further columns.
Workaround: None.
- **Issue:** Though the SAPUI5 version is 1.62 or greater, Overview Page applications can fail to launch in the SAPUI5 Visual Editor.
Workaround: Add the following section in the neo-app.json file.

```
{
  "path": "/webapp/resources",
  "target": {
    "type": "application",
    "name": "sapui5preview",
    "entryPath": "/resources",
    "version": "snapshot-untested"
  },
  {
    "path": "/webapp/test-resources",
    "target": {
      "type": "application",
      "name": "sapui5preview",
      "entryPath": "/test-resources",
      "version": "snapshot-untested"
    },
  },
}
```


- **Issue:** You cannot adapt any of the dialog boxes in the application. Also controls associated to the dialog boxes are not listed in the outline pane.
Workaround: None.
- **Issue:** While working with Overview Page applications that contains a header and no cards – you cannot select the controls in the SAPUI5 Visual Editor canvas. You can make property changes to the controls by selecting them from the outline pane.
Workaround: None.

7.1.12 Using Source Control (Git)

SAP Web IDE enables you to connect and interact with Git source control system, letting you connect and interact with remote Git repositories.

Git Tools

SAP Web IDE provides a graphical user interface for executing Git commands and managing your source control and versioning. The following are the main tools for working with Git:

- **Git Menu:** Access the menu from . The menu includes the ability to clone a repository, as well as other Git commands for working with cloned repository.
- **Git Pane:** The Git pane provides a graphic user interface for executing Git commands on a specific Git repository, as well as a status table that lists all the uncommitted changes you've made to your project.

Git

Repository myTest

Branch testBranch + -

Pull Fetch Rebase Merge Show Stash Reset Fetch from Gerrit

Commit

Stage All Discard All Changes

Status	Name	Stage	Discard
M	webapp/Component.js	<input type="checkbox"/>	
M	webapp/index.html	<input type="checkbox"/>	
Status table			

Commit Description

Insert commit description

A Change ID for Gerrit will not be added Amend Changes

Commit and Push Commit Push Stash


To open the *Git* pane:

1. From the workspace, select a Git repository
2. From the right sidebar, choose (Git pane).

- **Git History Pane:** Lets you view the commits for different branches, as well as perform some commands on those commits.







To open the *Git History* pane:

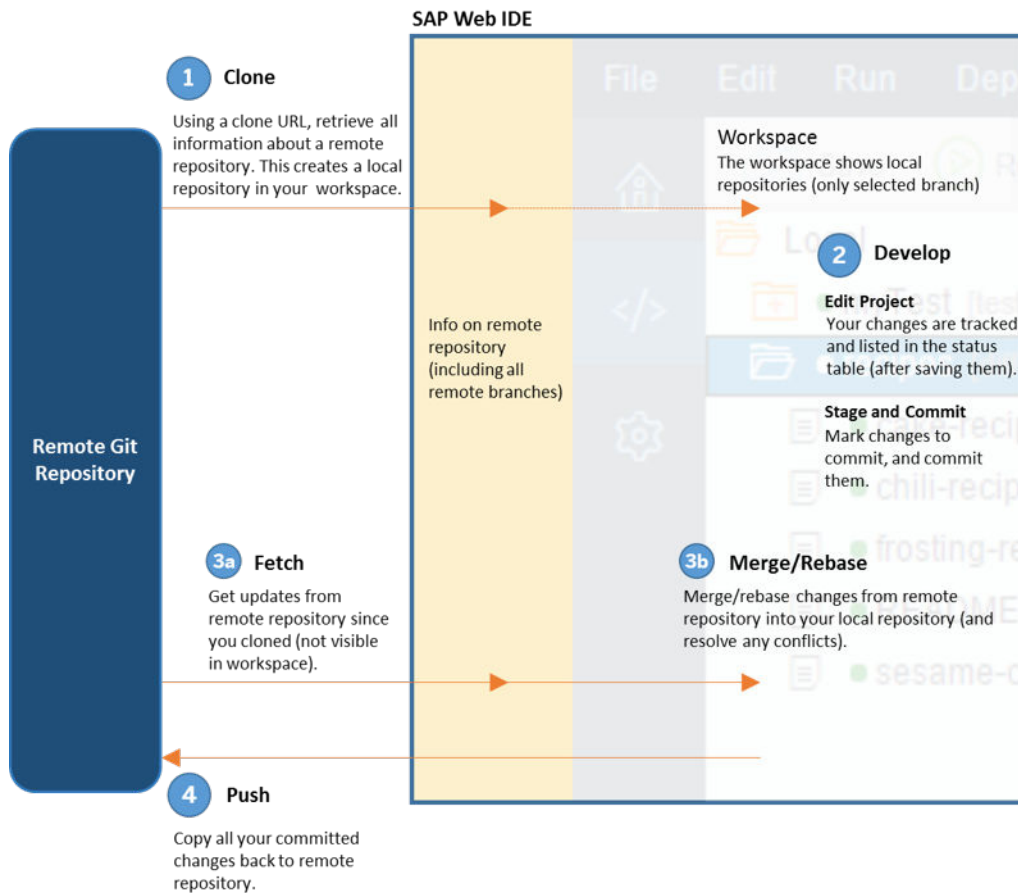
1. From the workspace, select a Git repository

2. From the right sidebar, choose  (Git History pane).
- **Git Blame:** When working in a code editor, you can select *Show Git Blame* from the context menu to show who was the last developer to change each line, the commit that included the change, and the date of the change.
For more information, see [Git Blame \[page 388\]](#).

Workflow

Using Git with SAP Web IDE is easy. The basic workflow is as follows:

1. **Clone:** Clone a repository from a remote Git source control system. All the information about the repository is copied, and a local master branch is created and is visible in your workspace. If the remote repository has several branches, you can create additional local branches based on those remote branches.
You can also create a new repository by creating a new project in your SAP Web IDE, and choosing  *Git*  *Initialize Local Repository* . You can then connect the local repository to a remote repository by choosing  *Git*  *Set Remote* .
2. **Develop:** Once you have the code, you can develop – add files, delete files, modify files. Your changes are visible in the status table of the *Git* pane. When you are ready, you can stage your changes and commit them.
3. **Fetch and Merge/Rebase:** (Optional) Before sending back your changes to the remote repository, you can fetch all the changes made by others. Then you can merge or rebase the changes into your changes to make sure there are no conflicts. If there are conflicts, you can adjust your code.
4. **Push:** Add your changes to the remote repository.



Related Information

[Git Troubleshooting \[page 555\]](#)

7.1.12.1 Set Up Git

To use source control in your SAP Web IDE project, your user name and email address must be set for your Git account.


Context

By carrying out the following steps, you can either set your Git user settings or check whether they are already set correctly.

i Note

Your user name and email address will be stored on the remote Git server. This is mandatory for Git operations and cannot be undone.

Procedure

1. Open SAP Web IDE in one of the supported browsers using the subscription URL.
2. Choose  (Preferences) and select *Git Settings*.
3. Enter your email address and name.

i Note

The email address field is case-sensitive.

i Note

If you have not set your Git user name and email address, SAP Web IDE extracts this information from the identity provider defined in your account and pre-populates these fields in the *Git Settings* page.

4. Choose *Save*.

7.1.12.2 Connect to your Corporate Git System

You can manage the connectivity to your on-premise Git repository.

Context

i Note

The corporate Git connectivity supports only secure HTTPS connections. HTTP, SSH and other protocols are not supported.

Procedure

1. Install and configure a Cloud Connector. For more information, see [Cloud Connector](#).
2. Configure the cloud connector to open a channel to your Git system. Follow the instructions as described in [Configuring Access Control \(HTTP\)](#). Use the following settings:

Field	Value
<i>Back-end Type</i>	<i>Non-SAP System</i>
<i>Protocol</i>	<i>HTTPS</i>
<i>Internal Host / Port</i>	Enter the internal host and port for your Git system.
<i>Host / Port</i>	Enter a virtual host and port for your Git system. You can use the same host and port as for the virtual host and port.
<i>Principal Type</i>	<i>None</i>

For the system you just added, specify the resources to enable, using the following settings:

Field	Value
<i>Enabled</i>	Checked
<i>URL Path</i>	<i>/</i>
<i>Access Policy</i>	<i>Path and all sub-paths</i>

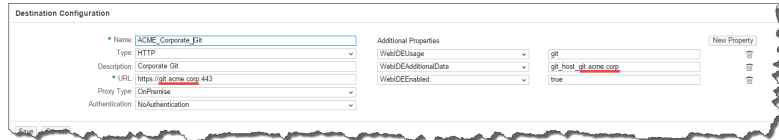
3. Upload your organization's Git server certificate to the cloud connector (if your Git server is using certificate-based authentication).
4. If you defined a custom identity provider, make sure that you have configured the assertion-based attributes mapping for this identity provider. For more information, see [Configure Trust to the SAML Identity Provider](#).
5. Define your corporate Git destination. For more information, see [Connect to ABAP Systems \[page 57\]](#).
 - a. In the SAP BTP cockpit, select **Connectivity > Destinations**.
 - b. Select *New Destination*.
 - c. In the *Destination Configuration* section, set the *Proxy Type* to *OnPremise*.
 - d. In the *Additional Properties* section, configure the following:

Property	Value
WebIDEEEnabled	true
WebIDEUsage	git

Property	Value
WebIDEAdditionalData	git_host_<Git host name>

The Git host name must be entered as it appears in the URL. For example, if the URL is `https://git.acme.corp:443`, the host name would be `git.acme.corp`.

So you would enter in the field: `git_host_git.acme.corp`



i Note

When you define the cloud connector, there are two types of hosts: Internal and Virtual. Make sure to use the virtual host.

Next Steps

Test your new corporate Git system by cloning a repository from your corporate Git system (see [Clone Repositories \[page 371\]](#)) or by initializing a local repository, setting the remote repository, fetching and pushing to the remote repository (see [Initialize a Local Git Repository \[page 372\]](#)).

7.1.12.3 Connect to a Public Git Server that Requires a Certificate

You can manage the connectivity to a public Git server that requires a client certificate.

Context

i Note

The Git connectivity supports only secure HTTPS connections. HTTP, SSH and other protocols are not supported.

Procedure

1. In the SAP BTP cockpit, select **Connectivity > Destinations**.
2. Select *New Destination*.
3. In the *Destination Configuration* section, do the following:
 - a. Set *Proxy Type* to *Internet*.
 - b. Set *Authentication* to *ClientCertificateAuthentication*.
 - c. Upload the client certificate in the *Key Store Location* field.
 - d. Enter your *Key Store Password*.
4. In the *Additional Properties* section, configure the following:

Property	Value
WebIDEEnabled	true
WebIDEUsage	git

WebIDEAdditionalData git_host_<Git host name>

The Git host name must be entered as it appears in the URL. For example, if the URL is `https://git.acme.corp:443`, the host name would be `git.acme.corp`.

So you would enter in the field: `git_host_git.acme.corp`

Destination Configuration

*Name: ACME_GIT
Type: HTTP
Description: Public git server that requires certificate
*URL: https://git.acme.corp
Proxy Type: Internet
Authentication: ClientCertificateAuthentication
*Key Store Location: p12
Key Store Password:

Additional Properties

WebIDEAdditionalID: git_host_git.acme.corp
WebIDEEnabled: true
WebIDEUsage: git
 Use default JDK truststore

5. Select the *Use default JDK truststore* checkbox.




Next Steps

Test your new Git system by cloning a repository (see [Clone Repositories \[page 371\]](#)) or by initializing a local repository, setting the remote repository, fetching and pushing to the remote repository (see [Initialize a Local Git Repository \[page 372\]](#)).

7.1.12.4 Clone Repositories

You can clone an existing Git repository into your workspace.

Procedure

1. From the *File* menu, choose  *Git*  *Clone Repository* .

i Note




You can clone a Git repository only if your Git user settings have been defined. You can do this in the *Git Settings* dialog box from the *Tools* menu. If you have not defined these settings, then before the *Clone Repository* dialog box opens, SAP Web IDE checks if your Git user name and email address exist. If not, SAP Web IDE extracts this information from the identity provider defined in your account and updates Git.

2. In the *URL* field, enter the Git repository URL and press .
3. If your remote Git system works with Gerrit, select *Add configuration for Gerrit*.
4. Choose *Clone*. The cloning starts. When the process is finished, the content of the repository appears in the workspace.

7.1.12.4.1 Configure Git Repositories

You can configure the Git repository for your project by creating new entries or deleting and editing existing entries.

Procedure

1. Right-click a Git repository, and choose  *Project Settings*  *Git Repository Configuration* .
2. Configure the Git repository as follows:
 - To create a new entry, choose *Add Entry* and type the relevant values in the *Key* and *Value* fields.



i Note

Use the following format for the *Key* entry: `<section>.<name>`. For example, `user.name`

i Note

The *Key* field is mandatory and cannot be duplicated.

- To edit an entry, choose  (Edit).

- To delete an entry, choose  (Delete).
3. Choose  (Save).

7.1.12.5 Initialize a Local Git Repository

You can initialize a local repository for any project that is not already connected to a Git repository.

Context

You can create an empty local repository for your project. This local repository can then be connected to a remote repository.

Procedure

1. Select the desired project.
2. Right-click and select  *Git*  *Initialize Local Repository* .








Related Information

[Set a Remote Repository \[page 372\]](#)

7.1.12.6 Set a Remote Repository

After initializing a local repository for your project, you likely will want to set a remote repository for your project, so you can push your work to a central Git repository.

Procedure

1. Select a project that has been initialized as a local Git repository (with  *File*  *Git*  *Initialize Local Repository* .
2. Right-click and select  *Git*  *Set Remote* .
3. Enter the remote repository URL, and a name for the remote repository.

i Note

Once you have set a remote repository, you can change it from the project settings.

1. Right-click the desired project and select *Project Settings*.
2. Select *Git Repository Configuration*.
3. Click *Add Entry* and add the following:

Key	Value
remote.origin.fetch	<code>+refs/heads/*:refs/remotes/origin/*</code>
remote.origin.url	<Remote Git repository URL>
branch.master.merge	<code>refs/heads/master</code>
branch.master.remote	<code>origin</code>

i Note

When you connect your local repository to a remote repository, before you push your changes you need to first perform a fetch. Otherwise you will not be able to see the remote branches, such as `origin/master`.

4. If your remote Git system works with Gerrit, select *Add configuration for Gerrit*.
5. Choose *OK*.

Related Information

[Initialize a Local Git Repository \[page 372\]](#)

7.1.12.7 Fetch Changes

Fetching enables you to download objects and references from another repository into your local repository. You can then merge or rebase the changes into your project.

Procedure

1. In the *Git Pane*, choose *Fetch*.
2. Choose *Remember Me* to avoid being asked for credentials again in this session.
3. Choose *OK*.

4. Choose *OK*.

7.1.12.8 Rebase Changes

Rebasing enables you to take all the committed changes from one branch and incorporate them into a different branch.

Procedure

1. In the *Git Pane*, choose *Rebase*.
2. Select the branch from which you want to obtain the changes.

i Note

The branch that is currently checked out is automatically disabled. It cannot be selected. By default, the corresponding remote branch is selected.

3. Choose *OK*. The latest changes are integrated and shown in your workspace.

i Note

Rebase can fail due to conflicts between the current branch and the branch whose changes you want to incorporate. When conflicts are identified, the Git pane switches to *Rebase Interactive* mode and different actions are available to enable conflict resolution.

Related Information

[Rebase Interactive Mode \[page 374\]](#)

7.1.12.8.1 Rebase Interactive Mode

When conflicts occur while rebasing a branch, rebase interactive mode is triggered.

When rebase interactive mode is enabled, the caption `[rebase in progress]` is displayed next to the repository name in the Git pane.

The following actions can be executed in the rebase interactive state:

- Continue the rebase process.
 1. Fix the conflicts that are visible in the status table and save the fixed files.
 2. Stage the fixed files.

3. Choose *Continue*.
4. If all conflicts are resolved, the repository returns to its normal mode. If errors still exist, repeat the procedure.

i Note

You can only continue a rebase process if your Git user settings have been defined. The IDE checks if your Git user name and email address exist. If not, a dialog box opens and by default displays the user name and email that was extracted from the identity provider defined in your account. You can either correct these entries or enter new details.

- Abort the rebase process.
Choose *Abort*.
- Skip a specific conflicting commit.
Choose *Skip Patch*.
- Reset changes. Resetting changes discards all staged and unstaged changes on the current local branch, so that it is identical to the remote branch.
Choose *Reset*.
- Use the compare editor to easily toggle between the changes and resolve any conflicts that exist.

7.1.12.9 Merge Changes

You can incorporate all the changes from one branch into another in a single commit.

Procedure

The IDE recognizes merge conflicts. The differences are highlighted and you can choose whether to accept the local change or the new one. You can also accept both modifications at the same time.

1. In the *Git Pane*, choose *Merge*.
2. Select the branch from which you want to obtain the changes.
3. Choose *OK*. The latest changes are integrated and shown in your workspace.

i Note

Merge operations can fail due to conflicts between the current branch and the branch you chose from which to incorporate the changes.

7.1.12.10 Pull Changes

Pulling is the same as fetching and merging. Pulling enables you to download objects and references from another repository into your local repository, and then merge the changes into your project.

Procedure

1. In the *Git Pane*, choose *Pull*.
2. Perform one of the following:
 - If the Git repository uses the SSH protocol, choose *Browse*, navigate to your SSH private key, and open it.
 - If the Git repository uses the HTTPS protocol, enter your Git repository user name and password.
3. Choose *Remember Me* to avoid being asked for credentials again in this session.
4. Choose *OK*. The changes are fetched from the specific branch and merged into your local checked-out branch.

7.1.12.11 Stage Files

The staging table shows changed files, and lets you select files to stage.

Context


Whenever a file is updated, added, or deleted, it appears in the staging table, but is not staged. After staging, you can commit the staged files.


Procedure

In the staging table of the *Git Pane*, choose the *Staged* toggle button to the right of the change that you want to stage.

i Note

To stage all files in the staging table, select the checkbox at the top of the table, and choose *Stage Selected* from the dropdown list.

You can edit a file listed in the staging table by clicking  (More Actions) and choosing *Edit*. This option is enabled for all files, except for deleted.

You can delete a file listed in the staging table by clicking  (More Actions) and choosing *Delete*. This removes the file from the staging table and from the workspace.

Related Information

[Discard Changes \[page 378\]](#)

7.1.12.11.1 Compare and Merge Code


You can compare different versions of your code and merge the changes.

Context

Use the compare editor to compare a previous version of your code (the server version), with your modified version from the staging table in the *Git Pane*.







You can also compare two committed versions from the *Git History* pane.






Procedure

1. In the *Git Pane*, double-click on a modified file (*) in the staging table, or click  (More Actions), and choose *Compare*.

The title of each pane indicates whether it contains the current modified or the previous read only version.

2. Compare your modified code with the original code as follows:

- Click  (Sync/Unsync scrolling between panes) to decide if you want both panes to scroll together or independently.
- Click  (Next) and  (Previous) to navigate between the highlighted differences in the code of the original and modified versions. As you navigate through the changes in the file, the color of the changed code deepens to indicate your cursor position.
- Click  (Show/Hide trailing spaces) to decide if you want to include the trailing white spaces in the compare.
- Click  (Undo) and  (Redo) to undo or redo changes.

- Click  (Accept my changes) to add your changes to the current file. Click  (Accept their changes) to overwrite your changes with the previous changes.
 - Click  (Push to left) to move selected code from the previous version to the modified version. The highlighted lines on the right side will replace the highlighted lines on the left side.
3. If there is a conflict between the 2 versions, a three-way merge editor opens.
- The left pane contains your changes, the right pane contains the previous changes, and the middle pane is the editor in which you perform the code merge.
- Click  (Push to left) to move selected code from the original version to the merged version.
 - Click  (Push to right) to move the selected code from your version to the merged version.

7.1.12.11.2 Discard Changes

Discarding removes all changes from an existing file in the local environment. For example, discarding a new file deletes the file from the branch.

Context

i Note

Only unstaged files can be discarded.

Procedure

In the staging table of the [Git Pane](#), select the checkbox in the row that contains the change that you want to discard, and choose [Discard](#).

All changes that you made to the file are removed.

i Note

To discard all files, select the checkbox at the top of the table, and choose [Discard](#). All unstaged files in the staging table are discarded.

7.1.12.11.3 Stash Changes

If you made some changes that you are not yet ready to commit, you can stash (store away) and revert them from your working directory, and resume working on them later.

Context

i Note

This feature is not available in SAP Web IDE personal edition.

Procedure

1. In the *Git Pane*, choose the *Stash* button.
2. (Optional) Modify the *Description* for the stash. By default, the description is *on <branch> : <creation time stamp>*.

Each time you perform a stash operation, the new stash is saved at the top of the stash list. Meaningful descriptions will help you to locate the stash that you want to apply.

3. Choose *Stash*.

Results

The changes are stored in a new stash, and reverted from your working directory. The list of changed files is emptied.

Apply Stashes

Context

If you have stashed changes, you can apply the stashed changes to the files in your working directory by clicking the *Apply Stash* icon in the Git pane toolbar.

You may need to resolve conflicts between the version of a file in your stash and the version in your working directory.

- If the file in your working directory has been committed and has conflicts with the stashed changes, the conflicts are shown in the file in the working directory.
- If the file in your working directory has not been committed, you may not be able to apply the stash. Either commit or reset the changes to the file in your working directory, and then apply the stash again.

i Note

It is recommended to stage the newly created files before stashing them. For example, consider the following scenario:

1. Create file 1.
2. Change file 2.
3. Stash your changes. This stashes both new file 1 and the changes to file 2.
4. Change file 2 and commit the changes.
5. Apply the stash.

In this case, the stashed changes to file 2 are applied successfully, though you will get a conflict to resolve. But the new file 1 is not applied and is lost.

You can avoid this issue by staging the new file 1 before stashing, or even committing the new file and not stashing it.

Procedure

1. Choose *Apply Stash*.
2. From the dropdown list, select the stash that you want to apply.
3. Choose one of the following options:
 - *Apply*: Reapply the stashed changes to your working directory, and keep the stash in the stash list.
 - *Pop*: Reapply the stashed changes to your working directory, and remove the stash from the stash list.
 - *Drop*: Delete the stashed changes so they can no longer be applied to your working directory. The working directory is unaffected.
4. Choose *Continue*.

7.1.12.12 Commit Changes

You can commit changes to the repository locally.

Procedure

1. In the *Git Pane* status table, select the *Stage* checkbox for the files you want to stage (or click *Stage All* above the table).

In the status table, you can double-click a row to see the differences between the current file and the previous (HEAD) version.

2. Enter a description of the change in *Commit Description*.
3. If you want to add the current changes to the last commit, select the *Amend Changes* checkbox. The commit description of the last committed change appears in *Commit Description*, which you can modify.

4. Choose [Commit](#). The changes are committed locally, and one is added to the counter for unsynced commits at the top of the [Git Pane](#), next to the repository name.

i Note

You can only commit a change to the repository if your Git user settings have been defined. The IDE checks if your Git user name and email address exist. If not, a dialog box opens and by default displays the user name and email that was extracted from the identity provider defined in your account. You can either correct these entries or enter new details.

7.1.12.13 Push Changes

The Push option incorporates all unsynced committed changes into the remote branch of the currently checked-out local branch. The number of unsynced committed changes is displayed next to the repository name. All tags created within the open repository are pushed.

Context

The process flow for pushing code changes can differ between projects according to the [Code Checking Triggers](#) configuration in the project settings.

- No code checking is performed and all changes are pushed.
- Code checking is performed and problems are found. The push is not started and notification is sent about the problems. You can choose to fix the problems and try to push again or to push anyway.
- Code checking is performed and problems are found. The push is blocked and notification is sent about the problems. You need to fix the problems before pushing.

Procedure

1. In the [Git Pane](#), choose [Push](#).

For your convenience, you can instead use [Commit and Push](#) to commit the currently staged changes and then immediately push them to a remote branch. Before choosing [Commit and Push](#), remember to stage your changes and to add a description for the commit.

2. Choose one of the following from the dropdown list:
 - [origin/<remote branch>](#) if your local branch is based on a specific remote branch.
 - [Remote Branch](#) to select a different remote branch.
3. Perform one of the following, according to the format of the Git repository that you selected:
 - If the Git repository uses the SSH protocol, choose [Browse](#), navigate to your SSH private key, and open it.
 - If the Git repository uses the HTTPS protocol, enter your Git repository user name and password.

4. Choose *Remember Me* to avoid being asked for credentials again in this session.

Results

If the project settings are configured to notify, you will receive notification about problems before the push starts:

i Note

This feature is not available in SAP Web IDE personal edition.

- You can click *View Problems* to see information and fix the problems before pushing.
- If the push process is not blocked, you can choose *Push* to continue with the push anyway.

If the project settings are configured not to notify, code is pushed to the source control repository without code checking.

Example

Notification is configured for errors and warnings, and the push process is blocked if errors are found.

- You receive notifications about problems with warning severity only.
You can choose to continue with the push, or you can view and fix the problems before pushing.
- You receive notification about problems with error and warning severities.
You can view the problems but you cannot continue with the push until the errors are fixed.

7.1.12.14 Multiple Local Branches

From the *Git Pane*, you can check out a local branch, add a new local branch, and remove a local branch.

Checking Out a Local Branch

Procedure

In the *Git Pane*, select the desired local branch.

i Note

If you have uncommitted changes in your workspace, a dialog box containing the list of conflicting files opens. Choose *Cancel* to abort, or *Reset and Checkout* to remove all uncommitted changes.

The selected branch is checked out. The name of the selected branch is shown in the workspace next to the name of the project.

Creating a New Local Branch

Context

You can create a new local branch referencing any available remote or local branch.

Procedure

1. In the *Git Pane*, choose **+** (Add Branch).
The *Create a New Branch* dialog box appears.
2. From the *Source Branch* dropdown list, select the desired local or remote branch.
3. Enter a name for the new local branch.
4. Choose *OK*.

i Note

If you have uncommitted changes in your workspace, a dialog box containing the list of conflicting files opens. Choose *Cancel* to abort, or *Reset and Checkout* to remove all uncommitted changes.

The new local branch is created and checked out.

Deleting a Local Branch

Context

If there is only one branch available, it cannot be deleted.

Procedure

1. In the *Git Pane*, choose **-** (Delete Branch). The *Delete Branch* dialog box appears showing all the branches of the selected repository.
2. Select one or more branches that are not checked out.
3. Choose *Delete*.

The selected branches are deleted.

7.1.12.14.1 Reset Local Branches

You can delete all new objects and references that were added to an existing local branch to make it identical to its remote branch.

Context

When you reset a branch, all unsynced committed changes are removed, and all staged and unstaged files are reverted to their original state in the local copy of the respective remote branch.

Procedure

1. In the *Git Pane*, choose *Reset*.
2. Select the branch that you want to revert back to.

i Note

The branch that is currently checked out is automatically disabled. It cannot be selected. By default, the corresponding remote branch is selected.

3. Choose a *Reset Type*.

i Note

If you choose a *Hard* reset, all changes are removed.

4. Choose *OK* to reset the branch.

7.1.12.15 Create Remote Branches

You can create a new branch in the remote repository.

Context

To work on the remote branch you just created, you still must check out the branch by creating a new local branch for this new remote branch.

i Note

To create a local branch, you can choose the menu **Git > Create Local Branch** or click the plus sign in the *Git Pane*.

Procedure

1. Select your project.
2. From the menu, select **Git > Create Remote Branch**.

Note

Deleting remote branches is currently not supported. You can delete a remote branch using a different Git client.


Related Information

[Multiple Local Branches \[page 382\]](#)

7.1.12.16 Git History

From the Git *History* pane, you can explore the history of committed changes that were made for repositories, folders, and files in a specific project.

Git history is located in a dedicated pane that you can access in one of the following ways:

- Using the main menu: **File > Git > History**.
- Using the context menu of the selected Git repository.
- Using the dedicated icon  (Git History pane) from the right sidebar.
- Using the *Show Git Blame* option from the context menu of the file editor.

Once you have selected a Git repository in the workspace, use the Git *History* pane to:

- Explore the history of any branch of your committed code by selecting one or more branches from the *History of Branch* dropdown list. When you first select a file or folder in the workspace, details of the master branch are displayed in the Git *History* pane by default – you can either select a different branch or all branches.
- View a list of commits (descriptions of the commits) in the central area of the pane. A commit graph on the left gives you a visual representation of the commit history.

Any tagged commit is shown with the tag icon (); hover over the icon to see a list of tags.

- Search commits by:
 - The author of the commit
 - Person who committed the change
 - Commit ID
 - DateThe results of your search are highlighted and you can toggle between them using the arrows on the right of your filter.
- Select a commit in the list to view details of that commit in the area below.

- View a list of all the files packaged in this commit in the *File* column in the area below. From the *Status* column, you can see one of the following statuses of the file:

Status	Description
N	New
D	Deleted
M	Modified
C	Contains conflicts

- Execute the following Git commands:
 - Tag a commit.
 - Cherry-pick a change.
 - Revert a commit.
 - Check out a commit.
 - Compare commits.

[Git Commands from the Git History Pane \[page 386\]](#)

From the Git *History* pane, you can execute a number of Git commands.

[Git Blame \[page 388\]](#)

Use the Git Blame feature to view the details of the last revision for each line in a code file.

7.1.12.16.1 Git Commands from the Git History Pane

From the Git *History* pane, you can execute a number of Git commands.

Context

Following are the main Git commands that you can execute from the Git *History* pane.

Operation	Description	Steps
Tag a commit	Allows you to mark specific points in history as important, usually used for release points.	<ol style="list-style-type: none"> 1. Choose <i>Tag</i>. 2. Enter a name for the new tag. 3. Choose <i>OK</i>.

Cherry-pick a change

Allows you to apply the changes of the selected commit to only the current branch.

i Note

You can only cherry-pick a commit if your Git user settings have been defined. The IDE checks if your Git user name and email address exist. If not, a dialog box opens and by default displays the user name and email that was extracted from the identity provider defined in your account. You can either correct these entries or enter new details.

1. Choose *Cherry-Pick*.
2. Choose *OK* to confirm the commit.
3. If there is a conflict, a warning is displayed. From the Git pane, resolve any conflicts as follows:
 1. Open the Git pane and find the conflicting files in the staging table.
 2. Edit the files to resolve the conflicts.
 3. Stage the edited files.
 4. Commit the changes.

Revert a commit

Allows you to undo all the changes that were incorporated in the selected commit.

i Note

You can only revert a commit if your Git user settings have been defined. The IDE checks if your Git user name and email address exist. If not, a dialog box opens and by default displays the user name and email that was extracted from the identity provider defined in your account. You can either correct these entries or enter new details.

1. Select the commit in the list of commits and choose *Revert*.
2. If there is a conflict, a warning is displayed. Open the Git pane to resolve any conflicts as above.

Check out a commit

Allows you to take the code from a specific commit and create a new branch based on it.

1. Select the commit in the list of commits, and choose *Check Out*.
2. Enter a name for the new branch.
3. Choose *OK*. The new branch appears as checked out in the Git pane, and all the commit content is refreshed in the workspace.

i Note

When checking out an already committed change, conflicts can occur. To resolve these conflicts, make sure that the conflicting file is not open in the staging table.

Compare commits Allows you to compare a modified version of your code with the original version from the staging tables in the Git pane.

You can also compare two committed versions.

Double-click the changed file to open a compare view in the code editor, so that you can compare the different versions of the file.

For more information, see [Compare and Merge Code \[page 377\]](#).

To compare two committed version:

1. Select two commits from the list.
2. Select the common file to compare.
3. Choose [Compare](#).

7.1.12.16.2 Git Blame

Use the Git Blame feature to view the details of the last revision for each line in a code file.

When you open a code file in the code editor, and choose [Show Git Blame](#) in the context menu, a read-only file named `Blame-<original file name>` opens in a new tab. Details of the last revision, such as commit ID, date, and author, are displayed for each code line in the file's gutter.

You can further explore the revision details:

- Hover over a line to display a tooltip with more revision details.
- Click a line to open the respective commit in the [History](#) pane and display all the commit details.

Press **Esc** to hide the blame annotations.

7.1.12.17 Set Up Git to Work with Gerrit

Gerrit is a web-based software code review tool for reviewing, approving, or rejecting changes to the source code developed by your colleagues. Gerrit works as an intermediate environment for source control between the local environment and the remote Git repository.

Procedure

1. From the **File > Git** menu, choose [Clone Repository](#) or [Set Remote](#).

If you select [Set Remote](#), your project must have been initialized as a local repository with **File > Git > Initialize Local Repository**.

2. Select the [Add configuration for Gerrit](#) checkbox.

Results

Anytime you push, the changes will be sent to Gerrit for code review.

i Note

Make sure to only select the checkbox if your Git uses Gerrit.

Related Information

[Clone Repositories \[page 371\]](#)

[Initialize a Local Git Repository \[page 372\]](#)

[Set a Remote Repository \[page 372\]](#)

[View Changes in the Gerrit Pane \[page 390\]](#)

[Fetch Changes from Gerrit \[page 391\]](#)

Manual Setup

You can still set up a local Git repository to work with Gerrit even if you did not specify the Gerrit configuration when cloning or setting a remote repository.

Procedure

1. Right-click your project and choose **► Project Settings ► Git Repository Configuration ▾**.
2. Choose **Add Entry**.
3. In the **Key** field, enter **gerrit.createchangeid**
4. In the **Value** field, enter **true**..
5. Choose **OK**.

7.1.12.17.1 View Changes in the Gerrit Pane

You can view your open changes in your Gerrit system from the [Gerrit](#) pane, and navigate directly to a specific change in your Gerrit system. You can also submit changes that are ready to be merged.



Prerequisites

You have enabled the [Gerrit](#) pane by going to [Preferences](#) > [Git Settings](#) and selecting [Show Gerrit Pane](#).

Context

When you push a change to Git in a project configured to use Gerrit, your open changes can be viewed in the Gerrit pane – both changes you submitted and changes for which you are a reviewer.

Procedure

1. Open the [Gerrit](#) pane by selecting  in the right-side panel.
You can show all your changes, only the changes you own, or only changes you have been added to as a reviewer.
2. Click the [Refresh](#) icon  to get the latest changes.
For each change you can see the change name, the developer who committed the change, whether the change was code reviewed ([CR](#)), and whether the change was verified ([V](#)).
3. Click on a change to open the change in your Gerrit system.

i Note

If the change is ready to be merged, you can click on [Submit](#).

Related Information

[Set Up Git to Work with Gerrit \[page 388\]](#)

7.1.12.17.2 Fetch Changes from Gerrit

When your repository is set up to work with Gerrit, you work with Git as normal, and review code changes in your Gerrit system. You can also fetch a change from Gerrit and create a local branch from the change, and then collaborate with a colleague using Gerrit before merging the change in Git.

Procedure

1. From the workspace, select your project.
2. Open the [Git Pane](#).
3. Choose [Fetch from Gerrit](#).
4. Enter the change that you want to download, and choose [OK](#).

Use the ref specification for the change, in the form of `refs/changes/79/2565079/3`. This is available under [Download](#) in the standard Gerrit UI.

The changes are fetched into a new local branch, and the branch is checked out.

i Note

If you have uncommitted changes in your workspace, a dialog box containing the list of conflicting files opens. Choose [Cancel](#) to abort, or [Reset and Checkout](#) to remove all uncommitted changes.

Results

When you change or update your code in a repository associated with Gerrit, and then stage, commit and push the changes to Gerrit, a notification confirms that the push operation was successful. A link to the committed change in the Gerrit tool is displayed.

7.1.13 Running Applications in Development Mode

Evaluate the progress of your application's development by running an application to test in-development functionality and design.

You can test your application from the workspace, running in simulators, on devices, with mock data, in the SAP Fiori launchpad environment, or with predefined URL parameters. You can define run configurations to determine how to preview your application. For more information, see [Create Run Configurations \[page 394\]](#).

If your HTML5 application reuses applications that are contained in your workspace or SAP BTP, you can preview your HTML5 application together with these reuse applications.

SAP Web IDE implements the application cache buster feature of SAPUI5 to improve performance when previewing applications. For more information, see [Application Cache Buster](#).

i Note

To run your application, there must be a destination to the SAP Web IDE backend with following configuration:

≡, Sample Code

```
Name=webide_di
Type=HTTP
TrustAll=true
Description=Destination to SAP Web IDE backend
URL=https://disapwebide-<id of consumer account>.<host> //(e.g. https://
disapwebide-wbac0632b.eu1.hana.ondemand.com)
ProxyType=Internet
Authentication=AppToAppSSO
WebIDEEnabled=true
WebIDEUsage=true
```

For more information, see [Configure Destinations from the Cockpit](#).

[Run Applications from the Workspace \[page 392\]](#)

Run and test your application using different run configurations.

[Create Run Configurations \[page 394\]](#)

Create run configurations that define how your project or unit test is executed.

[Run Applications in a Frame \[page 399\]](#)

Check the localizations and functionality of your application by running it in a frame..

[Run Applications in the SAP Fiori Launchpad Environment \[page 400\]](#)

Test your application within the SAP Fiori launchpad environment, a runtime shell that hosts SAP Fiori applications and provides the applications with services such as navigation, personalization, embedded support, and application configuration.

[Run Unit Tests on Applications \[page 402\]](#)

You can run your application as a unit test to determine if it is working properly.

[Run Applications with Mock Data \[page 402\]](#)

Run a web application using a client mock server to test your application without connecting to the OData provider.

7.1.13.1 Run Applications from the Workspace

Run and test your application using different run configurations.

Context

You can run and test an application using different parameters to ensure the quality and robustness of the application. You can create and save run configurations with different parameters and then run the application or test it with these run configurations without adjusting the application code. For more information about run configurations, see [Create Run Configurations \[page 394\]](#).

If you do not create a new run configuration or select an existing one, SAP Web IDE uses default run configurations that use the SAP Web IDE default settings as follows:

Run As	Default Behavior	Default Settings
Web Application	<ul style="list-style-type: none"> • If you select an <code>HTML</code> file in your project and run it from the context menu or from the main toolbar, the <code>HTML</code> file runs as is and a new web application run configuration is created in the background. • If you select any project file (can be a non-<code>HTML</code> file) and run it, SAP Web IDE searches all the <code>HTML</code> files in the project. <ul style="list-style-type: none"> ◦ If no matched files are found, an error message prompts you to configure a new run configuration for the web application. For more information, see Create Run Configurations [page 394]. ◦ If a single match is found, a new web application run configuration is created in the background and the file runs automatically. ◦ If a few matches are found, a dialog box opens displaying a list of all the matched files. Choose the file that you want to run and choose <i>OK</i> to create a new web application run configuration in the background. The file runs automatically. 	<ul style="list-style-type: none"> • Open without a frame • Run without mock data • Run with all existing URL parameters
SAP Fiori Launchpad Sandbox	<ul style="list-style-type: none"> • SAP Web IDE uses the selected <code>component.js</code> file and runs it from the context menu or from the main menu in the SAP Fiori launchpad sandbox. For more information, see Run Applications in the SAP Fiori Launchpad Environment [page 400]. • If there is no selected <code>component.js</code> file, SAP Web IDE looks in the <code>project/src/main/webapp</code> file path or in the project route. 	<ul style="list-style-type: none"> • Open without a frame • Run without mock data • Run with all existing URL parameters
Unit Test	<ul style="list-style-type: none"> • If you select an <code>HTML</code> file in your project and run it as a unit test from the context menu or from the main menu, the <code>HTML</code> file runs as is and a new unit test run configuration is created in the background. • If you select any project file (can be a non <code>html</code> file) and run it, SAP Web IDE searches all the <code>HTML</code> files in the project that include <code>qunit</code> or <code>testsuite</code> in their names. <ul style="list-style-type: none"> ◦ If no matched files are found, an error message prompts you to configure a new run configuration for the unit test. For more information, see Create Run Configurations [page 394]. ◦ If a single match is found, a new unit test run configuration is created in the background and the file runs automatically. ◦ If a few matches are found, a dialog box opens displaying a list of all the matched files. Choose the file that you want to run and choose <i>OK</i> to create a new unit test run configuration in the background. The file runs automatically. 	

You can run a project in any of the following ways:

Procedure

- Select a project and in the main toolbar, choose *Run*.

The project runs with the run configuration that is displayed in the drop-down menu, which is the last-used run configuration. You can choose a different run configuration in the list. All the displayed run configurations can be used with the selected project.

- In the project's context menu, choose *Run*.
 - Choose a recently-used run configuration from the list. The list displays only configurations that you can run for the selected project.
 - Choose *Run As* and select a project type (for example, *Web Application* or *SAP Fiori Launchpad Sandbox*), or select *Unit Test*.
- Create or edit a run configuration and run a project using this configuration.

To open the *Run Configurations* page, in the project's context menu, choose  *Run*  *Run Configurations* .

Results

i Note

If you are making HTTP requests from your application and you need to send headers, you need to list the header names in the `neo-app.json` file in a top-level attribute called `headerWhiteList`. For example, the following enables you to send the headers `apikey` and `batch-operation` in any HTTP request.

```
"headerWhiteList": ["apikey", "batch-operation"]
```

For more information, see [Header Whitelisting](#).



7.1.13.2 Create Run Configurations

Create run configurations that define how your project or unit test is executed.


Context

SAP Web IDE provides default run configurations for your projects. You can create and configure additional run configurations that define how your project or unit test is executed. You can also create a new run configuration by duplicating an existing run configuration and then editing it.

Procedure

1. From the context menu of any file in your project, choose  **Run** > **Run Configurations** .

The run configurations that are relevant for the project types that are defined for your project are displayed. For more information, see [Set Project Types \[page 117\]](#).

2. In the **Run Configurations** window, create a new configuration from scratch or duplicate an existing configuration.
 - Choose **+** and select the type of run configuration that you want to create, for example, **Web Application** or **Unit Test**.
A new configuration with a default name appears under the category that you selected. The default name is `Run <run_application_filename>`. If the run application file is not known, the default name is `Configuration`.
 - Select an existing run configuration and choose  (**Duplicate**).
A new configuration with the default name `Copy of <configuration_name>` appears under the category of the configuration that you duplicated.

You can now edit the configuration that you created.

3. Change the name of the run configuration, if required, and edit the run configuration settings in each tab.
4. To save the configuration and run your project or unit test with this configuration, choose **Save and Run**.
To save changes to the run configuration without running a project, choose **OK**.

[General Tab \[page 395\]](#)

Define general settings for the run configuration.

[URL Components Tab \[page 397\]](#)

Define navigation information within the application.

[Advanced Settings Tab \[page 398\]](#)

Define advanced settings for the run configurations.

7.1.13.2.1 General Tab

Define general settings for the run configuration.

If the tab contains a required field that is incomplete or incorrect, an error icon appears by the tab name and the relevant field is outlined in red. Hover over the field to display an error message that describes how to fill in the field correctly.

Setting	Description
Run Application File	<p>The file that is used to run the application.</p> <ul style="list-style-type: none"> • If you chose <i>Run Configurations</i> from the context menu of a file that runs the application, this file name is automatically displayed. • If you chose <i>Run Configurations</i> for a project that has only one file that can run the application, this file name is automatically displayed. • If there are more files can that run the application, you can select the required file from the drop-down list.
Preview Mode	<p>By default, your application opens without a frame. This preview mode makes issue detection more apparent, as SAP Web IDE runs only the application.</p> <p>Select <i>With Frame</i> to open your application in a frame with configurable viewing options. For more information, see Run Applications in a Frame [page 399].</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>i Note</p> <p>This option is not available for unit tests.</p> </div>
Mock Data	<p>Select <i>Run with mock data</i> to use mock data in your application.</p> <p>If you use mock data for your application, make sure that:</p> <ul style="list-style-type: none"> • You have configured settings for the mock server. For more information, see Configure Mock Data Usage [page 116]. • The application that you want to run uses the SAPUI5 OData model with JSON format. <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>i Note</p> <ul style="list-style-type: none"> • This option is only available for projects of type SAP Fiori • This options in only available for services of type OData Version 2. • This option is not available for unit tests. </div>
Support Assistant	<p>Select <i>Run with Support Assistant</i> to enable the Support Assistant tool in your web application.</p> <p>Use the Support Assistant tool to check whether your application is built according to the best practices for building SAPUI5 apps. The tool uses a set of pre-defined rules to check all aspects of an application, for example, accessibility, performance, and data-binding. With a simple click, you can check the current state of your application. After execution, you can check the results and apply corrective measures based on the outcome. The tool aims to reduce maintenance and consulting times and to streamline SAPUI5 app development.</p> <p>For more information, see Support Assistant.</p>

Parent topic: [Create Run Configurations \[page 394\]](#)

Related Information

[URL Components Tab \[page 397\]](#)
[Advanced Settings Tab \[page 398\]](#)

7.1.13.2.2 URL Components Tab

Define navigation information within the application.

i Note

These options are not available for unit tests.

If the tab contains a required field that is incomplete or incorrect, an error icon appears by the tab name and the relevant field is outlined in red. Hover over the field to display an error message that describes how to fill in the field correctly.

Setting	Description
URL Parameters	<p>You can define additional parameters as name-value pairs to be used when running the application. In the full application URL, the URL parameters are preceded by a question mark character (?).</p> <ol style="list-style-type: none">To add a new row for an additional parameter, choose Add Parameter.In the new row, enter a parameter name and its value.
URL Hash Fragment	<p>To define a hash fragment (also known as a fragment identifier) for the URL, in the URL Hash Fragment field, enter the fragment identifier without the hash delimiter.</p> <p>In the full navigation URL, the fragment identifier is appended after the URL parameters, and is preceded by a hash (#) delimiter.</p>

Parent topic: [Create Run Configurations \[page 394\]](#)

Related Information

[General Tab \[page 395\]](#)
[Advanced Settings Tab \[page 398\]](#)

7.1.13.2.3 Advanced Settings Tab

Define advanced settings for the run configurations.

i Note

This feature is not available in SAP Web IDE personal edition.

These settings override configurations in the `neo-app.json` application descriptor file when you run the application. No changes are made to the `neo-app.json` file.

If the tab contains a required field that is incomplete or incorrect, an error icon appears by the tab name and the relevant field is outlined in red. Hover over the field to display an error message that describes how to fill in the field correctly.

Setting	Description
SAPUI5 Runtime Settings	<p>By default, your project uses the SAPUI5 version that is specified in the project <code>neo-app.json</code> file at runtime. If the version is not specified there, the project uses the latest official version of SAPUI5.</p> <p>Alternatively, you can choose a specific SAPUI5 version that will be used with this run configuration. Select Use another version and select the required version.</p> <p>i Note</p> <p>If a SAPUI5 version is specified in the project <code>neo-app.json</code> file, that is the default version.</p> <p>i Note</p> <p>If a minimum SAPUI5 version is specified in the project <code>manifest.json</code> file, the version list displays only versions equal to or higher than the specified version.</p>
Application Destination Resources	<p>You can test your application with a destination system that is different from the one defined in the application <code>neo-app.json</code> file. Map the destinations that are defined in your project to any system that is included in your SAP BTP account.</p>
Application Resources	<p>If your project references any SAPUI5 resources, you can change the version of the library to use for this run configuration. By default, the version that you selected when you made the reference is used.</p> <p>Choose Get Library Versions to see a list of referenced libraries. Choose the version to use for this run configuration.</p> <p>If you are working on library projects and want to use the version of a library currently in your workspace, choose Use my workspace first. If you no longer have a version of the library in your workspace, the selected version on SAP BTP is automatically used.</p>

Parent topic: [Create Run Configurations \[page 394\]](#)

Related Information

[General Tab \[page 395\]](#)

[URL Components Tab \[page 397\]](#)

7.1.13.3 Run Applications in a Frame

Check the localizations and functionality of your application by running it in a frame..



Prerequisites

- Run the application with a frame, and ensure that the *With Frame* option is set in the run configuration that you want to use.
The frame keeps the menu bar visible, and allows you to toggle between all predefined and custom simulators.
- Know which HTML file loads the application. There might be multiple HTML files in one project; therefore, when running an application for the first time, you must select the correct one.

i Note

SAP Web IDE remembers the HTML file on subsequent run actions: the previously selected file is used until you choose a new one from the current project.

Procedure

1. Select a file in the project that you want to run.
2. Choose *Run* and choose the relevant run configuration.
 - To preview the application in different view modes, select the *Switch Device* icon () and select large/desktop, medium/tablet, or small/mobile. Each time you select a different menu option, the proportions change accordingly. You can also select a custom dimension that fits the size of a specific device.
 - To toggle the orientation, choose the *Switch Orientation* icon ()
 - To validate the languages that the application supports, select a *Language*. The application re-renders accordingly.

i Note

Languages vary from application to application. If an application supports multiple languages, codes are read from the `.project.json` file in the order in which they appear. For example, `"supportedLanguages": "en, fr, de, zh_cn"` in `.project.json`, displays English, French, German, and Simplified Chinese in the SAP Web IDE language preview.

7.1.13.4 Run Applications in the SAP Fiori Launchpad Environment

Test your application within the SAP Fiori launchpad environment, a runtime shell that hosts SAP Fiori applications and provides the applications with services such as navigation, personalization, embedded support, and application configuration.

The local sandbox environment for SAP Fiori launchpad is a simplified environment that you can use for local development and testing. This allows you to ensure that the application can be embedded properly into SAP Fiori launchpad. The sandbox shell implementation uses local configuration files instead of ABAP or SAP HANA back-end services.

[Test a Single SAP Fiori Application \[page 400\]](#)

Test your SAP Fiori application in a simplified SAP Fiori launchpad environment.

[Test Multiple SAP Fiori Applications in the FLP Sandbox \[page 401\]](#)

Test the interaction of multiple SAP Fiori applications in a simplified SAP Fiori launchpad environment.

7.1.13.4.1 Test a Single SAP Fiori Application

Test your SAP Fiori application in a simplified SAP Fiori launchpad environment.

Prerequisites

- You have created an SAP Fiori application in SAP Web IDE. A `Component.js` file was automatically created and is accessible in your workspace.
- You have selected *SAP Fiori* as a project type. For more information, see [Set Project Types \[page 117\]](#).

Procedure

1. In the workspace, select the `Component.js` file.
2. In the main toolbar or context menu, choose *Run*.
 - Choose a recently-used run configuration from the list. The list displays only configurations that you can run for the selected project.
 - Choose *Run as SAP Fiori Launchpad Sandbox*. The application runs with the last-used run configuration that matches the SAP Fiori component project type. If there is no matching run configuration, a default run configuration is created and used.
 - Choose *Run Configurations* to create a new run configuration and run your project. For more information, see [Create Run Configurations \[page 394\]](#).

Task overview: [Run Applications in the SAP Fiori Launchpad Environment \[page 400\]](#)

Related Information

[Test Multiple SAP Fiori Applications in the FLP Sandbox \[page 401\]](#)

7.1.13.4.2 Test Multiple SAP Fiori Applications in the FLP Sandbox

Test the interaction of multiple SAP Fiori applications in a simplified SAP Fiori launchpad environment.

Prerequisites

- Make sure you have 2 (or more) applications in your SAP Web IDE workspace.

Procedure

1. From the workspace, select the app from which you want to navigate.
2. From the context menu, select *Enable App To App Navigation*. The *App To App Navigation* dialog box opens.
3. From the *Navigate to* dropdown list, select one or more applications to which you want to enable navigation.
4. Click *Enable*. A new `FLPSandbox` project is created in the workspace. This project contains a `neo-app.json` file with routes to the origin and target applications, as well as a `fioriSandboxConfig.json` file with an intent section for each application.

For more information on the `fioriSandboxConfig.json` file, see [Local Configuration File for the Launchpad Sandbox](#).

5. Implement the navigation code in the origin application as a callback to the navigation event.

The following APIs contain information that can help you establish the navigation between apps:

- [Cross Application Navigation](#)
- [Navigation Handler](#)

Note

Make sure to use the same name for the intent in the `fioriSandboxConfig.json` file and for the navigation. This name should be aligned with the name used in the productive SAP Fiori Launchpad.

6. Right-click the *FLP Sandbox* project and select **Run > App2App Navigation** to test that the navigation works.

App2App Navigation is a run configuration generated by default when the App2App navigation is enabled. It is an *SAP Fiori Launchpad Sandbox* configuration which has the *Use my workspace first* checkbox selected.

Task overview: [Run Applications in the SAP Fiori Launchpad Environment \[page 400\]](#)

Related Information

[Test a Single SAP Fiori Application \[page 400\]](#)

7.1.13.5 Run Unit Tests on Applications

You can run your application as a unit test to determine if it is working properly.

Context

You can configure as many unit tests as you want and then run them on different HTML files in your project. There are two ways to configure and run a unit test for your application:

- Using a customized run configuration.
For more information, see [Create Run Configurations \[page 394\]](#).
- Using a default run configuration.
For more information, see [Run Applications from the Workspace \[page 392\]](#).

7.1.13.6 Run Applications with Mock Data

Run a web application using a client mock server to test your application without connecting to the OData provider.

Prerequisites

- Make sure that you have an existing web application project in SAP Web IDE that uses the SAPUI5 OData model with JSON format.
- Make sure that you have configured settings for the mock server. For more information, see [Configure Mock Data Usage \[page 116\]](#).

Context

Running a web application using a client mock server allows you to test your application without depending on the OData provider. Likewise, it allows you to work on your application while offline.

i Note

If you make changes to an application HTML file while running the application with mock data, you must rerun the application in SAP Web IDE to see the changes. You cannot see the changes by refreshing the preview window.

You can run an application with mock data in one of the following ways:

Option 1

1. From the project context menu, select **Run > Run Configurations**.
2. Create a new run configuration or select the existing run configuration for this project. The configuration should be of type *Web Application* or *SAP Fiori Launchpad Sandbox*.
3. In the *General* tab, select **Mock Data > Run with mock data**.
4. Choose *Save and Run* to save your configuration and run your project or *OK* to save your changes.

Option 2

1. In your project folder, select the HTML file used to run your project.
2. In the context menu, select **Run > Run with Mock Data**.

Related Information

[Add Custom Mock Requests \[page 405\]](#)

[Create Run Configurations \[page 394\]](#)

7.1.13.6.1 Edit Mock Data

You can model the service data that you want to use as mock data in your application.

Prerequisites

- The service must be of type OData Version 2.
- The project must be of type SAP Fiori.

Context

Depending on the project template that was used to create the project, the metadata file and mock data files are stored in the following folders:

Project	Location of Metadata File	Location of Mock Data Files
Based on new SAPUI5 project templates	webapp/localService	webapp/localService/mockdata
Other	model	model

Procedure

1. Open your project.
2. Expand the folder that contains the metadata file.
3. Right-click a `metadata.xml` or EDMX file and select *Edit Mock Data*. The *Edit Mock Data* page is displayed.

Note

If your metadata was updated in the back end, you can sync the metadata to the latest updates, and then edit the mock data.

To sync your metadata, right-click the `metadata.xml` file and select *Sync Metadata*.

If you chose to cache the metadata in your system, you will need to update it by executing these reports:

- On SAP backend: `/IWBEP/CACHE_CLEANUP`
- On SAP Gateway: `/IWFND/CACHE_CLEANUP`

4. Select an entity set.
If a JSON file for the selected entity set exists, its mock data is displayed in the Mock Data table.
5. Add, delete, or edit the mock data as necessary.
6. To add random data to an entity set, choose *Generate Random Data*.
10 rows of random data are added to the selected entity set each time you choose this option.
7. Choose *OK*.
If not previously available, a JSON file is created and added to the mock data folder for the corresponding entity set.

7.1.13.6.2 Add Custom Mock Requests

You can add a file containing custom mock requests that will be used when running applications with mock data.

Context

By default, the files containing custom mock requests are included when running the application with mock data. To change this setting, see [Configure Mock Data Usage \[page 116\]](#).

Procedure

1. Create a `mockRequests.js` file containing your desired custom mock requests. This file must contain a `getRequests` function that returns an array of the custom mock requests. Follow the file structure shown below:

```
jQuery.sap.declare("dev1.model.mockRequests");
dev1.model.mockRequests = {};
dev1.model.mockRequests.getRequests = function() {
    return [ dev1.model.mockRequests.mockAddFunctionImport() ];
};
dev1.model.mockRequests.mockAddFunctionImport = function() {
    return {
        method : "GET",
        path : new RegExp("SomeAction"),
        response : function(oXhr) {
            oXhr.respondJSON(204);
        }
    };
};
```

2. Add the `mockRequests.js` file to your project under the folder containing the `metadata.xml` and optionally the JSON mock data files.

7.1.14 Building Applications

You can build your project using the UI5 builder or the Grunt builder.

SAP Web IDE support two types of task runner platforms:

- **UI5 build**

The `package.json` and `ui5.yaml` files are mandatory for using the UI5 build capabilities in your project. The `ui5.yaml` file contains information that configures the UI5 builder's behavior and includes custom tasks. The `package.json` points to the UI5 build, the UI5 dependencies, and the custom SAP Web IDE extension.

The UI5 build is now the default build for SAP Fiori non-MTA projects. See [UI5 Command Line Interface](#) .

- **Grunt build**

The `package.json` and `Gruntfile.js` files are mandatory for using the Grunt capabilities in your project. The `Gruntfile.js` file contains configurations for running npm and Grunt. You can activate the Grunt build from your project context menu. See [GRUNT: The JavaScript Task Runner](#).

Related Information

[Run a UI5 Build \[page 406\]](#)

[Run a Grunt Build \[page 409\]](#)

7.1.14.1 Run a UI5 Build

This topic describes the UI5 build's improvements and its mandatory configuration files.

SAP Fiori projects will now include the UI5 build, by default. The UI5 build replaces the Grunt build for projects deployed to the Neo or ABAP environment.

The UI5 build introduces the following new features:

- The build is created using the standard [UI5 builder](#), which is part of the [UI5 Tooling](#).
- The UI5 build from SAP Web IDE can be triggered from any outside command-line interface (CLI) tool. See [UI5 Command Line Interface](#).
- You can easily add extensions and custom tasks to the build, providing greater flexibility. See [UI5 Build Extensibility](#).
- You can choose which files or folders to include in the `component-preload.js` and which to exclude.

The following are mandatory configuration files for the UI5 build and are generated automatically during project creation:

- `ui5.yaml` file

```
specVersion: '1.0'
metadata:
  name: <<app-name>>
  type: application
builder:
  customTasks:
    - name: webide-extension-task-updateNeoApp
      afterTask: generateVersionInfo
      configuration:
        destDir: dist
        appFolder: webapp
        namespace: <<namespace>>
    - name: webide-extension-task-updateManifestJson
      afterTask: webide-extension-task-updateNeoApp
      configuration:
        appFolder: webapp
        destDir: dist
    - name: webide-extension-task-lint
      afterTask: webide-extension-task-updateManifestJson
      configuration:
        destDir: dist
        appFolder: webapp
```

```
    nameSpace: <<namespace>>
  - name: webide-extension-task-resources
    afterTask: webide-extension-task-lint
    configuration:
      nameSpace: <<namespace>>
```

i Note

The `webide-extension-task-lint` is a task that runs static code validations on your project files and collects any issues in the `dist` folder.

If you encounter the following error when trying to build your application, this may indicate that you have an exceptionally big JavaScript file that is causing ESLint to crash with an out of memory exception:

```
FATAL ERROR: Ineffective mark-compacts near heap limit Allocation failed -
JavaScript heap out of memory
```

This error may also affect your design-time static code checks, such as LSP validations. To avoid this problem, add an `eslintignore` configuration file that ignores the problematic file:

1. Create an `.eslintignore` file in the root of your project.
2. Copy the path of the file that you want to ignore into the `.eslintignore` file

i Note

If some build files and components, such as `neo-app.json`, `di.code-validation.core_issues.json`, and `resources` can't be located in the `dist` folder after the build process is finished, they may have been misplaced under the `resources` folder.

This may occur when the project name and namespace in the `ui5.yaml` file don't match the `"sap.app" > "id"` that appears in the `manifest.json` file. The ID in the `manifest.json` file is made up of the namespace.projectname. Take the namespace and project name from the `manifest.json` file and enter these attributes in the `ui5.yaml` file.

- `package.json` file

```
{
  "name": "<<testUI>>",
  "version": "0.0.1",
  "description": "",
  "private": true,
  "devDependencies": {
    "@ui5/cli": "1.4.3",
    "@sap/ui5-builder-webide-extension": "1.0.2"
  },
  "scripts": {
    "build": "ui5 build --include-task=generateManifestBundle
generateCachebusterInfo"
  },
  "ui5": {
    "dependencies": [
      "@sap/ui5-builder-webide-extension"
    ]
  }
}
```

7.1.14.1 Updating Your Project's Grunt Build to the UI5 Build

The following topic provides instructions for updating your project's Grunt build to the UI5 build.

Procedure

1. Delete the `gruntfile.js` file from your project folder.
2. Create a `ui5.yaml` file in your project folder and add the required content for the file:

```
specVersion: '1.0'
metadata:
  name: <<app-name>>
  type: application
  builder:
    customTasks:
      - name: webide-extension-task-updateNeoApp
        afterTask: generateVersionInfo
        configuration:
          destDir: dist
          appFolder: webapp
          nameSpace: <<namespace>>
      - name: webide-extension-task-updateManifestJson
        afterTask: webide-extension-task-updateNeoApp
        configuration:
          appFolder: webapp
          destDir: dist
      - name: webide-extension-task-lint
        afterTask: webide-extension-task-updateManifestJson
        configuration:
          destDir: dist
          appFolder: webapp
          nameSpace: <<namespace>>
      - name: webide-extension-task-resources
        afterTask: webide-extension-task-lint
        configuration:
          nameSpace: <<namespace>>
```

- a. You must replace the `<<app-name>>` and `<<namespace>>` placeholder text with the name of your application and your namespace.

i Note

You must replace periods (.) with slashes (/) when adding the application name or namespace. For example, use `com/sap/wat` instead of `com.sap.wat`.

3. Delete the `package-lock.json` file from your project folder.
4. Update the `package.json` file.
 - a. Remove the Grunt build from the `devDependencies`:

```
"devDependencies": {
  "@sap/grunt-sapui5-bestpractice-build": "1.3.65"
}
```


- b. Add the UI5 devDependencies:

```
"devDependencies": {
  "@ui5/cli": "1.13.0",
  "@sap/ui5-builder-webide-extension": "1.0.x"
},
```

- c. Add the build script.

```
"scripts": {
  "build": "ui5 build --clean-dest --include-task=generateManifestBundle
generateCachebusterInfo"
},
```

- d. Add the UI5 dependencies.

```
"ui5": {
  "dependencies": [
    "@sap/ui5-builder-webide-extension"
  ]
}
```

7.1.14.2 Run a Grunt Build

Instructions for enabling and configuring a Grunt build.

Context



To enable the Grunt build for your project, you first need to add two mandatory configuration files to the root of your project:

- `package.json`
- `Gruntfile.js`

i Note

- The Grunt build currently supports the **ES6** edition of the ECMAScript standard. The **ES6** edition is only supported from version 1.3.65 of the Grunt best-practice build.
- Newer ECMAScript standards, such as ES6, only work with SAPUI5 in browsers with HTML5 capabilities. For more information, see [Browser and Platform Support](#).

Procedure

1. To create each file, in the SAP Web IDE workspace, right-click your project folder, and choose  **New** . Add the required content for each file as follows:

- package.json

```
{
  "name": "grunt-build",
  "version": "0.0.1",
  "description": "Grunt build",
  "private": true,
  "devDependencies": {
    "@sap/grunt-sapui5-bestpractice-build": "1.3.65"
  }
}
```

When you select **Build** from the context menu, these settings instruct npm to install Grunt and the `grunt-sapui5-bestpractice-build` Grunt plugin that contains tasks for building your SAPUI5 project.

Note

The `grunt-sapui5-bestpractice-build` Grunt plugin is published on the SAP npm registry. To run the Grunt build using this plugin outside of SAP Web IDE, such as from a CLI as part of the CI process, add the following configuration option to the npm configuration file:

```
@sap:registry=https://registry.npmjs.org/
```

For more information, see [npmrc: The npm config files](#).

- Gruntfile.js

To include the build tasks listed below, you need to add the following content to the `Gruntfile.js` file, according to the compatible version that you want to define.

This first code content defines the `compatVersion` property for a specific SAPUI5 version using the two-digit format: "xx".

```
module.exports = function (grunt) {
  "use strict";
  grunt.loadNpmTasks("@sap/grunt-sapui5-bestpractice-build");
  grunt.config.merge({ compatVersion: "1.38" });
  grunt.registerTask("default", [
    "clean",
    "lint",
    "build"
  ]);
  grunt.loadNpmTasks("@sap/grunt-sapui5-bestpractice-test");
  grunt.registerTask("unit_and_integration_tests", ["test"]);
  grunt.config.merge({
    coverage_threshold: {
      statements: 0,
      branches: 100,
      functions: 0,
      lines: 0
    }
  });
};
```

For modules created using the HTML5 Application Repository service, enter the following content. Here, the `compatVersion` property is defined as "edge", which is based on the SAP Innovation SAPUI5 version.

```
module.exports = function (grunt) {
  "use strict";
  grunt.loadNpmTasks("@sap/grunt-sapui5-bestpractice-build");
  grunt.config.merge({
```

```

        compatVersion: "edge",
        deploy_mode: "html_repo"
    });
    grunt.registerTask("default", [
        "clean",
        "lint",
        "build"
    ]);
    grunt.loadNpmTasks("@sap/grunt-sapui5-bestpractice-test");
    grunt.registerTask("unit_and_integration_tests", ["test"]);
    grunt.config.merge({
        coverage_threshold: {
            statements: 0,
            branches: 100,
            functions: 0,
            lines: 0
        }
    });
};

```

i Note

The test task for the SAP Web IDE best-practice build can only be triggered from the CLI. Triggering the test task for the build from SAP Web IDE causes the build to fail. For instructions for running automated testing from the CLI, see [Grunt Build Troubleshooting \[page 558\]](#).

These settings instruct Grunt to load the plugin that provides tasks for building SAPUI5 applications and running the tasks in the specified order:

- o `lint`
 - o Validates the project code using ESLint according to the rules defined in the `.eslintrc` configuration file located in the root of your project. It collects any issues in the `dist` folder

i Note


- o If the `.eslintrc` file is not found, the code is validated against default ESLint rules.
- o The default ESLint rules do not align with projects generated from SAP Web IDE templates. If your project was generated from an SAP Web IDE template, we recommend using the generated rules as described below.
- o Central rule definition is not considered for the `lint` task in the Grunt build. For more information, see [Setting JavaScript Rules for All Users](#).
- o If you encounter the following error when trying to build your application, this may indicate that you have an exceptionally big JavaScript file that is causing ESLint to crash with an out of memory exception:


```
FATAL ERROR: Ineffective mark-compacts near heap limit Allocation failed - JavaScript heap out of memory
```

 This error may also affect your design-time static code checks, such as LSP validations. To avoid this problem, add an `eslintignore` configuration file that ignores the problematic file:
 1. Create an `.eslintignore` file in the root of your project.
 2. Copy the path of the file that you want to ignore into the `.eslintignore` file

You can import your own configuration file or generate one in SAP Web IDE as follows:

1. In the workspace, right-click your project root folder, and choose **Project Settings** **>** **Code Checking** **>** **JavaScript**.

2. From the *Validator* dropdown list, select *Basic JavaScript*.
3. Add or modify the code formatting rules according to your requirements and choose *Save*. If you want to display the generated file, you need to first show hidden files from the SAP Web IDE toolbar: .

i Note

You can skip the linting step for your project by removing the `lint` task from the `Gruntfile.js` file.


- `clean`
Cleans the `dist` target folder from the previous build results.
- `build`
Produces a new build output in the `dist` folder of your project that is ready and optimized for better performance in the productive environment. The following tasks are executed during the build:
 - Minification of `.css` files.
 - Minification of JavaScript files (minified files).
 - Copying of the original files to the `dist` folder with the `-dbg` suffix added for debugging purposes.
 - Generation of the `Component-preload.js` and `Component-preload-dbg.js` preload files for the debug and minified files.
 - Minification of the preload file.
 - Generation of the `CachebusterInfo.json` file.
 - Generation of the `changes-bundle.json` file. This file contains a collection of all the changes that are made to an SAP Fiori element application and are located in the `changes` folder.

i Note



The changes are only displayed if you are using SAPUI5 version 1.52 and higher. If you are using an older SAPUI5 version, do not bundle the changes. For a list of tasks to execute when creating a build with an older SAPUI5 version, see [Grunt Build Troubleshooting \[page 558\]](#).

- Generation of the `manifest-bundle.zip` file, which contains the `manifest.json` and `i18n` files.
2. To run the Grunt build, in the workspace, right-click your project and select *Build*.

When the build finishes:

- A new `dist` folder appears in your project folder and is automatically added to `.gitignore`, ensuring that the folder is not checked in the project's Git repository. Any build errors are reported to the console, which you can display from the SAP Web IDE toolbar by selecting .
- A new `package-lock.json` file is added to your project, unless it already exists. It is a special file that npm uses for locking the version of each package in the dependency tree, including the final resolved versions of ranges such as `~1.7.0` or `^1.0.9`.

If your project is connected to Git, commit and push the file to your Git repository.

For more information, see [Semantic Versioning 2.0.0](#)  and [npm](#) .

→ Tip

If the build fails, do one of the following depending on the failure reported in the console:

- **ERROR:** `Unable to locate local Grunt`
Right-click the project folder in the workspace, select *Clean npm Folder*, and build again. This option removes the `node_modules` folder that is automatically created by npm. For more information, see [npm-folders: Folder structures Used by npm](#). After the build finishes, this folder is not displayed in your project, but is stored by the SAP Web IDE build infrastructure.
- **ERROR:** `npm ERR! code EINTEGRITY`
Delete your `package-lock.json` file from the project and run the build again.

⚠ Caution

The `dist` folder and its contents should be treated as read-only. Don't create or change files in this folder – all changes are always overwritten by the next build.

i Note

If you want to build your project outside of SAP Web IDE, for example from CLI or as part of your CI process, and want to apply the best-practice Grunt script that we provide, you should use npm version 5.71 or higher.

Related Information

[Grunt Build in SAP Web IDE](#)

7.1.14.3 Previewing Your Built Application

After building an application using the UI5 builder or the Grunt builder, you can preview your application from the build results.

Procedure

1. In the workspace, in the project folder, locate and open the `dist` folder, then right-click the required HTML file.
2. In the context menu, choose *Run*, then choose the configuration you require.

Related Information

[Application Build \[page 423\]](#)

[Running Applications in Development Mode \[page 391\]](#)

7.1.15 Deploying Applications

You can deploy new applications from SAP Web IDE to different servers.

[Deploy Applications to the SAPUI5 ABAP Repository \[page 414\]](#)

You can deploy an existing application from the SAP Web IDE workspace to the SAPUI5 ABAP repository.

[Update Existing Applications Residing in the SAPUI5 ABAP Repository \[page 417\]](#)

You can update your applications that reside in the SAPUI5 ABAP repository.

[Deploy Applications to SAP BTP \[page 418\]](#)

You can deploy any project from SAP Web IDE to SAP BTP as a new application or as an update to a previously deployed application in any of your SAP BTP accounts.

[Register Applications to SAP Fiori Launchpad \[page 419\]](#)

You can register your SAP BTP deployed application to SAP Fiori launchpad directly from SAP Web IDE. Once registration is complete, a new tile is created in SAP Fiori launchpad, assigned to a site, a catalog, and a group.

[Check the Application Status \[page 422\]](#)

You can check whether your application has been deployed to SAPUI5 ABAP Repository and/or SAP BTP.

[Application Build \[page 423\]](#)

Applications in production environments generally require a build to package resources in an optimal format, for example, to improve loading and delivery.

7.1.15.1 Deploy Applications to the SAPUI5 ABAP Repository

You can deploy an existing application from the SAP Web IDE workspace to the SAPUI5 ABAP repository.

Prerequisites

Make sure that you have complied with all the items described in [Requirements for Connecting to ABAP Systems \[page 60\]](#).

Procedure

1. In your workspace, right-click the desired project.
2. Select ► *Deploy* ► *Deploy to SAPUI5 ABAP Repository* ►.
3. From the *System* dropdown list, select the desired system.

i Note

Validation checks are run and info/warning/error messages are displayed according to the results:

- If the application was created using an SAPUI5 version which is different to that installed in the selected SAP system, a warning is displayed since this may cause issues at runtime.
- If the application was created with a namespace that already exists in an application in the selected SAP system, a warning is displayed and with a recommendation to update the existing application instead of creating another one, as this too can cause issues.

For more information, see [Server Version Check for SAPUI5 Runtime Libraries \[page 416\]](#).

4. Select the *Deploy a new application* radio button and choose *Next*.
5. Provide a name and meaningful description for the application.
6. Choose *Browse*.

i Note

The *Browse* button is enabled only after providing a name for the application.

i Note

If you selected an S/4HANA system, a prefix is added to application name and a package is assigned automatically. The *Browse* button in this case is disabled.

7. In the *Package Selection* search field, enter the name of the desired package and choose *OK*.
8. Choose *Next*.
9. If the selected package is local, choose *Finish*. If it requires transport, select a transport request for your application using one of the following options:
 - Enter a request number.
 - Create a new request and enter its description.

i Note

The request ID is automatically generated.

- Select a request in which you are involved from the table.
10. Choose *Next* and then *Finish*.

You can follow the progress and the completion of the deployment process in the SAP Web IDE console. To open the console, select ► *View* ► *Console* ►.

A notification message displays once the deployment is complete.

Results

When you deploy a project, an application build is performed in the background. The artifact deployed to the SAPUI5 ABAP repository is the result of this application build which represents only the productive version of the application and does not reflect the project's source files.

⚠ Caution

The artifact deployed to the SAPUI5 ABAP repository is the result of this application build which represents only the **productive** version of the application and **does not reflect the project's source files**. It should not be imported back into the workspace for further development. If more development is needed, use source control (such as Git) to maintain your source code.

If the UI5 builder or the Grunt builder is used to build the application, only the content of the build target folder (called `dist`) is deployed. Otherwise, if the application contains a `webapp` folder, only the content within this folder is deployed.

For more information, see [Application Build \[page 423\]](#).

Related Information

[The SAPUI5 ABAP Repository and the ABAP Back-End Infrastructure](#)

7.1.15.1.1 Server Version Check for SAPUI5 Runtime Libraries

Applications with a different SAPUI5 version than that used in the ABAP server may not work properly.

When you deploy an SAPUI5 application project to the SAPUI5 ABAP repository (or open the deployment wizard), the version being used in the application is compared to that of the ABAP server. If the SAPUI5 versions differ, a warning message shows the current versions. If you want to proceed, you can ignore this warning message.

To prevent the warning message:

- Check the JavaScript documentation of the used controls and their methods for `@since` tags. They indicate which version has introduced a new feature which you are going to use.
- Create your application based on the same version of runtime libraries as in your server.
- When testing your application in SAP Web IDE, you can change the SAPUI5 version in the Run Configurations to use the same version of runtime libraries as in your server (for more information, see [Advanced Settings Tab \[page 398\]](#)).
- Always test your application on the server after deployment.

7.1.15.2 Update Existing Applications Residing in the SAPUI5 ABAP Repository

You can update your applications that reside in the SAPUI5 ABAP repository.

Procedure

1. In your workspace, right-click the desired project.
2. Select **Deploy** > **Deploy to SAPUI5 ABAP Repository**.

i Note

If the project was already deployed, the fields will be automatically populated.

3. From the **System** dropdown list, select the desired system.

i Note

Only systems that have the SAPUI5 ABAP Filestore installed are valid for deployment.

4. Select the **Update an existing application** radio button.
5. Choose **Next**.
6. Select the desired application.

i Note

If you receive a message informing you that the system you selected supports only local object creation, implement SAP Note [2046730](#). Otherwise, applications whose packages are not \$TMP will not be updated.

7. If the selected package is local, choose **Next** and then **Finish**. If the package requires transport, choose **Next** to select or create a transport request for your application using one of the following options:
 - Enter a request number.
 - Create a new request and enter its description.

i Note

The request ID is automatically generated.

- Select a request in which you are involved from the table.
8. Choose **Next** and then **Finish**.
 9. A list showing updated files that will be overwritten and new files that have been added is displayed. Choose **OK** to confirm the update.

7.1.15.3 Deploy Applications to SAP BTP

You can deploy any project from SAP Web IDE to SAP BTP as a new application or as an update to a previously deployed application in any of your SAP BTP accounts.

Prerequisites

- You must be working in the cloud edition of SAP Web IDE.

Procedure

1. In the workspace, right-click the project and choose **Deploy > Deploy to SAP BTP**.
2. If you are deploying a new application, your default SAP BTP account, project name, application name, and version are displayed. If necessary, choose [Get Accounts](#) to select a different account (of which you are a member) from the drop down list. You can then edit the application name and version.

i Note

- The [Get Accounts](#) option is not available if you are using a custom identity provider (IdP).
- If you are deploying to an account different to the one on which you run SAP Web IDE, you will be asked to enter your SAP BTP credentials.

The application name must follow these naming conventions:

- The name must start with a letter.
- Do not exceed 30 characters.

i Note

If you exceed this number, only the first 30 characters are visible in the [Deploy](#) dialog box.

- Use only lower-case alphanumeric characters.

i Note

Any upper-case characters will be changed automatically to lower case. Any special characters will be removed automatically.

3. If you want to update a previously deployed application, the application's state, URL, and previous versions are also displayed. If necessary, choose [Get Accounts](#) to select a different account (of which you are a member) from the drop down list. You can then select a different application and edit the version.

i Note

If this application was previously deployed and is still available for update, the dialog will be automatically populated with the application details on SAP BTP.

4. Choose [Deploy](#).

The new application is deployed to SAP BTP. The new version is created and activated (if selected), and if the version is activated, the application is started.

5. In the *Successfully Deployed* dialog box, click the link to preview the latest version of the application on SAP BTP.

i Note

Make sure that the path to the application's executable HTML file is configured properly in the `neo-app.json` file so that it can be previewed on SAP BTP.

If your application does not contain an executable HTML file, the link to the application URL will not be available and the application will not run on SAP BTP.

SAP Fiori applications can be run from SAP Fiori launchpad on SAP BTP. In this case, they don't have to include an executable HTML file. For more information, see [Register Applications to SAP Fiori Launchpad \[page 419\]](#).

After performing a change in the application and deploying it, the change may not be visible when running from the SAP Fiori launchpad until the site administrator clears the HTML5 application cache. For more information, see [Clearing the App Cache](#).

Results

When you deploy a project, an application build is performed in the background. The artifact deployed to SAP BTP is the result of this application build which represents only the productive version of the application and does not reflect the project's source files.

The content from the `dist` folder, which is the build target folder, is used for the deploy process.

For more information, see [Application Build \[page 423\]](#).

7.1.15.4 Register Applications to SAP Fiori Launchpad

You can register your SAP BTP deployed application to SAP Fiori launchpad directly from SAP Web IDE. Once registration is complete, a new tile is created in SAP Fiori launchpad, assigned to a site, a catalog, and a group.

Prerequisites

- You must be working in the cloud edition of SAP Web IDE.
- The application that you want to register must be already deployed to SAP BTP. For more information, see [Deploy Applications to SAP BTP \[page 418\]](#).
- You must be assigned the `TENANT_ADMIN` role in the SAP BTP cockpit. For more information, see [Accessing Services](#).
- You must have at least one site created in advance. For more information, see [Creating a Site Instance in the SAP BTP Cockpit](#).

i Note

You can access the SAP BTP cockpit from SAP Web IDE by selecting **Tools > SAP BTP Cockpit**.

Procedure

1. In the workspace, right-click the desired application and choose **Deploy > Register to SAP Fiori Launchpad**.

i Note

After deploying an application, you can also access the wizard by choosing **Register to SAP Fiori Launchpad** from the **Successfully Deployed** dialog box.

2. Choose **Next**.
3. In the **General Information** step, select the provider account to which you want to register the application.
4. Enter the application name.
5. (Optional) Enter a description and intent, and then choose **Next**.
6. In the **Tile Configuration** step, choose the tile type:
 - **Static** - Enter a title, a subtitle, and choose **Browse** to select an SAPUI5 icon for the tile.
 - **Dynamic** - SAP Web IDE fetches the application's OData service (if the service is already configured in the application's `Configuration.js` or `Component.js` files). If the service is not already configured, enter it manually in the **Service URL** field and then choose **Get Collections**.

i Note

The service path should be relative, for example, `/sap/opu/odata/iwfnd/RMTSAMPLEFLIGHT`.

The service URL is used to get the service's addressable collections. Select the desired addressable collection from the **Collection** dropdown list.

The tile is updated with the `count` property of the selected collection.

i Note

The service must have at least one addressable collection.

The **Number Unit** field is populated with a default value of the entity type of the selected collection. This value can be edited.

The **Refresh Rate (Sec)** field is populated with a default value of 10 seconds. This value can be edited.

i Note

This field accepts only numbers because it expects a number of seconds.

7. Choose **Next**.
8. In the **Assignment** step, depending on the SAP Fiori launchpad that you selected, assign the tile to a **Site**, a **Catalog**, and a **Group**.

- Sites are SAP Fiori launchpad sites.
 - Catalogs are authorization objects used to enable role-based access to apps and groups for a particular launchpad site.
 - Groups are a titled grouping in which apps (represented by tiles) are organized in a launchpad site.
9. Choose *Next*.
 10. Choose *Finish* to confirm and register your application to SAP Fiori launchpad.

Results

After the registration is complete, a success dialog box appears with a link to your application on SAP Fiori launchpad.

i Note

For the application to run, it must be started.

7.1.15.4.1 Connect a Project to the SAP BTP Git Repository

When you deploy the application to SAP BTP, the source code is not automatically pushed to the SAP BTP Git. You can connect your project to any Git repository and push your changes there later.

Prerequisites

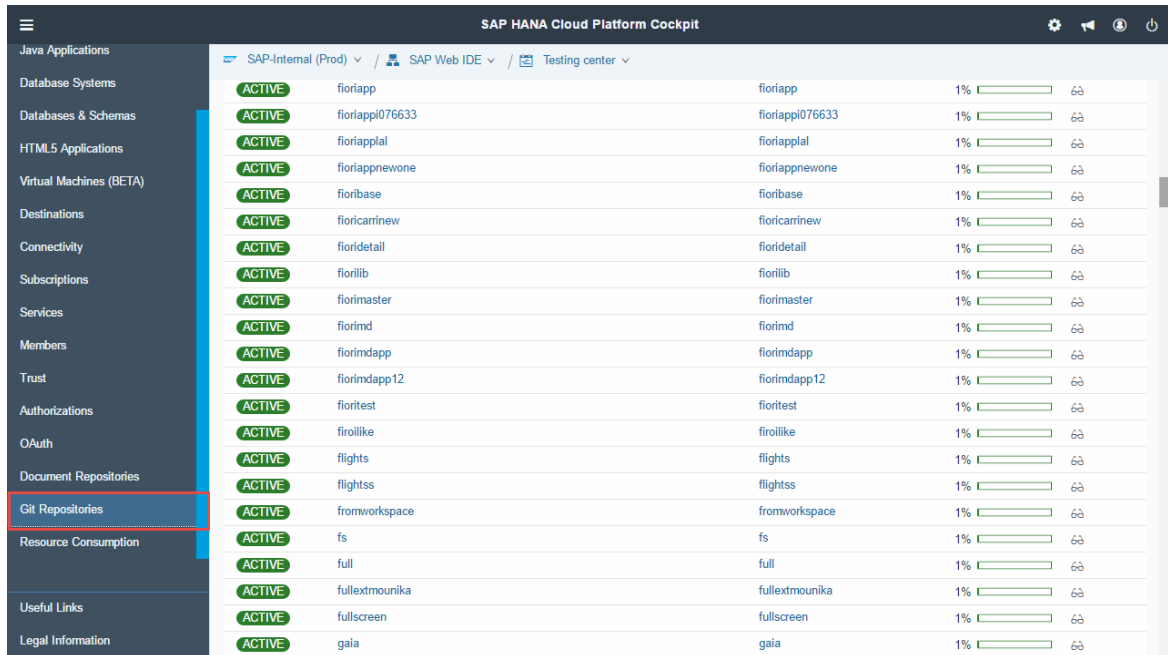
You must be an account administrator.

Context

The application source code should be managed in Git.

Procedure

1. Go to **Tools > SAP BTP Cockpit > Git Repositories**.



Application Name	Status	Progress	Link
fioriapp	ACTIVE	1%	🔗
fioriappi076633	ACTIVE	1%	🔗
fioriapplal	ACTIVE	1%	🔗
fioriappnewone	ACTIVE	1%	🔗
fioribase	ACTIVE	1%	🔗
fioricarrinew	ACTIVE	1%	🔗
fioridetail	ACTIVE	1%	🔗
fiorilib	ACTIVE	1%	🔗
fiorimaster	ACTIVE	1%	🔗
fiorimd	ACTIVE	1%	🔗
fiorimdapp	ACTIVE	1%	🔗
fiorimdapp12	ACTIVE	1%	🔗
fioritest	ACTIVE	1%	🔗
fiorilike	ACTIVE	1%	🔗
flights	ACTIVE	1%	🔗
flightss	ACTIVE	1%	🔗
fromworkspace	ACTIVE	1%	🔗
fs	ACTIVE	1%	🔗
full	ACTIVE	1%	🔗
fullextmounika	ACTIVE	1%	🔗
fullscreen	ACTIVE	1%	🔗
gaia	ACTIVE	1%	🔗

2. Create a new Git repository, select it and locate the created Git URL you want to use. For more information, see [Creating a Repository](#).
3. Initialize your local repository and connect it to the remote GIT repository by using the URL you obtained from the cockpit. For more information, see [Initialize a Local Git Repository \[page 372\]](#).
4. Use the Git operations to fetch, commit, and push your changes. For more information, see [Using Source Control \(Git\) \[page 363\]](#).

7.1.15.5 Check the Application Status

You can check whether your application has been deployed to SAPUI5 ABAP Repository and/or SAP BTP.

Procedure

- In your SAP Web IDE workspace, right-click the desired application and choose **Deploy** > **Application Status**. One of the following occurs, depending on the application status:
 - If the application is deployed to SAPUI5 ABAP Repository, its system, name, package, and URL are displayed.
 - If the application is deployed to SAP BTP, its account, name, state, URL, link to the app page on SAP BTP, and versions are displayed.

Note

If your application does not contain an `index.html` file, the link to the application URL will not be available.

→ Tip

If the application is not deployed, you can choose [Deploy](#) to deploy it.

7.1.15.6 Application Build

Applications in production environments generally require a build to package resources in an optimal format, for example, to improve loading and delivery.

If you include the `ui5.yaml` file and the `package.json` file, with the required dependencies and `devDependencies` for UI5, the UI5 build is created.

If you include the `Gruntfile.js` and `package.json` files in your project, the Grunt build is activated when you deploy your application to either SAP BTP or the SAPUI5 ABAP repository. The results of the build (the `dist` folder content) are deployed.

If the mandatory files do not exist in your project, the build step is skipped and the original project sources are deployed.

Related Information

[Building Applications \[page 405\]](#)

7.1.16 Extending SAPUI5 Applications

You can extend SAPUI5 applications residing remotely on the SAPUI5 ABAP repository or SAP BTP.

i Note

When extending an application, compatibility issues may arise between the original and the extended application. For more information, see [Caveats Regarding Stability Across Application Upgrades](#).

[Extend Applications that Reside in the SAPUI5 ABAP Repository \[page 424\]](#)

You can extend an existing SAP Fiori application that resides in the SAPUI5 ABAP repository without importing it to SAP Web IDE.

[Extend Applications that Reside on SAP BTP \[page 425\]](#)

You can extend an existing SAP Fiori application that resides on SAP BTP without importing it to SAP Web IDE.

[Create New Extensions \[page 426\]](#)

Extensions enable you to change the views or the logic of an extended project.

7.1.16.1 Extend Applications that Reside in the SAPUI5 ABAP Repository

You can extend an existing SAP Fiori application that resides in the SAPUI5 ABAP repository without importing it to SAP Web IDE.

Prerequisites

- Activate the `/sap/bc/adt` and the `/sap/bc/ui2/app_index/` services in your back end.
- Make sure you have configured the connectivity to your ABAP system as described in the [Connect to ABAP Systems \[page 57\]](#) topic.
- Make sure you comply with the [Requirements for Connecting to ABAP Systems \[page 60\]](#).

Context

To extend a SAPUI5 application, you need to create an extension project to which you add new extensions as described in the [Create New Extensions \[page 426\]](#) topic.

It is recommended to use source control (such as Git) to maintain your source code. Once you deploy the extension project to the ABAP system, only the source code can be extended.

i Note

The SAPUI5 ABAP repository is technically based on the BSP repository of the ABAP Server. The BSP repository is used only as a repository or storage for SAPUI5 application files. However, the BSP server-side processing is not used at runtime and therefore the flow logic of ABAP parts cannot be used, since they are not executed at runtime.

Procedure

1. From the *File* menu, choose **► New ► Extension Project**.
2. Choose **► Select Application ► SAPUI5 ABAP Repository**. The *Select SAPUI5 ABAP Repository Application* dialog box is displayed.
3. Select the desired remote system.
4. Search for the application that you want to extend.
5. Select the application and choose *OK*. The *Extension Project Name* field is automatically populated in the wizard. If necessary, you can edit this name.

i Note

The name entered for the extension project together with its namespace is later used as the component name when deploying this extension project to the SAPUI5 ABAP repository, therefore it should be unique in the ABAP system.

6. If necessary, select the *Import original application* checkbox. Use this imported application as reference only. Do not modify it.
7. If necessary, select the *Open extension project in extensibility pane* checkbox to automatically open the extensibility pane after the project is generated.
8. Select the SAPUI5 version you want to use when running the extended application in SAP Web IDE.

i Note

If no specific version is selected, the version of the selected SAP system will be used.

The version must be equal to or lower than the SAPUI5 version of the selected SAP system. Selecting a higher version may cause errors when running the application from the SAP system and/or SAP Fiori launchpad.

If the SAPUI5 version of the selected SAP system is lower than 1.28, the version of the extended application will be 1.28.4. For more information, see [Server Version Check for SAPUI5 Runtime Libraries \[page 416\]](#).

9. Choose *Next*.
10. Choose *Finish* to confirm and create your extension project. The new extension project is added to the workspace.

Related Information

[The SAPUI5 ABAP Repository and the ABAP Back-End Infrastructure Create New Extensions \[page 426\]](#)
[Deploy Applications to the SAPUI5 ABAP Repository \[page 414\]](#)

7.1.16.2 Extend Applications that Reside on SAP BTP

You can extend an existing SAP Fiori application that resides on SAP BTP without importing it to SAP Web IDE.

Context

i Note

This feature is unavailable in SAP Web IDE personal edition.

Procedure

1. From the *File* menu, choose **New** > *Extension Project*.
2. Choose **Select Application** > **SAP BTP**. The **Select Application from SAP BTP** dialog box is displayed.
3. Enter your SAP BTP account, user name, and password.
4. Choose *Get Applications*.
5. Search for the application that you want to extend.
6. Select the desired application and choose *OK*. The *Extension Project Name* field is automatically populated in the wizard. If necessary, you can edit this name.
7. If desired, select the *Open extension project in extensibility pane* checkbox to automatically open the extensibility pane after the project is generated.
8. Choose *Next*.
9. Choose *Finish* to confirm and create your extension project. The new extension project is added to the workspace.

7.1.16.3 Create New Extensions

Extensions enable you to change the views or the logic of an extended project.

Prerequisites

There must already be an extension project in your workspace.

Context

You can create extensions to:

- Replace an existing view with a new view in an existing project.
- Add logic to an existing view using an extension point that is defined in the original project.
- Change control visibility.
- Extend an existing controller with new logic.
- Implement a UI controller hook with new logic.
- Customize the strings of the original application.
- Replace the OData service of the original application.

Once the extensions have been created, a reference to them is created in the `Component.js/manifest.json` file of the extended project.

i Note

If you delete or rename a file that is referenced from the `component.js/manifest.json` file, the application does not work properly. Make sure that you delete the reference or update the file name on the `component.js/manifest.json` file as well.

i Note

If you hide a control and then want to show it again, you must delete the extension from the `component.js/manifest.json` customizing block. Changing the `visible` property from `False` to `True` does not make the control reappear.

[Extend UI Elements Using the Extensibility Pane \[page 427\]](#)

You can extend an existing SAP Fiori extension project from the extensibility pane, as well as by using the Extension wizard.

[Extend Controllers \[page 431\]](#)

You can extend a controller of the original application by replacing it with an empty controller or with a copy of the original controller. You can also implement UI controller hooks if they are provided by the original application. Once one of these controllers is in place, you can customize it as needed.

[Extend Views \[page 432\]](#)

You can extend a view using an extension point.

[Hide Controls \[page 433\]](#)

You can hide a specific control in the original application.

[Edit Strings \[page 434\]](#)

The `i18n` Resource Text Customization extension allows you to copy the `i18n` folder of the original application to your extended application. This allows you to edit the UI strings in the extended application without altering the original application.

[Replace OData Services \[page 435\]](#)

You can replace the extended application's OData service with a new OData service.

[Replace Views \[page 437\]](#)

You can replace a specific view in an original application with a new view.

7.1.16.3.1 Extend UI Elements Using the Extensibility Pane

You can extend an existing SAP Fiori extension project from the extensibility pane, as well as by using the Extension wizard.

Context

The extensibility pane shows an extension project in preview mode.

You can choose the *Extensibility Mode* option from the toolbar. This enables a two-way selection of elements from the application and displays the available extension points.

i Note

The extensibility mode disables the ability to preview the application's functionality.

The *Outline* section shows the UI elements available in the application, as well as extension points and UI

controller hooks (both identifiable by the  icon).

You can filter the UI elements displayed in the outline as follows:

- *All Elements* - Shows all UI elements contained in the application's `view.xml` file. Elements without a configured ID and/or aggregations appear grayed out because they cannot be extended.

i Note

- You cannot extend elements that do not have configured IDs. In addition, you cannot hide elements that do not have a `visible` property.
- Elements that are set as `visible` manually by the view's controller will not be hidden.
- You can try and hide these controls by replacing the view or extending the controller that hides the control and override the method.

- *Extended Elements* - Shows all UI elements that are already extended.
- *Extensible Elements* - Shows all UI elements that have a configured ID.
- *Extension Points* - Shows all extension points that are available in the application (identifiable by the



icon).

You can select a UI element's ID in the *Outline* pane and see it highlighted in the previewed application. Likewise, if you hover over a UI element in the previewed application, the respective element's ID is highlighted in the extensibility pane.

When you click a UI element in the previewed application when in extensibility mode, the element is selected and highlighted in the *Outline* pane. You can deselect it by clicking it again.

i Note

This two-way selection is available only if *Extensibility Mode* is selected from the menu bar.

i Note

Only UI elements that are part of the application's XML views appear in the *Outline* pane.

The extensibility pane provides information regarding UI elements (for example, where the element is located in the application, which view holds this element, its ID, and so on), in a visual manner.

When you extend an SAP Fiori app from the SAPUI5 ABAP repository, information about extension points and UI controller hooks is displayed as a tooltip if you hover over these controls in the *Outline* pane.

From the extensibility pane, you can also view the original application's views and controllers in read-only mode.

i Note

You can view the original application's code even if it does not reside in the workspace.

You can also use mock data for the extensibility pane by selecting the desired extension project, and from the *Tools* menu, choosing *Extensibility Pane with Mock Data*.

i Note

If an extended application requires URL parameters and/or a hash fragment, these can be defined as a run configuration in the extension project.

The extensibility pane uses the first run configuration of type `Web Application` defined in the extension project.

For more information, see [Create Run Configurations \[page 394\]](#).

Procedure

1. In SAP Web IDE, select the desired extension project.
2. From the *Tools* menu, choose *Extensibility Pane*.
3. Select the UI element that you want to extend.
4. From the *Extend* button, select the extension that you want to add, or right-click and select the extension.
The extension is added to the project and the extended UI element is marked with its extension in parentheses.
5. Refresh the application to make the change visible.
6. Select the extended UI element and choose **► Open ► Extension Code ▾** to go directly to the relevant file that represents the extension.
7. You can remove an extension by right-clicking an extended UI element and selecting *Remove Extension* or by selecting the extended element and choosing *Remove Extension*.
This functionality is unavailable for the *Extend Controller* and the *Replace View* extensions.
8. You can view the relevant code from the original application by selecting an element in the *Outline* pane (for example view, controller, fragment, UI control) and choosing **► Open ► Original Code ▾**. This opens the relevant file in the SAP Web IDE editor in read-only mode.

i Note

If the original code resides in your workspace, the file opens ready for editing.

9. For replaced views and extended extension points, you can access the layout editor from the extensibility pane by choosing **► Open ► Layout Editor ▾**.

Task overview: [Create New Extensions \[page 426\]](#)

Related Information

[Extend Controllers \[page 431\]](#)

[Extend Views \[page 432\]](#)

[Hide Controls \[page 433\]](#)

[Edit Strings \[page 434\]](#)

[Replace OData Services \[page 435\]](#)

[Replace Views \[page 437\]](#)

7.1.16.3.1.1 Extensibility Pane Troubleshooting

Steps you can take if you have trouble using the Extensibility pane.

The following lists error messages you may receive when using the Extensibility pane, possible causes of the error, and possible solutions.

Error in the Extensibility Pane

- The first time you open the extensibility pane you might get the following error:

An error has occurred. Click OK to refresh the application.

Solution:

Click *OK* to refresh the application.

- When in Extensibility Mode, if you select a UI element in the application and it does not appear in the outline (for example when showing only **Extended Elements**), you might get the following error:

Cannot find the UI element in the outline. Make sure you are using the “Extensible Elements” or “All Elements” outlines.

Solution:

Make sure you are using the **Extensible Elements** or the **All Elements** outlines.

Unexpected Behavior when Running an Extended Application in the Extensibility Pane

When running an extended application within the extensibility pane you might encounter unexpected behavior, for example, no data is displayed.

Solution:

When extending an application which requires URL parameters and/or a hash fragment, these should be defined in the run configuration of the extension project.

7.1.16.3.2 Extend Controllers

You can extend a controller of the original application by replacing it with an empty controller or with a copy of the original controller. You can also implement UI controller hooks if they are provided by the original application. Once one of these controllers is in place, you can customize it as needed.

Task overview: [Create New Extensions \[page 426\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 427\]](#)

[Extend Views \[page 432\]](#)

[Hide Controls \[page 433\]](#)

[Edit Strings \[page 434\]](#)

[Replace OData Services \[page 435\]](#)

[Replace Views \[page 437\]](#)

Replacing Controllers

Procedure

1. Select the extension project to which you want to add the extension.
2. From the *File* menu, choose ► *New* ► *Extension* ▾.
3. Make sure that the desired extension project is selected and choose *Next*.
4. Select *Extend Controller* and then choose *Next*.
5. Select the controller that you want to extend.
6. From the *Replace with* dropdown list, select *Copy of the original controller* to edit the controller based on the original controller, or select *Empty Controller* to replace the controller with an entirely new one and choose *Next*.
7. Choose *Finish* to add the extension to the selected extension project.

i Note

The new controller extends the controller that is provided by SAP. Methods of the custom controller override standard methods with the same name (except for the controller lifecycle methods that are called in addition to the original controller method implementations). When overriding a controller method, any functionality that was previously provided by the SAP controller in this method is no longer available. Likewise, any future changes made to the SAP controller method implementation will not be reflected in the custom controller.

Implementing UI Controller Hooks

Procedure

1. Select the extension project to which you want to add the extension.
2. Choose **File** > **New Extension**.
3. Select the location of the extension project to which you want to add the extension and choose **Next**.
4. Select **Implement UI Controller Hook** and choose **Next**.
5. Select the controller and the UI controller hook that you want to implement and choose **Next**.
6. Choose **Finish** to add the extension to the selected extension project.

7.1.16.3.3 Extend Views

You can extend a view using an extension point.

Prerequisites

You must have defined extension points in the original application.

Procedure

1. Select the extension project to which you want to add the extension.
2. From the **File** menu, choose **New** > **Extension**.
3. Select the extension project to which you want to add the extension and choose **Next**.
4. Select **Extend View/Fragment** and choose **Next**.
5. Select the view or fragment that you want to extend.
6. Select the desired extension point.

Note

Not all views or fragments have extension points.

7. Choose **Next**.
8. Choose **Finish** to add the extension to the selected extension project.

Task overview: [Create New Extensions \[page 426\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 427\]](#)

[Extend Controllers \[page 431\]](#)

[Hide Controls \[page 433\]](#)

[Edit Strings \[page 434\]](#)

[Replace OData Services \[page 435\]](#)

[Replace Views \[page 437\]](#)

7.1.16.3.4 Hide Controls

You can hide a specific control in the original application.

Context

i Note

You can only hide controls that have their `visible` property defined as `true`. If the `visible` property does not exist, you cannot hide the control.

Controls that are configured in a fragment that is loaded dynamically might still appear in the UI. Elements that are set as *visible* manually by the view's controller will not be hidden.

You can try and hide these controls by replacing the view or extending the controller that hides it and override the method.

Procedure

1. Select the extension project to which you want to add the extension.
2. From the *File* menu, choose **► New ► Extension ▾**.
3. Select the extension project to which you want to add the extension and choose *Next*.
4. Select *Hide Control* and choose *Next*.
5. Select the view or fragment containing the control that you want to hide.
6. Select the specific control that you want to hide.

i Note

Only controls with an ID that is defined in the original application appear in the list.

7. Choose *Next*.
8. Choose *Finish* to add the extension to the selected extension project.

Task overview: [Create New Extensions \[page 426\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 427\]](#)

[Extend Controllers \[page 431\]](#)

[Extend Views \[page 432\]](#)

[Edit Strings \[page 434\]](#)

[Replace OData Services \[page 435\]](#)

[Replace Views \[page 437\]](#)

7.1.16.3.5 Edit Strings

The `i18n` Resource Text Customization extension allows you to copy the `i18n` folder of the original application to your extended application. This allows you to edit the UI strings in the extended application without altering the original application.

Prerequisites

The original application must have an `i18n` folder with at least one `Properties` file that contains the relevant strings.

Procedure

1. Select the extension project to which you want to add the extension.
2. From the *File* menu, choose **► New > Extension**.
3. Select the extended application that you want to customize.
4. Choose *Next*.
5. Select the *i18n Resource Text Customization* tile.
6. Choose *Next*.
7. Choose *Finish* to confirm and add the extension.

Results

The `i18n` folder of the original application is copied to your extended application, including all its `.properties` files. You can change one or more of the strings in the `.properties` file and run your extended application to see them in runtime. The original application remains unchanged.

Task overview: [Create New Extensions \[page 426\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 427\]](#)

[Extend Controllers \[page 431\]](#)

[Extend Views \[page 432\]](#)

[Hide Controls \[page 433\]](#)

[Replace OData Services \[page 435\]](#)

[Replace Views \[page 437\]](#)

7.1.16.3.6 Replace OData Services

You can replace the extended application's OData service with a new OData service.

Prerequisites

The new OData service must be compatible (similar metadata, similar operations, as well as any extensions) with the current OData service of the original application project.

Procedure

1. Select the extension project to which you want to add the extension.
2. From the *File* menu, choose **► New > Extension ▾**.
3. Select the extended application that you want to customize.
4. Choose *Next*.
5. Select the *Replace Service* tile.
6. Choose *Next*.
7. Select the new OData service in one of the following ways:
 - Choose *Service Catalog* and select the desired data source from the list. Once you select the desired data source, choose a service and then choose *Select*.

- Choose *Workspace* and browse for the relevant metadata in the SAP Web IDE system.
- Choose *File System* and browse for the relevant metadata in your file system.
- Choose *Service URL* and select the desired data source from the list. Then paste the relevant URL in the field beneath the data source.

i Note

If the system belongs to an API Management service, you are required to enter an application key in the relevant field.

After the data source is selected, the service details are displayed.

i Note

If you select an OData service, a `model` folder containing the `metadata.xml` file is automatically created during the project generation.

i Note

If the data source is selected from a local file (using Browse), the created application project can be run only using Mock Data, unless the service URL is manually added to the generated application code. For more information, see [Running Applications in Development Mode \[page 391\]](#).

8. Choose *Next*.
9. Choose *Finish* to confirm and replace the OData service.

Results

You can run your extended application to see data from the new OData service.

Task overview: [Create New Extensions \[page 426\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 427\]](#)

[Extend Controllers \[page 431\]](#)

[Extend Views \[page 432\]](#)

[Hide Controls \[page 433\]](#)

[Edit Strings \[page 434\]](#)

[Replace Views \[page 437\]](#)

7.1.16.3.7 Replace Views

You can replace a specific view in an original application with a new view.

Context

- The new view can be edited using the layout editor.
- The new view replaces the view provided by SAP. Any future changes made to the SAP view will not be reflected in the new view. Furthermore, if the new view is an empty view, any functionality that was previously provided by the SAP view will not be available.

Procedure

1. Select the extension project to which you want to add the extension.
2. From the *File* menu, choose **► New ► Extension ▾**.
3. Select the extension project to which you want to add the extension.
4. Click *Next*.
5. Select *Replace View*.
6. Click *Next*.
7. Select the view that you want to replace.
8. From the *Replace with* dropdown list, select *Copy of the original view* to edit the view based on the original view, or select *Empty View* to replace the view with an entirely new one.
9. Click *Next*.
10. Click *Finish* to add the extension to the selected extension project.

Task overview: [Create New Extensions \[page 426\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 427\]](#)

[Extend Controllers \[page 431\]](#)

[Extend Views \[page 432\]](#)

[Hide Controls \[page 433\]](#)

[Edit Strings \[page 434\]](#)

[Replace OData Services \[page 435\]](#)

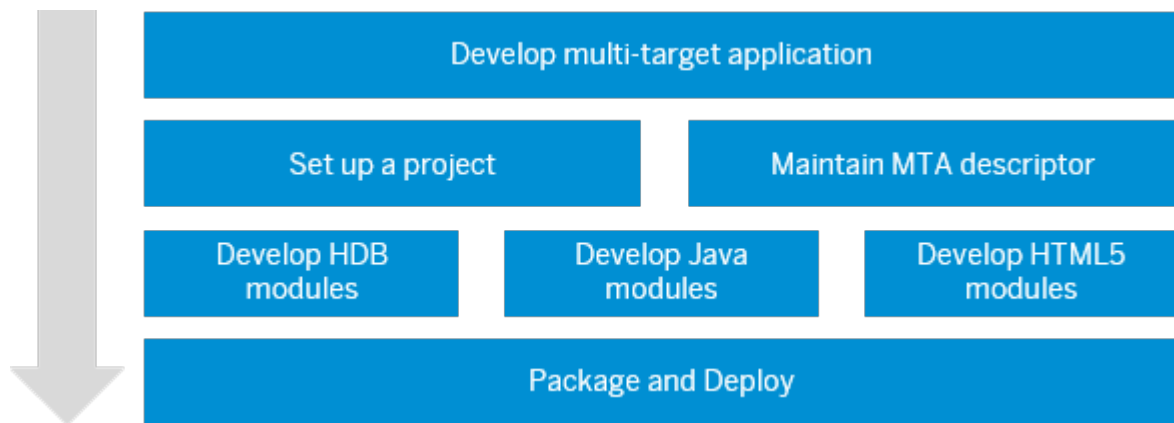
7.2 Developing Multitarget Applications

Multitarget, or multi-tier applications are composed of multiple software modules representing the data, business logic, and UI tiers. These modules are created with different technologies and are deployed to different target platforms, yet share the same development lifecycle.

i Note

You can only deploy multitarget applications to the SAP BTP, Cloud Foundry environment.

The multitarget application concept aims at orchestrating the deployment of all these modules so that all runtime dependencies are properly resolved and the application functions as expected. This is achieved by supplying to the deployment tools a set of descriptors that define the interdependencies and deployment scenarios for all modules in the application.



- [Developing Multitarget Applications \[page 438\]](#)
- [Setting Up Application Projects \[page 449\]](#)
- [Inside an MTA Descriptor \[page 440\]](#)
- [Developing SAP HANA Database \(HDB\) Modules \[page 457\]](#)
- [Developing Java Modules \[page 478\]](#)
- [Developing HTML5 Modules \[page 490\]](#)
- [Packaging and Deploying Applications to Production Systems \[page 506\]](#)

Terms and Concepts

Term	Description
Multitarget application (MTA)	An application comprised of multiple software modules, which are created with different technologies and deployed to different target platforms, yet share the same lifecycle. In the context of this guide, an "application" is an MTA.

Term	Description
Target platform	A platform to which a module is deployed, such as an SAP HANA database.
MTA descriptor	A YAML file named <code>mta.yaml</code> that contains a list of all entities, such as modules, resources, and properties that belong to an application or are used by it at runtime, and the dependencies between them. It is automatically generated when an MTA project is created or modified, or when a module is added or removed. The developer needs to edit the descriptor manually to define resources, properties, and dependencies, as well as fill in missing information.
Module	A self-contained application of a certain type, which is developed, packaged, and deployed to a target platform as part of an MTA.
Module type	A type that defines the structure, development technology, and target platform of a module. SAP Web IDE supports the following module types, representing the three application tiers: <ul style="list-style-type: none"> • SAP HANA database (HDB) module - represents the data tier • Java and Node.js modules - represent the business logic tier • HTML5 module - represents the UI tier
<div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>i Note</p> <p>In the current version, Node.js modules are not supported.</p> </div>	
Resource	Any resource, such as an external service, property, or environment variable, that is required by a module at runtime but not provided by the module itself.
Property	A property (key-value pair) of an application, module, or resource, that is used during deployment or at runtime.
Parameter	A reserved variable belonging to a module or resource, whose value is used during deployment or at runtime. Parameters can be read-only, write-only, or read-write. The values of writable parameters can be specified in the descriptor.
Dependency	A relationship between a module and another module, resource, or property, such as <i>provides</i> and <i>requires</i> . <ul style="list-style-type: none"> • <i>provides</i>: indicates the properties or parameters that are provided by a module or resource to other modules. • <i>requires</i>: indicates other modules or resources that are required by a module in order to run.
Deployment archive	An archive similar to JAR into which all the application artifacts are packaged for deployment.

[Inside an MTA Descriptor \[page 440\]](#)

The multi-target application (MTA) descriptor contains the metadata of all entities comprising an application or used by it during deployment or runtime, and the dependencies between them.

[Setting Up Application Projects \[page 449\]](#)

You can set up a multi-target application (MTA) project in various ways.

[Using the HTML5 Application Repository in a Multitarget Application \[page 453\]](#)

Developers can create a Multitarget Application that uses the HTML5 Application Repository service, and benefit from enhanced capabilities in their project.

[Developing SAP HANA Database \(HDB\) Modules \[page 457\]](#)

An SAP HANA database (HDB) module is a collection of related design-time database artifacts, such as data models, views, or procedures.

[Developing Node.js Modules \[page 467\]](#)

A Node.js module is a collection of related JavaScript files and service definitions that implement the business logic of your application.

[Developing Java Modules \[page 478\]](#)

A Java module is a collection of related Java files and service definitions. Java modules implement the business logic of your application, either instead of or in addition to Node.js modules. A Java module can be either a Java Web Archive (WAR) or Java Archive (JAR) built with Apache Maven.

[Developing HTML5 Modules \[page 490\]](#)

An HTML5 module is a collection of related HTML5 files that implement the user interface of your application.

[Developing SAP S/4HANA Service Extensions \[page 503\]](#)

SAP Web IDE enables you to create S/4 HANA extensions. You implement an application with an OData service that extends an existing S/4HANA service and exposes additional data from a different source, such as an SAP HANA database.

[Developing Business Applications \[page 504\]](#)

SAP Web IDE enables you to develop business applications using the application programming model for SAP BTP.

[Adding a Service Dependency to your MTA Project \[page 505\]](#)

The Business Application Development tools for SAP BTP extension allows you to add a service dependency to your MTA project.

[Packaging and Deploying Applications to Production Systems \[page 506\]](#)

At the last stage of multi-target application (MTA) development, you need to package your application and deploy it to a target production system. You can only deploy multitarget applications to the SAP BTP, Cloud Foundry environment.

7.2.1 Inside an MTA Descriptor

The multi-target application (MTA) descriptor contains the metadata of all entities comprising an application or used by it during deployment or runtime, and the dependencies between them.

The MTA descriptor (the `mta.yaml` file located in the root project folder) is automatically generated when an application project is created from scratch, and it is updated when the project properties change or when a module is added or removed. However, not all the necessary information can be generated automatically. You need to maintain the descriptor manually to define resources, properties, and dependencies, as well as fill in missing information.

Related Information

[MTA Editor \[page 441\]](#)

[MTA Descriptor Model \[page 441\]](#)

[MTA Descriptor Example \[page 442\]](#)

[MTA Descriptor Elements \[page 443\]](#)

[Resource Manager \[page 445\]](#)

7.2.1.1 MTA Editor

The MTA descriptor is written in the YAML format, which has strict syntax requirements. You can edit the descriptor in the text-based code editor, but we recommend you use the visual MTA editor because it provides input validation.

To set the MTA editor as the default for the MTA descriptor, go to **Preferences > Default Editors** and set the editor for the *MTA Application Descriptor*.

Note

The MTA visual editor removes comments and formats the file. If you want to add comments, use the code editor. To open the code editor, either make the code editor the default editor or right-click the file and choose **Open With > Code Editor**.

If you edit the file with the code editor, it is important to use spaces rather than tabs for indentation.

Related Information

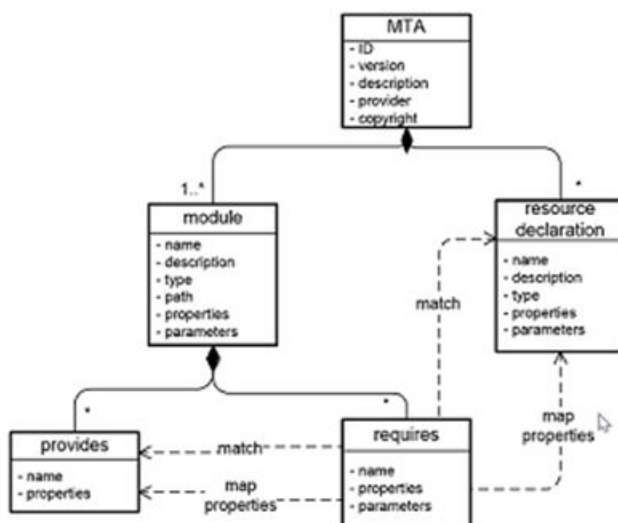
[MTA Descriptor Model \[page 441\]](#)

[MTA Descriptor Example \[page 442\]](#)

[MTA Descriptor Elements \[page 443\]](#)

7.2.1.2 MTA Descriptor Model

The following figure illustrates the MTA descriptor model.



Related Information

[MTA Editor \[page 441\]](#)

[MTA Descriptor Example \[page 442\]](#)

[MTA Descriptor Elements \[page 443\]](#)

7.2.1.3 MTA Descriptor Example

An example of an MTA descriptor is provided here.

```
ID: com.sap.node.hello.world.db
version: 1.0.0
description: A Hello World sample application
provider: SAP Sample generator
copyright: 2016 SAP SE
modules:
  - name: node-hello-world-db
    type: hdb
    path: db
    requires:
      - name: hdi-container
    provides:
      - name: node-hello-world-db
  - name: node-hello-world
    type: html5
    path: web
    requires:
      - name: uaa # dependency on the UAA service
      - name: backend_api # name of the 'provides' element in the
noje.js module # name of a target environment variable
  group: destinations # name of a target environment variable
  properties:
    name: nodejs # key values in json format in the
'destinations' variable
    url: ~{url}
    forwardAuthToken: true
  - name: node-hello-world-backend
    type: nodejs
    path: js
    requires:
      - name: node-hello-world-db
      - name: hdi-container
      - name: uaa
    provides:
      - name: backend_api
      properties:
        url: ${default-url}
# Resources describe required services
resources:
  - name: hdi-container
    type: com.sap.xs.hdi-container
  - name: uaa
    type: com.sap.xs.uaa
```

Related Information

[MTA Editor \[page 441\]](#)

[MTA Descriptor Model \[page 441\]](#)

[MTA Descriptor Elements \[page 443\]](#)

7.2.1.4 MTA Descriptor Elements

The following table provides an overview of the MTA descriptor elements and their possible values.

Sections	Element	Description	Possible Values
General	<code>id</code>	The unique application ID.	Any number of unicode characters in a reverse-URL dot-notation, for example <code>com.sap.mta.sample</code> . The ID should be unique in the target runtime environment.
General	<code>version</code>	The application version.	Should follow the semantic versioning standard format. For more information, see https://semver.org/ .
General	<code>description</code>	(Optional) A description of the application.	
General	<code>provider</code>	(Optional) The application vendor/provider name.	
General	<code>copyright</code>	(Optional) The copyright notice of the provider.	
modules	<code>name</code>	The module name.	Should be unique within the descriptor.
modules	<code>path</code>	The relative path to a module from the application root.	For example, <code>./backend</code>
modules	<code>type</code>	The module type, which defines the design-time tools and builders required for the module.	One of the following values: <code>hdb</code> , <code>nodejs</code> , <code>html5</code> .
modules	<code>requires</code>	A subsection of a module section that contains the names of resources, other modules, and/or properties provided by other modules, which are required by the current module to run.	For example: <pre>requires: - name: backend properties: name: external url: ~{url}</pre>

Sections	Element	Description	Possible Values
modules	provides	A subsection of a module section that specifies the properties provided by the current module or resource.	For example: <pre>provides: - name: price_opt properties: protocol: http uri: host.domain</pre>
modules, resources	properties	(Optional) A flat or hierarchical collection of properties, provided or required by a module or resource.	For example: <pre>properties: - name: node-hello-world-db - name: hdi-container - name: uaa</pre>
modules, resources	parameters	(Optional) Reserved variables that can contain read-only, read-write, or write-only values, which can be used by deployment tools, but not at runtime. Parameters can be referenced with placeholders enclosed in <code>\${</code> and <code>}</code> , which are resolved during deployment or runtime.	For example: <pre>parameters: domain: price.sap.com</pre>
modules	group	(Optional) A named group of properties from different providers used as a single lookup object.	For example: <pre>group: DESTINATIONS</pre>
modules	description	(Optional) A description of the module, which does not appear in the application UI.	

Related Information

[MTA Editor \[page 441\]](#)

[MTA Descriptor Model \[page 441\]](#)

[MTA Descriptor Example \[page 442\]](#)

7.2.1.5 MTA Service Provisioning

A service instance required by the MTA application is created automatically for specific resource types.

For example, when you add an HDB module to your MTA project, a resource is added to the `mta.yaml` file.

When you build the module, the required hdi-container instance is created automatically in the selected space.

```
ID: mta_db
_schema-version: '2.1'
```

```


version: 0.0.1
modules:
  - name: db
    type: hdb
    path: db
    requires:
      - name: hdi_db
resources:
  - name: hdi_db
    properties:
      hdi-container-name: ${service-name}
    type: com.sap.xs.hdi-container

```

Service instances are created automatically for the following resource types (defined in the `mta.yaml` file):

- `com.sap.xs.hdi-container`
- `com.sap.xs.uaa`
- `org.cloudfoundry.managed-service`
- `com.sap.xs.dwf-edw-client`

Note

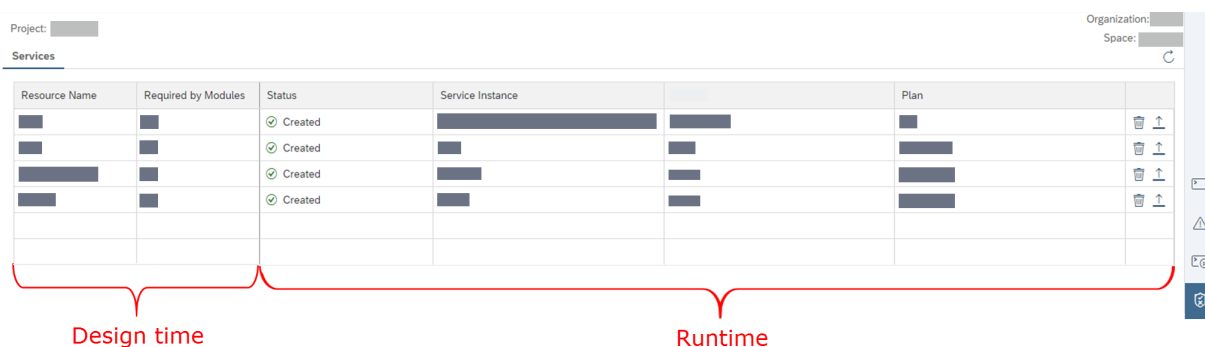
Only the initial service instance is created automatically. The service instance is not updated automatically if the resource parameters change. To update the service instance, choose  in the Resource Manager. For more information, see [Resource Manager \[page 445\]](#).

7.2.1.6 Resource Manager

The *Resource Manager* tool lets you view information about all of the resources (service instances) required by the modules based on the `mta.yaml` file in the multitarget application project.


To open the tool, from the right sidebar, choose .



The table of service instances for the project is divided into design time and runtime.



Resource Name	Required by Modules	Status	Service Instance	Plan
[Redacted]	[Redacted]	Created	[Redacted]	[Redacted]
[Redacted]	[Redacted]	Created	[Redacted]	[Redacted]
[Redacted]	[Redacted]	Created	[Redacted]	[Redacted]
[Redacted]	[Redacted]	Created	[Redacted]	[Redacted]

You can also delete, update, and refresh service instances:

- Choose  (Refresh) to rebuild the table based on the resources listed in the `mta.yaml` file and the service instances created in the runtime environment.

- Choose  (Delete Service Instance) to delete a service instance from the runtime environment.
- Choose  (Update service instance configuration according to the mta.yaml) to update a service instance from the runtime environment, based on the design time configuration (mta.yaml file).

7.2.1.7 Configuring Resource Parameters

How to configure resource parameters within and outside an MTA descriptor.

For the application deployment you sometimes need to configure additional parameters for resources. You can configure these parameters directly in the MTA descriptor, for example:

```
resources:
  - name: hdi-container
    type: com.sap.xs.hdi-container
    parameters:
      config:
        schema: ${default-container-name}
```

These configurations can be lengthy. To keep the MTA descriptor short and easy to maintain, you can define them in separate .json files within the project, and specify the file's path in the resource's path of the corresponding resource. For example, it is a good practice to define security configurations in a separate file:

```
resources:
  - name: node-uaa
    type: com.sap.xs.uaa
    parameters:
      path: ./xs-security.json
```

Here is how you can specify an external configuration file in the MTA descriptor.

```
resources:
  - name: some_resource
    parameters:
      path: myconfig/configuration.json #path to the json file within the
project
```

Let's say, the specified file contains a configuration { "config1": "somevalue", "config2": [123, 456] }.

During the application build, the .json file will not be included in the resulting MTA archive. Rather, the content of this file will be integrated into the deployment descriptor (mtad.yaml) in the following way:

```
resources:
  - name: some_resource
    parameters:
      config:
        config1: somevalue
        config2:
          - 123
          - 456
```

7.2.1.8 Supported MTA Parameters When Running Applications

This topic includes a list of supported MTA parameters when running applications in SAP Web IDE.

Note

For a list of MTA parameters that are supported during deployment, see [MTA Module Types, Resource Types, and Parameters for Applications in the Cloud Foundry Environment](#).

Supported Parameters

Parameter	Scope	Read-Only (System)	Description	Default Value	Example
app-name	modules		The name of the application in the Cloud Foundry environment to be deployed for this module, based on the module name.	The module name with a generated unique prefix.	node-hello-world
buildpack	modules		The name or the URL of a custom buildpack required by the application.	Empty, or as specified in the deploy service configuration	buildpack: git://github.acme.com/xs2-java/xs2javabuildpack
command	modules		A custom command required to start the application.	Empty, or as specified in the deploy service configuration	command: node index.js
default-url	modules	Yes	The default URL, composed as <code>\${protocol}://\${default-uri}</code> . Note that the <code>\${default-uri}</code> placeholder is resolved as <code>\${host}.\${domain}</code> (host-based routing).	n/a	<code>\${protocol}://\${default-uri}</code>
disk-quota	modules		The disk space that will be available to the application. This parameter requires a unit of measurement M, MB, G, or GB in upper or lower case.	1, or as specified in module-type	disk-quota: 1G
domain	modules		The domain on which the application will be available.	<code>\${default-domain}</code>	domain: <code>\${default-domain}.acme.com</code>
generated-password	All	Yes		n/a	IG@zGg#2g-cvMvsW

Parameter	Scope	Read-Only (System)	Description	Default Value	Example
generated-user	All	Yes	A generated user ID, composed of 16 characters. The ID can contain upper and lower case letters, digits, and special characters (., -, @, \$, &, #, *).	n/a	uYi\$d41TzM1-Dm6f
health-check-type	modules		The application health check type.	http	health-check-type: port health-check-type: http health-check-type: process
host	modules		The hostname or subdomain where the application will be available.	\${default-host}	host: \${space}-node-hello-world
instances	modules		The number of application instances that will be started during the deployment.	1, or as specified in module-type	instances: 2
memory	modules		The memory limit for all instances of an application. This parameter requires a unit of measurement M, MB, G, or GB in upper or lower case.	256M, or as specified in module-type	memory: 128M
no-route	modules	Write	Configures the deployer to create or skip the creation of a route for the application described by the module.	false	no-route: true
org	All	Yes	Name of the target organization.	The current name of the target organization.	initial, trial
service	resources		The type of the created service.	Empty, or as specified in resource-type	service: hana

Parameter	Scope	Read-Only (System)	Description	Default Value	Example
service-name	resources	Yes	The name of the service in the Cloud Foundry environment to be created for this resource.	The combination of the resource name and the generated unique suffix, which conforms to the service-name restrictions for length and legal characters.	nodejs-hdi-container
service-plan	resources		The plan of the created service.	Empty, or as specified in resource-type	service-plan: hdi-shared
space	All	Yes	Name of the target organizational space.	n/a	initial, a007007
stack	modules	Read/Write	Defines which prebuilt root file system (<code>rootfs</code>) you want to use.	n/a	<div data-bbox="1098 1070 1396 1361" data-label="Code-Block"> <pre> ≡ Sample Code modules: - name: foo type: application parameters: stack: cflinuxfs3 </pre> </div>

7.2.2 Setting Up Application Projects

You can set up a multi-target application (MTA) project in various ways.

To set up an MTA project, you can either:

- Create a project from scratch
- Import a project from an archive
- Clone a project from a Git repository

You must select a Cloud Foundry space for running your MTA projects in SAP BTP. For more information, see [Select a Cloud Foundry Space \[page 452\]](#).

Related Information

[Create a Project from Scratch \[page 450\]](#)

[Import a Project from an Archive \[page 451\]](#)

[Clone a Project from Git \[page 452\]](#)

7.2.2.1 Create a Project from Scratch

You create an MTA project using a dedicated template provided by SAP Web IDE.

Procedure

1. From the *Workspace* menu, choose **New** > *Project from Template*.
2. Choose the *Multi-Target Application Project* template and click *Next*.
3. Enter a project name and click *Next*.
4. If needed, modify the *Application ID* and *Version* properties of the project. Both properties appear in the MTA descriptor (`mta.yaml` file).
5. (Optional) Enter a description of the project.
6. If you want to store your UI modules centrally, select the *Use HTML5 Application Repository* checkbox.

Template Selection > Basic Information > **Template Customization** > Confirmation

New Multi-Target Application

Template Customization

MTA Details

Application ID*	MTA_HTML5_Repo
Application Version*	0.0.1
Description	

Use HTML5 Application Repository

This option is relevant for projects that run on Cloud Foundry. For more information, see [Using the HTML5 Application Repository in a Multitarget Application \[page 453\]](#).

7. Once you have entered all the mandatory fields, click *Next* or *Finish*.

A new project with the specified name is created in your workspace. The project contains the initial MTA descriptor. For example:

```
1 | _schema-version: 2.0.0
2 | ID: HelloWorld
3 | version: 0.0.1
```

Related Information

[Setting Up Application Projects \[page 449\]](#)

[Import a Project from an Archive \[page 451\]](#)

[Clone a Project from Git \[page 452\]](#)

7.2.2.2 Import a Project from an Archive

You can import to the SAP Web IDE workspace a multi-target application project that was previously exported to an archive.

Prerequisites

- You must have a compressed (.zip) file of the project that you want to import.
- The size of the (.zip) file must not exceed 20MB.

Procedure

1. In the workspace, select the top-level `Local` folder.
2. Choose **File** > **Import** > **File or Project**.
3. Click **Browse** to locate and select your archived project file, and choose **Open**. The file name appears in the **File** field.

The destination folder name is displayed in the **Import to** field. By default, this name is the same as for the archive file. You can change the folder name, but not its location.

4. Choose **OK**. The project is created in the specified folder.

If an MTA descriptor (`mta.yaml` file), located in the project root folder, contains the definitions of modules, the corresponding subfolders are automatically converted into modules.

Related Information

[Archive Import Troubleshooting \[page 551\]](#)

[Setting Up Application Projects \[page 449\]](#)

[Create a Project from Scratch \[page 450\]](#)

[Clone a Project from Git \[page 452\]](#)

7.2.2.3 Clone a Project from Git

You can clone an existing project from a Git repository.

Procedure

Follow the instructions in [Clone Repositories \[page 371\]](#).

Related Information

[Inside an MTA Descriptor \[page 440\]](#)

[Customizing Your Project \[page 109\]](#)

[Setting Up Application Projects \[page 449\]](#)

[Create a Project from Scratch \[page 450\]](#)

[Import a Project from an Archive \[page 451\]](#)

7.2.2.4 Select a Cloud Foundry Space

Select a Cloud Foundry space for running your MTA projects in SAP BTP.

Prerequisites

You have a Cloud Foundry subaccount.

i Note

If you don't have a Cloud Foundry subaccount, ask your administrator to create a subaccount for you and to assign you to a Cloud Foundry space by following the procedure in [Create Subaccounts In Your Global Account](#).

Context

You can configure a default Cloud Foundry space to be used for running your MTA projects in SAP BTP.

If you do not want to use the defined space for a specific project, you can select a different one in [Project Settings > Cloud Foundry](#).

Procedure

1. From the *Preferences* perspective, choose *Cloud Foundry*.

i Note

When connecting to Cloud Foundry, some operations may be processed by our partner Infrastructure-as-a-Service (IaaS) providers.

2. Select the *API Endpoint* and provide your user credentials in the dialog box that opens.

i Note

If your API endpoint isn't displayed in the dropdown list, select it by entering your API endpoint in the text field.

3. Select the *Organization* and *Space* from the respective dropdown lists.
4. Save your changes.

7.2.3 Using the HTML5 Application Repository in a Multitarget Application

Developers can create a Multitarget Application that uses the HTML5 Application Repository service, and benefit from enhanced capabilities in their project.

What is the HTML5 Application Repository?

The HTML5 Application Repository is a service that allows application developers to manage the lifecycle of their HTML5 applications. At runtime, the repository enables the consuming application, typically the application router, to access static HTML5 application content in a secure and efficient manner.

For more information about working with the HTML5 Application Repository, see: [HTML5 Application Repository](#).

How to use this capability

Template Selection > Basic Information > **Template Customization** > Confirmation

New Multi-Target Application

Template Customization

MTA Details

Application ID*

Application Version*

Description

Use HTML5 Application Repository

When creating a new MTA project, select the option [Use HTML5 Application Repository](#).

Benefits of storing HTML5 applications centrally

- **Zero Downtime** - the HTML5 applications are decoupled from the consuming application router. This enables you to update the HTML5 applications without restarting the application router in the Cloud Foundry environment.
- **Reduced memory footprint in Cloud Foundry** - having only one application router generated for the entire MTA project (as opposed to one application router per HTML5 module), and storing the static content centrally, reduces the amount of memory used.

What does SAP Web IDE generate?

SAP Web IDE generates everything you need to build, deploy, and run projects that use the HTML5 Application Repository.

- **Build** - the `deploy_mode: html5-repo` parameter is added to the descriptor file (`mta.yaml`), and indicates that the `mtar` file generated by the build is compatible with the HTML5 Application Repository.
- **Deploy** - a new module is created `<mta_name>_ui_deployer`, which uploads the HTML5 modules content to the HTML5 Application Repository. The deployer module requires a resource of the HTML5 Application Repository of service plan `app-host`. In its build parameters, the deployer module specifies which HTML5 modules are uploaded to the repository. When the upload finishes, the deployer application is stopped. For more information about deploying content to the HTML5 Application Repository, see: [Deploy Content](#).
- **Consume** content from HTML5 Application Repository - a new module is created called `<mta_name>_appRouter`, which fetches the resources from the HTML5 Application Repository. The `appRouter` module requires a resource of the HTML5 Application Repository of service plan `app-runtime`. For more information about consuming content, see: [Consuming Content](#).

Changes to the mta.yaml

The following code sample illustrates the changes to the multitarget application descriptor file.

MTA without HTML5 Application Repository

```
ID: mta_without_repo
_schema-version: '2.1'
version: 0.0.1
modules:
- name: listreport_onprem
  type: html5
  path: listreport_onprem
  parameters:
    disk-quota: 256M
    memory: 256M
  build-parameters:
    builder: grunt
  requires:
    - name: uaa_mta_without_repo
    - name: dest_mta_without_repo
    - name: conn_mta_without_repo
resources:
- name: uaa_mta_without_repo
  parameters:
    path: ./xs-security.json
    service-plan: application
    service: xsuaa
  type: org.cloudfoundry.managed-service
- name: dest_mta_without_repo
  parameters:
    service-plan: lite
    service: destination
  type: org.cloudfoundry.managed-service
- name: conn_mta_without_repo
  parameters:
    service-plan: lite
    service: connectivity
  type: org.cloudfoundry.managed-service
```

MTA with HTML5 Application Repository

```
ID: mta_with_repo
_schema-version: '2.1'
parameters: //this parameter indicates that the build will generate an mtar
that is suitable to deploy and consume HTML5 applications from
the HTML5 Application Repository
  deploy_mode: html5-repo
version: 0.0.1
modules:
- name: mta_with_repo_appRouter //this new module is responsible for consuming
HTML5 applications from the HTML5 Application Repository
  type: approuter.nodejs
  path: mta_with_repo_appRouter
  parameters:
    disk-quota: 256M
    memory: 256M
  requires:
    - name: rt_mta_with_repo_appRouter
    - name: dest_mta_with_repo
    - name: conn_mta_with_repo
    - name: uaa_mta_with_repo
- name: mta_with_repo_ui_deployer //this new module is responsible for storing
HTML5 applications in the HTML5 Application Repository
```

```

type: com.sap.html5.application-content
path: mta_with_repo_ui_deployer
requires:
  - name: dt_mta_with_repo_ui_deployer
build-parameters:
  requires:
    - name: listreport_onprem
      artifacts:
        - './*'
      target-path: resources/listreport_onprem
- name: listreport_onprem
  type: html5
  path: listreport_onprem
  build-parameters:
    builder: grunt
resources:
  - name: rt_mta_with_repo_appRouter //this new resource enables content
consumption from the HTML5 Application Repository
  parameters:
    service-plan: app-runtime
    service: html5-apps-repo
  type: org.cloudfoundry.managed-service
  - name: dt_mta_with_repo_ui_deployer //this new resource enables content
deployment to the HTML5 Application Repository
  parameters:
    service-plan: app-host
    service: html5-apps-repo
  type: org.cloudfoundry.managed-service
- name: uaa_mta_with_repo
  parameters:
    path: ./xs-security.json
    service-plan: application
    service: xsuaa
  type: org.cloudfoundry.managed-service
- name: dest_mta_with_repo
  parameters:
    service-plan: lite
    service: destination
  type: org.cloudfoundry.managed-service
- name: conn_mta_with_repo
  parameters:
    service-plan: lite
    service: connectivity
  type: org.cloudfoundry.managed-service

```

Changes to the xs-app.json

The following code sample illustrates the changes to the `xs-app.json` file, routes section.

```

{
    "source": "^/webapp(.*)$",
    "target": "$1",
    "service": "html5-apps-repo-rt", //added to each module
    "authenticationType": "xsuaa"
}

```


7.2.4 Developing SAP HANA Database (HDB) Modules

An SAP HANA database (HDB) module is a collection of related design-time database artifacts, such as data models, views, or procedures.

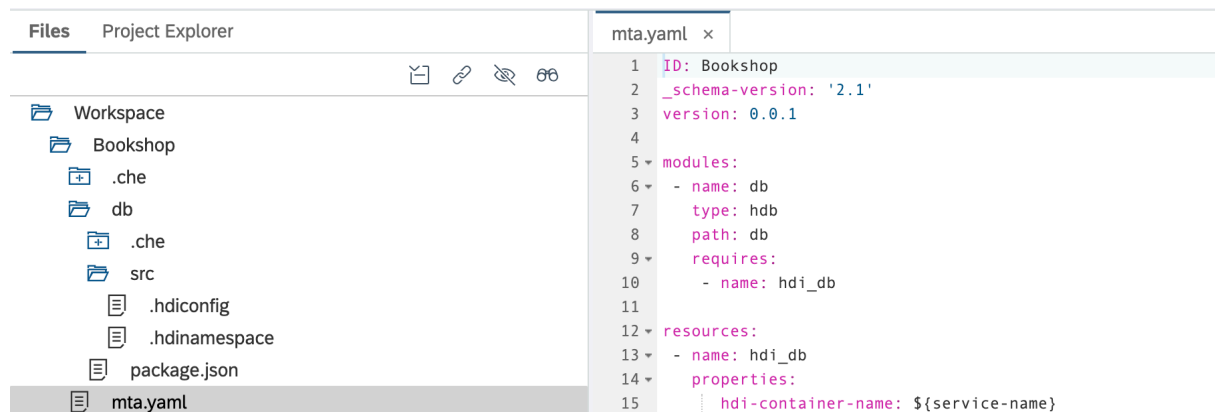
To develop and deploy these artifacts into the SAP HANA database, perform the following steps:



- [Create an HDB Module \[page 458\]](#)
- [Developing Database Artifacts \[page 459\]](#)
- [Inside an MTA Descriptor \[page 440\]](#)
- [Build an HDB Module \[page 462\]](#)

Module Folder Structure

The following figure depicts a sample HDB module folder structure alongside the corresponding entry in the `mta.yaml`.



Folder	Description
<code><module name></code>	Should not contain any files.
<code>src</code> (default)	The default location in which you store your design-time database artifacts. If needed, you can create additional subfolders for the same purpose.

Related Information

[Defining the Data Model in XS Advanced](#)

7.2.4.1 Create an HDB Module

You can create new SAP HANA database (HDB) modules.

Procedure

From the project context menu, choose **New** > **SAP HANA Database Module**, and follow the wizard steps to enter the module properties.

A new HDB module with the specified name is created in your project, and a corresponding section is added to the MTA descriptor (`mta.yaml`). For example:



Related Information

[Developing SAP HANA Database \(HDB\) Modules \[page 457\]](#)

[Import an HDB Module \[page 458\]](#)

7.2.4.2 Import an HDB Module

You can import archived SAP HANA database (HDB) modules.

Prerequisites

The `.zip` module archive that you want to import, which was exported from another MTA project, is available in the file system.

Procedure

1. From the root folder of the project, choose **File > Import > File or Project**.
2. Click *Browse* to locate and select your archive, and choose *Open*. The file name appears in the *File* field. The destination folder is displayed in the *Import to* field. To change this folder, choose *Select Folder*, and browse to the required folder, or create a new folder.

The specified folder, containing the artifacts extracted from the archive, is created in the project.

3. To make the imported folder a proper module in your project, you need to convert it into a module of the matching type. From the folder context menu, choose *Convert To*, and then the type of the target module.

i Note

The conversion process does not check whether the imported folder structure matches the selected module type. The process does not generate the module artifacts according to the selected type.

The imported module becomes a part of your MTA project, and the module entry is added to the MTA descriptor.

7.2.4.3 Developing Database Artifacts

Create the database artifacts required for your module.

Procedure

From the context menu of the module's `src` subfolder, choose *New*, and then choose one of the available artifacts.

i Note

If you don't see the required artifact type in the menu, this means that the optional feature (extension), which supports it, is disabled. To enable the relevant feature (extension), follow the instructions in [Enable SAP Web IDE Extensions \[page 510\]](#).

Artifact	Instructions
Database procedure (.hdbprocedure)	<p>Choose <i>Procedure</i>, and enter the file name. The new artifact is added to the module, and opens in a dedicated code editor.</p> <p>For information about developing HDB procedures, see Defining the Data Model in XS Advanced.</p>
Calculation view (.hdbcalculationview)	<p>Choose <i>Calculation View</i>. Enter a name, label, and select the type and category. The new artifact is added to the module, and opens in the chosen editor.</p> <p>For information about developing calculation views for SAP HANA Cloud, see Modeling Calculation Views in SAP Web IDE Full-Stack.</p>

Artifact	Instructions
A design-time definition of a calculation view, which can be referenced in an OData service definition.	<p>For information about developing calculation views in SAP HANA as a Service, see Modeling Calculation Views in SAP Web IDE.</p> <p>For information about developing calculation views in SAP Web IDE for SAP HANA (on-premise), see Creating a Graphical Calculation View..</p>
CDS (Core Data Services) Document (.hdbcds)	<div style="background-color: #f0f0f0; padding: 10px;"> <p>i Note</p> <p>This option is not available in SAP HANA Cloud.</p> </div> <p>Choose <i>HDB CDS Artifact</i>, and enter a name. The new artifact is added to the module, and opens in the default editor defined in Tools > Preferences > Default Editors > HDB CDS Source.</p> <p>For information about developing CDS documents, see:</p> <ul style="list-style-type: none"> ○ In the text editor: Create a CDS Document (XS Advanced) ○ In the graphical editor: Getting Started with the CDS Graphical Editor.
Analytic privilege (.hdbanalyticprivilege)	<p>Choose <i>Analytic Privilege</i>, and enter a name and label. The new artifact is added to the module, and opens in the dedicated editor.</p> <p>For information about developing analytic privileges, see Defining Data Access Privileges.</p>
Role (.hdbrole)	<p>You can create a new role and grant it additional roles or object, schema, analytic and system privileges. Choose <i>Role</i> and enter a name for the new role. Optionally, you can also choose an <code>.hdbroleconfig</code> file. To edit an existing role, right-click the role in the catalog and click <i>Open Role Editor</i>.</p> <p>For more information about roles, see Roles (.hdbrole).</p>

i Note

Currently, SAP Web IDE provides dedicated editors only for the artifacts listed above. You can develop other artifacts, supported by SAP HANA XS Advanced (XSA), using a text editor. Create a file in the module's `src` subfolder with an appropriate extension, and open it in a text editor.

You can now easily access the database artifacts of an HDB module in the database explorer. To do this, build your HDB module, and choose *Open HDI Container* from the module's context menu. The HDI (HANA Deployment Infrastructure) container with the deployed database artifacts of the module opens in the database explorer.

7.2.4.4 Connect to an External HDB Service

You can configure a connection to an external SAP HANA database (HDB) service outside of your current schema.

Prerequisites

- Your cloud information is updated in the MTA parent repository.
- The associated repository has a valid `mta.yaml` file.

Context

There are 2 ways to connect to an external HDB service:


- Using a user-provided service that holds connection details to the target HDB service. You can either select an existing user-provided service or create a new one.
- Binding to an existing HDI container service referring to the schema to which you want to connect.


The results of connecting to an external HDB service are:

- The creation of the corresponding dependency configuration in the `mta.yaml` file.
- If you choose to create a new user-provided service, the service is created in the target Cloud Foundry space that was configured in the project settings.

Procedure


1. Open SAP Web IDE.
2. In the *Workspace* view, right-click the SAP HANA Database Module in which you want to configure the `mta.yaml` file.
3. Choose **New** > *SAP HANA Service Connection*.
4. Configure the `mta.yaml` file according to one of the following scenarios.

Scenario	Steps
Adding an existing non-HDI container service	<ol style="list-style-type: none">1. In the <i>Add SAP HANA Service Connection</i> dialog box, from the <i>Type</i> field, select <i>Non-HDI Container</i>.2. From the <i>Action</i> field, select <i>Use an existing user-provided service</i> and select the required service.3. Optional - Click  to test the connection to SAP HANA Database.4. Choose <i>Add</i>.

Scenario	Steps
	The user-provided service is added and the <code>mta.yml</code> file is configured.
Creating a new non-HDI container service	<ol style="list-style-type: none"> 1. In the <i>Add SAP HANA Service Connection</i> dialog box, from the <i>Type</i> field, select <i>Non-HDI Container</i>. 2. From the <i>Action</i> field, select <i>Create a new user-provided service</i> and enter a name. 3. Provide details of the SAP HANA database that the selected user-provided service refers to. 4. Choose <i>Add</i>. The user-provided service is created in the Cloud Foundry space and the <code>mta.yml</code> file is configured.
Adding an existing HDI container service	<ol style="list-style-type: none"> 1. In the <i>Add SAP HANA Service Connection</i> dialog box, from the <i>Type</i> field, select <i>HDI Container</i>. The list of all HDI services is displayed. 2. Select the required HDI service that you want to configure in the <code>mta.yml</code> file. 3. Optional - Click  to test the connection to SAP HANA Database. 4. Choose <i>Add</i>. The service is added and the <code>mta.yml</code> file is configured.

Note

When creating a new user-provided service, you can define a secure connection to SAP HANA.


If you require a SAP HANA database certificate to establish this connection, you can download it from [DigiCert](#) 

7.2.4.5 Build an HDB Module

The build process deploys the design-time database artifacts and generates the corresponding actual objects in the SAP HANA database.

Procedure

From the module context menu, choose *Build*.

The steps and messages of the build process are displayed in the console that opens automatically at the bottom of the browser window. You can show and hide the console by choosing  from the main menu.

i Note

Sometimes the build fails because there is an error in the MTA descriptor (`mta.yaml`). The error messages displayed in the console indicate the corrections you should make in the descriptor. For more information about the descriptor syntax, see [Inside an MTA Descriptor \[page 440\]](#).

After making the corrections, rebuild the module.

By default, the build process generates a database schema with a unique name derived from the `schema` parameter defined in `mta.yaml`. If you want the schema name to be exactly the same as defined, set the `makeUniqueName` parameter to `false`.

For example:

Sample Code

```
- name: hdi-container
  parameters:
    config:
      schema: mySpecialSchema
      makeUniqueName: false
  properties:
    hdi-container-name: ${service-name}
  type: com.sap.xs.hdi-container
```

However, if you do so and use the same schema name in different projects, naming conflicts might occur in the database during builds.

i Note

If you have an MTA extension file with the name `dev.mtaext` that extends the `mta.yaml` file in the MTA project root, its contents will be merged with the `mta.yaml` file when running the module.

i Note

By default, the HDI deployer removes any object from the HDI container that does not exist in the database module. To prevent the HDI deployer from deleting an artifact that is not in the developer's database module, you need to set the `auto_undeploy` property to `false` in the `mta.yaml` file.

For development, set to `false` the `auto_undeploy` property in the MTA extension file (`dev.mtaext`) that will be automatically merged with the definitions provided in the `mta.yaml` file for building an HDB module and running operations. Do not change the `mta.yaml` file.

Sample Code

Editing the `dev.mtaext` file

```
ID: anyname
_schema-version: '2.1'
extends: mtaname

resources:
- name: hdi_db
  properties:
    hdi-container-name: ${service-name}
    HDI_DEPLOY_OPTIONS:
```

```
    auto_undeploy: false
  type: com.sap.xs.hdi-container
  parameters:
    service-name: my-service-name
```

For building and deploying to the target environment, set the `auto_undeploy` property to `false` in the `mta.yaml` file.

Sample Code

Editing the `mta.yaml` file

```
modules:
- name: db
  type: hdb
  path: db
  requires:
  - name: hdi_db
    properties:
      HDI_DEPLOY_OPTIONS:
        auto_undeploy: false
```

Related Information

[Selective Build of HDB Modules \[page 464\]](#)

[Developing SAP HANA Database \(HDB\) Modules \[page 457\]](#)

[HDI Deploy Options](#)

7.2.4.5.1 Selective Build of HDB Modules

You can optimize the build process of an SAP HANA Database (HDB) module by building only selected artifacts.

The full build of an HDB module containing many database artifacts can be time-consuming. Sometimes, when developing a module, you don't need to build the whole module. You can save time by building only the artifacts that you are currently working on.

In the module folder, select all the artifacts that you want to build, and choose *Build Selected Files* from the context menu.

Note

To use selective build effectively, note the following:

- Selective build works only if the selected files don't have dependencies on other artifacts that are not yet built or are not included in the selection.
- Selective build includes only the files that are selected explicitly. To include hidden files in a build, choose the option *Show Hidden Files* from the *View* menu, and then select the required files.

Deployment options

You have additional options to deploy your database artifacts.

Force Deployment

By default, only changed artifacts are deployed. If you want to deploy a single artifact that hasn't been changed, right-click it and choose *Build Selected Files*.

Simulate Build

You can test the integrity of your HDB module without deploying all its artifacts by simulating a build. Open the *Problems* view, select the HDB module, and click *Analyze and Display Problems*. This executes the build simulation command and displays the errors that occurred.

Related Information

[Build an HDB Module \[page 462\]](#)

[Developing Database Artifacts \[page 459\]](#)

7.2.4.6 Build an HDB Module in a Multi-Database Environment

The build process deploys the database artifacts and generates the corresponding actual objects in the SAP HANA databases.

If you are working in a Cloud Foundry space with multiple SAP HANA databases, you must define the database to which you want to deploy the module in the `mta.yaml` file before you deploy.

Procedure

1. Open your `mta.yaml` file.
2. Select the relevant `hdi-container` resource.
3. In the parameters section, add the SAP HANA database ID.

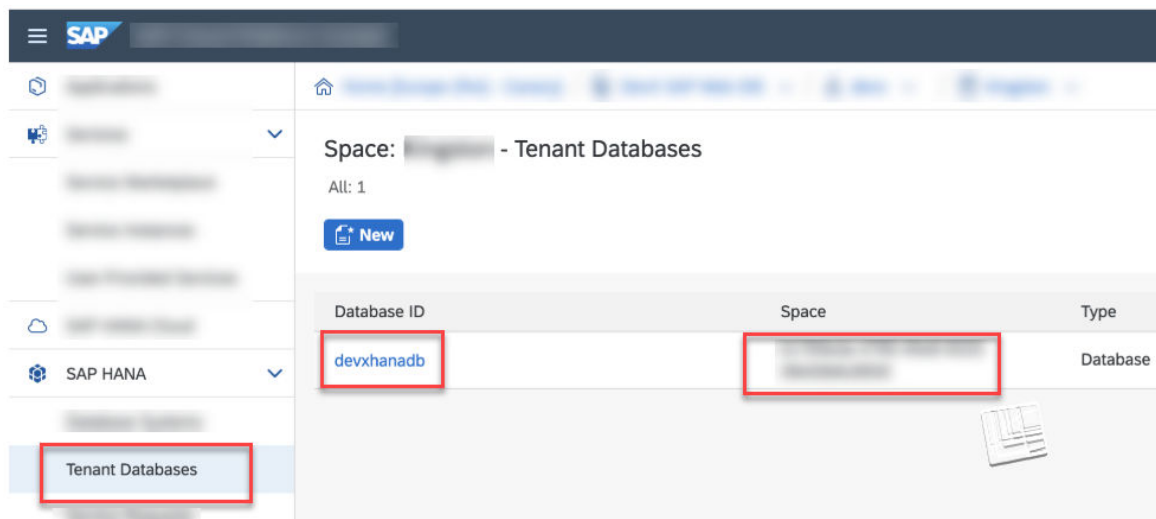
Sample Code

```
resources:
- name: hdi_db
  properties:
    hdi-container-name: ${service-name}
  parameters:
    config:
      database_id: '<database_id>'
    type: com.sap.xs.hdi-container
```

4. From the module context menu, choose *Build*.

Obtaining the Database ID for a PSA SAP HANA Database

1. In the SAP BTP cockpit, login to your Cloud Foundry space.
2. Select **SAP HANA** > **Tenant Databases**.
3. Search for the relevant SAP HANA database from the list.
4. Create the ID from the combination of the content in the Space and the Database ID fields:
'<space:Database ID>'



Obtaining the Database ID for an SAP HANA Cloud Database

1. From the CLI, login to your Cloud Foundry space.

```
cf login -a <cf end point>
```

2. Obtain a list of the hana-cloud services in your space.

```
cf service | grep hana-cloud
```

3. Extract the database ID from the relevant SAP HANA Cloud database service name.

```
cf service <hana-cloud-service-name> --guid
```

Obtaining the Database ID for an SAP HANA as a Service Database

1. From the CLI, login to your Cloud Foundry space.

```
cf login -a <cf end point>
```

2. Obtain a list of the hana-db services in your space.

```
cf service | grep hana-db
```

3. Extract the database ID from the relevant SAP HANA as a Service database service name.

```
cf service <hana-db-service-name> --guid
```

i Note

If you do not provide a database ID, the build process will fail. Before you can try again, you need to manually delete the `hdi-shared` service from your Cloud Foundry space.

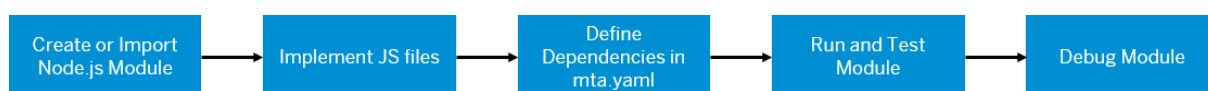
7.2.5 Developing Node.js Modules

A Node.js module is a collection of related JavaScript files and service definitions that implement the business logic of your application.

Prerequisites

You have enabled the optional *Tools for Node.js Development* extension. For instructions, see [Enable SAP Web IDE Extensions \[page 510\]](#).

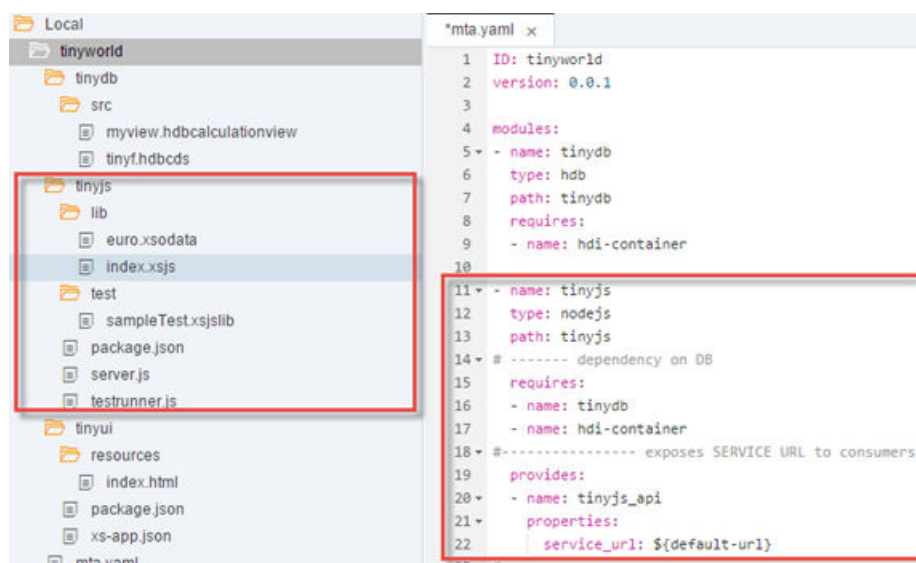
To develop Node.js modules, perform the following steps:



- [Create a Node.js Module \[page 468\]](#)
- [Implementing Node.js files \[page 468\]](#)
- [Inside an MTA Descriptor \[page 440\]](#)
- [Run Node.js Modules \[page 471\]](#)
- [Debug Node.js Modules \[page 476\]](#)

Module Folder Structure

The following figure depicts a sample Node.js module folder structure alongside its corresponding entry in the `mta.yaml`.



Folder	Description
<code><module name></code>	Contains the main <code>.js</code> file, which by default is <code>server.js</code> , and the <code>package.json</code> file (description and dependencies of Node.js modules). You can place other <code>.js</code> files anywhere in this folder.
<code>lib</code>	Created only if XSJS support is enabled for the module. Contains <code>.xsjs</code> (SAP HANA XS JavaScript) source files.
<code>test</code>	Created only if XSJS support is enabled for the module. Contains XUnit test (<code>.xsjslib</code>) files.
<code>tests</code>	This folder is created for modules without XSJS support. It contains <code>.js</code> test files.

Implementing Node.js files

In the root module folder, create and implement the required `.js` files.

Related Information

[Developing Multitarget Applications \[page 438\]](#)

<https://nodejs.org/en/> 

[Node.js Data Services in XSJS Compatibility Mode](#)


[Create a Node.js Module \[page 468\]](#)

[Import a Node.js Module \[page 470\]](#)

7.2.5.1 Create a Node.js Module

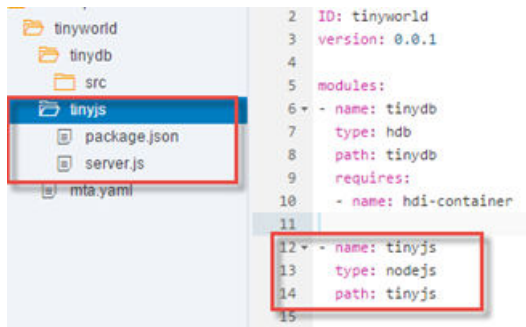
You can create new Node.js modules.

Procedure

1. From the project context menu, choose **New** > **Node.js Module** .
2. Follow the wizard steps to enter the module properties.

If you want to run your module in the XS JavaScript (XSJS) compatibility mode, select **Enable XSJS support**.

A new Node.js module with the specified name is created in your project, and a corresponding section is added to the MTA descriptor (`mta.yaml`). For example:



The default files for the new module contain basic Hello World code, which varies according to whether you enabled XSJS support.

Related Information

[Developing Node.js Modules \[page 467\]](#)

[Import a Node.js Module \[page 470\]](#)

7.2.5.1.1 Implement OData Service Definitions

Context

To expose a part of the data model, such as a table or a view, using OData, you should create a corresponding OData service definition in an `.xsodata` file. Since these files must be deployed to a JavaScript runtime, you should place them in a Node.js module.

Procedure

1. From the context menu of a Node.js module's `lib` subfolder, choose **New > File**.
2. In the *New File* box that opens, enter a file name with the extension `.xsodata`.
3. Double-click the file to open it in the dedicated editor. Use the Ctrl-Space key combination for hints while editing.

For more information, see [Defining OData v2 Services for XS Advanced JavaScript Applications](#)

7.2.5.2 Import a Node.js Module

You can import archived Node.js modules.

Prerequisites

The `.zip` module archive that you want to import, which was exported from another MTA project, is available in the file system.

Procedure

1. From the root folder of the project, choose **File > Import > File or Project**.
2. Click *Browse* to locate and select your archive, and choose *Open*. The file name appears in the *File* field. The destination folder is displayed in the *Import to* field. To change this folder, choose *Select Folder*, and browse to the required folder, or create a new folder.

The specified folder, containing the artifacts extracted from the archive, is created in the project.

3. To make the imported folder a proper module in your project, you need to convert it into a module of the matching type. From the folder context menu, choose *Convert To*, and then the type of the target module.

i Note

The conversion process does not check whether the imported folder structure matches the selected module type. The process does not generate the module artifacts according to the selected type.

The imported module becomes a part of your MTA project, and the module entry is added to the MTA descriptor.

Related Information

[Developing Node.js Modules \[page 467\]](#)

[Create a Node.js Module \[page 468\]](#)

7.2.5.3 Run Node.js Modules

To create a running application in XS Advanced and resolve all dependencies defined in `mta.yaml`, run each Node.js module in your project.

Prerequisites

Before running a module, make sure that:

- All the relevant dependencies of the module are defined in the MTA descriptor (`mta.yaml`).
- All the modules and resources on which the module is dependent are implemented and available .

i Note

You don't need to build your module explicitly before running, because it will be built automatically during the run process.

Procedure

From the module context menu, choose *Run*, then one of the following options:

Option	Description
<i>Run Configurations</i>	Create a new or modify an existing run configuration. For more information, see Create Run Configurations for Node.js Modules [page 474] .
<i>Run as Node.js Application</i>	Run the main <code>.js</code> file, which by default is <code>server.js</code> .

i Note

If you have an MTA extension file with the name `dev.mtaext` that extends the `mta.yaml` file in the MTA project root, its contents will be merged with the `mta.yaml` file when running the module.

Results

When the build finishes, its result is available in your workspace in the `node_modules` folder.

i Note

When you run a module for the first time, it is pushed to XS Advanced and bound to the required XS Advanced services that are defined in the MTA descriptor, which can take some time. After this, the run starts more quickly. If you modify only the source files, or `xs-app.json` in HTML5 modules, the module remains in XS Advanced, and the changes are injected directly into the running module.

Related Information

[Show Dependency Updates in package.json \[page 472\]](#)

[Use the Run Console \[page 473\]](#)

[Test Node.js Modules \[page 475\]](#)

[Create Run Configurations for Node.js Modules \[page 474\]](#)

7.2.5.3.1 Show Dependency Updates in package.json

You can check for dependency updates in the external dependencies defined in the `package.json` file.

Procedure

1. Select your Node.js module and from the toolbar choose **Build** > *Show Dependency Updates*.
2. Then go to the *Problems* view to see the outdated dependencies.

Related Information

[Run Node.js Modules \[page 471\]](#)

7.2.5.3.2 Scan and Fix Vulnerabilities

Find and fix vulnerabilities in your project's node modules.

Context

Check your direct dependencies defined in your modules `package.json`.

Procedure

Select your Node.js module and from the toolbar choose **Build** > *Scan for Vulnerabilities* or **Build** > *Scan and Fix Vulnerabilites*.


Options	Description
Scan for Vulnerabilities	Executes <code>npm audit</code> and lists found vulnerabilities in node modules included in your project.
Scan and Fix Vulnerabilites	Executes <code>npm audit fix</code> , which lists and, if possible, fixes found vulnerabilities in node modules included in your project. This action changes your <code>package.json</code> to fix vulnerabilities.

Related Information

<https://docs.npmjs.com/cli/audit> 




7.2.5.3.3 Use the Run Console

Progress and status messages, generated during the run process, as well as the log records, are displayed in the Run console.

The console opens automatically in the main browser tab. You can show and hide the console by clicking the  (*run console*) icon in the bottom pane's toolbar.

The left pane of the console displays a list of all running modules in the current MTA project along with the corresponding run configurations. Each entry is accompanied by a status icon with a tooltip. Click the run configuration name below a module name to view its most recent run log.

The controls displayed at the top of the console enable you to perform the following tasks:

- If the application URL is visible, click it to access the running application in the browser.
- To view the detailed server log of a run process, click *Logs*.
- To clear the log in the console, click  (*clear the log*).
- To stop running the module, click  (*stop*).
This causes the module to be removed from the runtime system, so that if you run the module again, the run process starts from the beginning.
- To run the module using a different configuration, select the configuration and choose the  (*run*) button in the task bar.

Related Information

[Run Node.js Modules \[page 471\]](#)

[Test Node.js Modules \[page 475\]](#)

7.2.5.3.4 Create Run Configurations for Node.js Modules

You can configure how to run, test, and debug Node.js modules in your project.

Procedure

1. From the context menu of a Node.js module in your project, choose **Run > Run Configurations**.
2. To create a new run configuration for a module, click **+ > Node.js Test**.
A new configuration with a default name appears under the selected category in the left pane.
3. Select a configuration to modify its name, and edit its properties as follows.

General tab

Property	Description
<i>Start with package.json script</i>	Run one of the scripts listed in the <code>package.json</code> file of the module, for example, a test script generated in an XSJS file.
<i>Start with application file</i>	Run a .js file of your module.
<i>Application File</i>	The name of .js file to run.
<i>Debugging Enabled</i>	If debugging is enabled, you can select or unselect the <i>Break on first statement</i> .
<i>Break on first statement</i>	This is only possible if you start your application with <i>Debugging Enabled</i> . Use this option to stop the application on the very first statement, even during the bootstrapping or initializing phase.
<i>Script Name</i>	Select a script name: <ul style="list-style-type: none">○ <i>start</i> Start the application.○ <i>test</i> Start the application with test execution.○ <i>test-coverage</i> Start the application with test execution and show code coverage. This script requires more resources than others.
<i>Arguments</i>	(Optional) The arguments to be passed to the script.
<i>Open a new browser window</i>	Select whether to open the script in a new browser tab. By default, this option is unselected for .js files.
<i>Service Path</i>	(Optional) Enter a relative path to append to the server root. The browser is redirected to this path.

4. To save the configuration and run your module using this configuration, choose *Save and Run*. To save all configurations without running, choose *OK*. To discard all your changes, choose *Cancel*.

These settings are persistent and are used every time that you run your application with the selected run configuration, until you edit or delete the settings.



Related Information

[Run Node.js Modules \[page 471\]](#)




7.2.5.4 Test Node.js Modules






Build and test your JavaScript application with unit tests.

Procedure

1. Add unit tests in the `test` folder structure.
2. Execute your test run configuration, which could be named, for example, *Run script test*.
3. From the context menu of your module, select  *Run* .
4. In your run configuration, specify whether to run the test with code coverage.
5. Review the test results.

Results

Each JavaScript module must provide its own unit tests. The JavaScript module template comes with a sample unit test. Use this test to get familiar with the test functionality and the  (*test results*) pane. By default, the  (*test results*) pane shows *Test Results*, *Stack Trace*, and *Coverage*. If you have specified it in the run configuration, code coverage, which enables you to write targeted unit tests for uncovered code, is shown in the editor. To disable this function, edit your  (*settings*) .

You can also use the  (*test results*) pane to  (*export*) or  (*delete*) all test results for the selected module. Your workspace is connected to your  (*test results*) pane. The results you see are for the currently selected module, which is also referenced in the pane as the path to your current module (`/<projectname>/<modulename>`). If you miss the latest test result, select  (*refresh*). Also check the logs to see if your test execution failed; if it has, no test result is available.

Related Information

[Run Node.js Modules \[page 471\]](#)

7.2.5.5 Debug Node.js Modules

Debug your application on demand using the debugger panel.

Prerequisites



Before debugging a Node.js module in the debugger panel, make sure that the following conditions are true:

- You have run your application successfully.
- Your Node.js application is running on Node.js 8.1 or higher.
If you use a different runtime version, you can debug your application using command-line tools.

i Note

To debug the initializing phase of your application, configure a run configuration and make sure to select *Break on first statement*

Procedure

1. Open the  (*debugger*) panel.
2.  (*attach*) your application.
3. Choose your multi target application and select a debug target.

→ Tip

If you want to debug a test do the following:



Create a run configuration that starts with an application file and select *rungulp.js* as the application file. Then select *Debugging Enabled* and *Break on first statement* and run your application.

Results



When the application reaches a breakpoint, you are notified that the execution is suspended. In this panel, you can perform the regular debugging tasks, such as viewing the call stack, examining the variables, stepping in and out of the functions, and so on. In the debugger console, you can also perform interactive evaluation of JavaScript statements in the context of the current stack frame.

7.2.5.5.1 Conditional Breakpoints

If you open the context menu on a breakpoint, select [Edit Breakpoint Condition](#) to enhance that breakpoint with a condition.

Breakpoint Status	Description
Enabled standard breakpoint	<p>Shape outline: blue</p> <p>Shape fill: light blue</p> <p>You have enabled the breakpoint, either explicitly, which means one by one, or globally. To enable breakpoints, go to the Breakpoints section and select the corresponding checkbox or activate breakpoints globally by pressing the  (activate breakpoints) button.</p> <p>Explicitly disabled breakpoints are not activated using the global option. You need to enable them explicitly.</p>
Enabled conditional breakpoint	<p>Shape outline: orange</p> <p>Shape fill: light orange</p> <p>You have defined an expression for a breakpoint. If the condition is met, the application suspends when it reaches the breakpoint.</p>
Disabled standard and conditional breakpoint	<p>Shape outline: blue or orange</p> <p>Shape fill: transparent</p> <p>You have disabled the breakpoint, either explicitly, which means one by one, or globally. To disable breakpoints, go to the Breakpoints section and select the corresponding checkbox or deactivate breakpoints globally by pressing the  (deactivate breakpoints) button.</p> <p>Explicitly disabled breakpoints are not activated using the global option. You need to enable them explicitly.</p>
Checkmark	<p>Shape outline and fill are not affected.</p> <p>You set the breakpoint successfully on a running debugging session. The checkmark indicates that the corresponding source file has been loaded. As a consequence, the application suspends when it reaches the breakpoint. For a conditional breakpoint, the user-defined expression has to be met.</p> <p>A breakpoint without a checkmark means, there is no debug session running, the source file has not been loaded, or the breakpoint could not be successfully resolved. An unresolved breakpoint may occur when the workspace resource version and the currently executed resource version don't match.</p>

7.2.5.5.2 JSON Data Preview

When you are inspecting variables in the  ([debugger](#)) panel you can see the current JSON for a complex variable. To see the JSON, select a variable and then  ([json data preview](#)). The JSON might contain a variable



with the value "<object variable not loaded>". In such a case, the complex variables are nested and only those that you already inspected are contained in this JSON.

Here is an example:

```
{
  "Local": {
    "this": {},
    "exports": "<object variable not loaded>",
    "require": "function require(path) {",
    "module": "<object variable not loaded>",
    "__filename": "/path/to/your/app/server.js",
    "__dirname": "/path/to/your/app",
    "http": "undefined",
    "port": "undefined"
  }
}
```

Objects are loaded when you expand them in the nested overview. In our example you already expanded `Local`. If you expand `exports` your JSON file grows.

7.2.5.5.3 Pause on Exceptions

You have started your application and attached the debugger. If the application should be paused in the event of an exception, go to the part of the  *(debugger)* panel where you (de-)activate breakpoints. Select  *(pause on exceptions)* and then one of the options.

If you decide to let your application pause in the event of an exception, the line where the exception occurs is highlighted.

7.2.6 Developing Java Modules

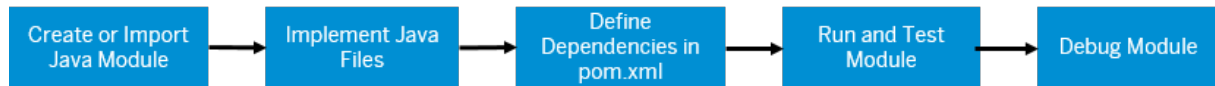
A Java module is a collection of related Java files and service definitions. Java modules implement the business logic of your application, either instead of or in addition to Node.js modules. A Java module can be either a Java Web Archive (WAR) or Java Archive (JAR) built with Apache Maven.

Prerequisites

- You have enabled the optional *Tools for Java Development* feature. For instructions, see [Enable SAP Web IDE Extensions \[page 510\]](#).

Workflow

To develop Java modules, perform the following steps:



- [Create a Java Module \[page 480\]](#)
- [Implementing Java Files \[page 479\]](#)
- [Define Dependencies \[page 482\]](#)
- [Run a Java Module \[page 483\]](#)
- [Debug Java Modules \[page 485\]](#)

Module Folder Structure

The default files of a new Java module (except Web application with OData support) already contain a basic, ready-to-run "Hello World" servlet. The module is structured following the Apache Maven standard directory layout.

Using a settings.xml File

If you decide to use a `settings.xml` file to provide project-specific Apache Maven settings, be aware that this file overrides default settings. During an update, changes to defaults do not overrule your `settings.xml`. Make sure you read the release notes to identify possible gaps between your `settings.xml` and the recommended default settings.

If needed, create the `settings.xml` next to the `pom.xml` on the same level. For more information about this optional step, see [Define Dependencies \[page 482\]](#)

Implementing Java Files

In the `src/main/java` folder, create and implement the required Java files.

→ Tip

While you are working on a Java file in a Java module, you can get assistance from the following features:

- Code Assist
Provides code proposals as explained below.
In a Java file, press `Ctrl` + `Space` for Java keyword artifact and local objects suggestions.

- **Code Validation**
Provides code validation as described below.
Problems are displayed in the file gutter. In addition, the problems view at the bottom of the screen displays complete problems in your Java module.

Related Information

[Apache Maven - Introduction to the Standard Directory Layout](#) 


[The SAP HANA XS Advanced Java Run Time](#)

[Developing Multitarget Applications \[page 438\]](#)

7.2.6.1 Create a Java Module

You can create a new Java module by following the steps below.

Procedure

1. From the project context menu, choose **New** > **Java Module** , and enter a module name.
2. Define your basic module settings. Depending on your application needs, choose one of the Java module templates. To see all the available templates, choose *All Categories* from the *Category* dropdown list.

Option	Description
<i>Simple Web Application</i>	Creates a simple Web application with a standard Maven directory layout. Produces a * .war file that contains a basic ready-to-run "Hello World" servlet.
<i>OData V2 Service</i>	A Java Web application module that exposes an OData V2 service. The application is based on the SAP BTP SDK for service development. Produces a * .war file and provides an OData V2 endpoint.
<i>OData V4 Service</i>	A Java Web application module that exposes an OData V4 service. The application is based on the SAP BTP SDK for service development. Produces a * .war file and provides an OData V4 endpoint.

Option	Description
Multi-Module Web Application	Creates a Web application consisting of two Java modules, common and web. The Maven build produces *.war and *.jar files with dependencies for deployment.
Spring Boot Application	Creates a module that is build with the Spring Boot framework using Maven. Produces a *.jar file that brings its own runtime.

3. Define other settings. If the default values don't fit your needs, change the Maven POM settings.

Related Information

[Import a Java Module \[page 481\]](#)

[Maintaining OData Services in XS Advanced](#)

7.2.6.2 Import a Java Module

You can import an archived Java module by following the steps below.

Prerequisites

The .zip module archive that you want to import, which was exported from another MTA project, is available in the file system.

Procedure

1. From the root folder of the project, choose **File > Import > File or Project**.
2. Click **Browse** to locate and select your archive, and choose **Open**. The file name appears in the **File** field. The destination folder is displayed in the **Import to** field. To change this folder, choose **Select Folder**, and browse to the required folder, or create a new folder.

The specified folder, containing the artifacts extracted from the archive, is created in the project.
3. To make the imported folder a proper module in your project, you need to convert it into a module of the matching type. From the folder context menu, choose **Convert To**, and then the type of the target module.

i Note


The conversion process does not check whether the imported folder structure matches the selected module type. The process does not generate the module artifacts according to the selected type.

The imported module becomes a part of your MTA project, and the module entry is added to the MTA descriptor.

7.2.6.3 Define Dependencies

You can define optional dependencies of a Java module after creating it.

Procedure

1. Modify `pom.xml` to add dependencies as found on the Maven Web site.
2. If your SAP Web IDE server cannot access the Internet, in particular `https://repo1.maven.org/maven2/`, you can set up your own Maven repository manager. Such custom repository setups can be configured by adding a `settings.xml` file to your Java module.
3. To identify outdated dependencies, select your Java module and go to **File** > *Show dependency updates*  Use the *Problems* view to see outdated dependencies.

Related Information

[Create a Java Module \[page 480\]](#)

[Run a Java Module \[page 483\]](#)

[Inside an MTA Descriptor \[page 440\]](#)

<http://search.maven.org/> 

<https://maven.apache.org/settings.html> 

<https://maven.apache.org/repository-management.html> 

<https://repo1.maven.org/maven2/> 

7.2.6.4 Build a Java Module

You can perform a full module build, or any build phase, or even a specific task (plugin goal) by using options from the module context menu.

Option	Description
<i>Build</i>	Build the module by executing the Maven <code>clean install</code> command.
<i>Execute Maven Goal...</i>	Execute any build lifecycle phase or even a granular task (plugin goal) by entering a Maven command without the <code>mvn</code> prefix in the <i>Command Line</i> field, and choosing <i>Execute</i> . Note that the input is not validated. If you enter an invalid command, the build will fail, and an error message <code>Unknown lifecycle phase...</code> will be displayed in the console.

i Note

All the artifacts, created by a Java module build in the Cloud Foundry environment under the `/src/gen` subfolder of the module, are copied to the corresponding `/src/gen` subfolder in the project.

7.2.6.5 Run a Java Module

To create a running application in the runtime system, and resolve all dependencies defined in `mta.yaml`, run each Java module in your project.

Prerequisites

Before running a module, make sure that:

- All the relevant dependencies of the module are defined in the MTA descriptor.
- All the modules and resources on which the module is dependent are implemented and available (HDB modules are built).

i Note

You don't need to build your module explicitly before running, because it will be built automatically during the run process. The result of that build will be available in your workspace in the `target` folder, either as a `.war` or `.jar` file, depending on the type of your Java module.

Procedure

From the module context menu, choose *Run* and select one of the following options:


Option	Description
Run Configurations	Create a new or modify an existing run configuration. In the <i>Browser Window</i> tab, you can optionally choose a service path for your module to open in a new browser window. For example, choose <i>/hello</i> for the Hello World sample servlet generated by the wizard
Run as Java Application	Run the <code>.war</code> or <code>.jar</code> archive that was last built by Maven.

Note

If you have an MTA extension file with the name `dev.mtaext` that extends the `mta.yaml` file in the MTA project root, its contents will be merged with the `mta.yaml` file when running the module.




7.2.6.5.1 Use the Run Console

Progress and status messages, generated during the run process, as well as the log records, are displayed in the Run console.

The console opens automatically in the main browser tab. You can show and hide the console by clicking the  (*run console*) icon in the bottom pane's toolbar.

The left pane of the console displays a list of all running modules in the current MTA project along with the corresponding run configurations. Each entry is accompanied by a status icon with a tooltip. Click the run configuration name below a module name to view its most recent run log.

The controls displayed at the top of the console enable you to perform the following tasks:




- If the application URL is visible, click it to access the running application in the browser.
- To view the detailed server log of a run process, click *Logs*.
- To clear the log in the console, click  (*clear the log*).
- To stop running the module, click  (*stop*).
This causes the module to be removed from the runtime system, so that if you run the module again, the run process starts from the beginning.
- To run the module using a different configuration, select the configuration and choose the  (*run*) button in the task bar.

All the log and error messages of a running Java application are immediately displayed in the Run console.

7.2.6.6 Test a Java Module

Build and test your Java application with unit tests.

- Add unit tests in the `test` folder structure.
- Execute your tests. In the context menu from your module select *Build and Run Tests*.
- Review the test results.

Each Java module must provide own its JUnit test. The Java module template comes with a sample JUnit test. Use this test to get familiar with the test functionality and the  (*test results*) pane. By default, the  (*test results*) pane shows *Test Results*, *Stack Trace*, and *Coverage* for the module that is selected in the workspace. Also by default, the code coverage is shown in the editor, which enables you to write targeted unit tests for uncovered code. To disable this function, edit your  (*settings*).


If you do not see coverage results in the test pane, open the `pom.xml` file, and make sure the following code is contained in the `build` section.



Sample Code

```
<plugin>
  <groupId>org.jacoco</groupId>
  <artifactId>jacoco-maven-plugin</artifactId>
  <version>0.7.9</version>
  <executions>
    <execution>
      <id>default-prepare-agent</id>
      <goals>
        <goal>prepare-agent</goal>
      </goals>
    </execution>
    <execution>
      <id>jacoco-report</id>
      <phase>test</phase>
      <goals>
        <goal>report</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

7.2.6.7 Debug Java Modules

You can debug your Java modules using the tools in the Debugger pane.

If you have enabled debugging in the run configuration, and set breakpoints in your `.java` files, the  (Debugger) pane will open as soon as the module is up and running. In this pane, you can perform regular debugging tasks, such as viewing the call stack, examining the variables, stepping in and out of the functions, and more.

You don't have to run Java modules explicitly in debug mode. Instead, you can  (*attach*) to any running Java module in the  (Debugger) pane. Your Java module automatically switches to debug mode.

To debug Java module applications, you must verify the following:

- If the Java module packaging type is WAR, then make sure the following code is included in your model's root `pom.xml` file:

```

<profiles>
  <profile>
    <activation>
      <property>
        <name>devmode</name>
        <value>>true</value>
      </property>
    </activation>
    <build>
      <plugins>
        <plugin>
          <artifactId>maven-war-plugin</artifactId>
          <configuration>
            <webResources>
              <resource>
                <directory>${
{project.build.sourceDirectory}</directory>
                <targetPath>sources</targetPath>
              </resource>
            </webResources>
          </configuration>
        </plugin>
      </plugins>
    </build>
  </profile>
</profiles>

```

- If the Java module packaging type is JAR, then make sure the following code is included in your model's root `pom.xml` file:

Sample Code

```

<profiles >
  <profile>
    <activation>
      <property>
        <name>devmode</name>
        <value>>true</value>
      </property>
    </activation>
    <build>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-antrun-plugin</artifactId>
          <version>1.8</version>
          <executions>
            <execution>
              <phase>install</phase>
              <configuration>
                <tasks>
                  <unzip src="${
{project.build.directory}/${artifactId}-${version}.jar" dest="${
{project.build.directory}/tmp"/><mkdir dir="${project.build.directory}/tmp/
sources"/>
                  <copy todir="${
{project.build.directory}/tmp/sources"><fileset dir="${basedir}/src/main/
java"/>
                  </copy><zip basedir="${
{project.build.directory}/tmp" destfile="${project.build.directory}/${
artifactId}-${version}.jar"/>
                </tasks>
              </configuration>
            </execution>
          </executions>
        </plugin>
      </plugins>
    </build>
  </profile>
</profiles >


```

```

        <goals>
          <goal>run</goal>
        </goals>
      </execution>
    </executions>
  </plugin>
</plugins>
</build>
</profile>
</profiles>

```

Breakpoints

Breakpoint Status	Description
Enabled standard breakpoint	<p>Shape outline: blue</p> <p>Shape fill: light blue</p>
Disabled standard breakpoint	<p>Shape outline: blue</p> <p>Shape fill: transparent</p> <p>Breakpoints have been explicitly disabled. Disable breakpoints in the <i>Breakpoints</i> section by selecting the corresponding checkbox or deactivate globally by pressing the  (<i>deactivate breakpoints</i>) button.</p>
Checkmark	<p>Shape outline and fill are not affected.</p> <p>You set the breakpoint successfully on a running debugging session. The checkmark indicates that the corresponding source file has been loaded. As a consequence the application will suspend when it reaches the breakpoint.</p> <p>A breakpoint without a checkmark can mean that there is no debug session running, the source file has not been loaded, or the breakpoint could not be successfully resolved. An unresolved breakpoint may occur when the workspace resource version and the currently executed resource version don't match.</p>

Related Information

[Build a Java Module \[page 483\]](#)

[Run a Java Module \[page 483\]](#)

[Test a Java Module \[page 484\]](#)

7.2.6.8 Use Eclipse to Develop Java Modules


(Optional) You can develop and debug Java modules for your multi-target applications (MTA) in Eclipse, while performing all other development tasks in SAP Web IDE.

In SAP Web IDE

Task	Information
1. Create an MTA project.	Setting Up Application Projects [page 449]
2. To generate the data objects in the database, develop and build an SAP HANA Database module.	Developing SAP HANA Database (HDB) Modules [page 457]
3. Create a Java module.	Developing Java Modules [page 478]
4. Submit the module to a Git repository, or export it to the file system.	Using Source Control (Git) [page 363]
5. Switch to Eclipse IDE.	

In Eclipse IDE

Task	Information
1. Clone the Java module from the Git repository.	Choose Window > Show View > Git Repositories > Clone a Git Repository and add the clone to the view. Alternatively, you can import the exported module from the file system.
2. Import the Java project into Eclipse.	1. Choose Window > Show View > Project Explorer > Import > Existing Maven Projects . 2. In <i>Root Directory</i> , enter the folder to which you have cloned the project, and click <i>Finish</i> .
3. Write Java code, then build and run the Java application.	1. From the project context menu, choose Run As > Maven install . 2. Choose Run As > Run on Server .

Task	Information
4. Deploy the application to Cloud Foundry.	<ol style="list-style-type: none"> 1. In the <i>Select the Server</i> dropdown list, either choose an existing server, and then , or define the server manually. 2. In the <i>Add and Remove</i> dialog box, add the required resources to the <i>Configured</i> list, and click <i>Finish</i>. 3. In the <i>Application Details</i> dialog box, make sure that the details of your application are correct, and click <i>Next</i>. 4. In the <i>Launch Deployment</i> dialog box, enter the deployment details, and click <i>Next</i>. 5. Bind a service to the application. In the <i>Service selection</i> dialog box, select the <code>hdi_container</code> of the HDB module that you previously built in SAP Web IDE. The name has the following pattern: <code><project name>-hdihdb-<userid><workspace></code>. Click <i>Next</i>.
5. Add environment variable PATH that points to the WAR file.	<p>Add the following parameters to the <code>manifest.yml</code> file:</p> <ul style="list-style-type: none"> • <code>path</code> that points to the WAR file. • In the <code>env</code> section, <code>JBP_CONFIG_RESOURCE_CONFIGURATION</code> <p>For example:</p> <pre> ... path: /Users/<UserID>/workspace/mtad4/java4/target/ java4-0.0.1-SNAPSHOT.war env: JBP_CONFIG_RESOURCE_CONFIGURATION: '[tomcat/ webapps/ROOT/META-INF/context.xml: {"service_name_for_DefaultDB" : "<name of the hdi-container>"}]' </pre>
6. Submit the Java project to the same Git repository.	
7. Switch to SAP Web IDE.	

In SAP Web IDE

Task	Information
1. Clone the Java module from the Git repository, and import it to your MTA project.	Create a Java Module [page 480]
2. Develop an HTML5 module as required.	Developing HTML5 Modules [page 490]
3. Run and test the application, then deploy it to Cloud Foundry.	Packaging and Deploying Applications to Production Systems [page 506]

7.2.7 Developing HTML5 Modules

An HTML5 module is a collection of related HTML5 files that implement the user interface of your application.

SAP Web IDE supports the following HTML5 module types:

- SAPUI5 application - an SAPUI5 app including an optional view.
- CRUD master-detail module - displays data from an OData service using the master-detail pattern, provides CRUD operations.
- List Report application - based on the list report and object page SAP Fiori elements.

To develop HTML5 modules, perform the following steps:



- [Create an HTML5 Module \[page 491\]](#)
- [Implementing HTML5 Files \[page 491\]](#)
- [Inside an MTA Descriptor \[page 440\]](#)
- [Running Applications in Development Mode \[page 391\]](#)

Module Folder Structure

The following figure depicts a sample basic HTML5 module folder structure alongside the corresponding entry in the `mta.yaml`.



Folder	Description
<code><module name></code>	Contains the <code>xs-app.json</code> (application router configuration), and <code>package.json</code> (application router details and dependencies) files.
<code>resources</code>	Contains the HTML5 resource files, including the default <code>index.html</code> file.

⚠ Caution

Do not change the version of `approuter` in the `package.json` of the module.

Implementing HTML5 Files

In the `\resources` subfolder, create and implement the required HTML5 files. You can design the XML views using the SAP Web IDE layout editor.

For information about developing HTML5 apps with SAPUI5, see [SAPUI5: UI Development Toolkit for HTML5](#).

Related Information

[Layout Editor \[page 256\]](#)

[Developing Multitarget Applications \[page 438\]](#)

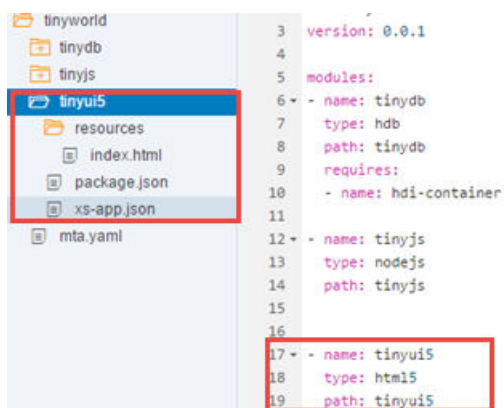
[Create List Report and Object Page Applications \[page 244\]](#)

7.2.7.1 Create an HTML5 Module

You can create new HTML5 modules.

Procedure

1. From the project context menu, choose **New** > **HTML5 Module**.
2. Depending on your application needs, choose one of the HTML5 module templates.
3. Enter the module name, and set the relevant module properties.
4. In the *Data Connection* wizard step, define a connection to the appropriate data source. For more information, see [Select a Data Source for an HTML5 Module \[page 492\]](#).
A new HTML5 module with the specified name is created in your project, and a corresponding section is added to the MTA descriptor (`mta.yaml`). For example:



i Note

The default `index.html` file of the new module implements a basic Hello World application.

[Select a Data Source for an HTML5 Module \[page 492\]](#)

Select a data source to access it from your application.

Related Information

[Developing HTML5 Modules \[page 490\]](#)

[Import an HTML5 Module \[page 498\]](#)

7.2.7.1.1 Select a Data Source for an HTML5 Module

Select a data source to access it from your application.

The following table describes how to define connections to the available data source types.

Data Source Type	Instructions
Current Project	Browse your project to locate and select a Java or Node.js module that exposes an OData service.
File System	Browse your file system to locate and select the required service <code>metadata.xml</code> document.
Workspace	Browse your workspace to locate and select the required service <code>metadata.xml</code> document.
Service Catalog	From the dropdown list, select a system to connect to the required service, and select this service from the Services table.
Service URL	From the dropdown list, select a system to connect to the required service, and enter the relative URL of this service.
SAP API Business Hub	From the dropdown list, select an API package containing the required service, and select this service from the table below.
SAP BTP Service	From the dropdown list, select the required SAP BTP Service OData service, and add the service to your project.

When you select a remote service, you are prompted to provide credentials for the requested system.

For more information, see [Select a Data Source \[page 95\]](#).

7.2.7.1.1.1 Create UI Based on Your OData Service

You can create UI based on an OData service in your current project.

Prerequisites

1. You created a Business Application Development tool for SAP BTP with a Java or Node.js module that exposes an OData service. See [Developing Business Applications \[page 504\]](#).
2. You selected **Build > Build CDS** by right-clicking on the project folder. For an application with a Node.js module, the `CatalogService.xml` file is then added to the `srv > gen` folder. For an application with a Java module, the `CatalogService.xml` file is then added to the `srv > src > main > resources > edmx` folder.

Procedure

1. Right-click the MTA folder, and select **New > HTML5 Module**.
2. Select the relevant template:
 - Select the *SAP Fiori Worklist Application OData V4* template when selecting a Node.js module that exposes an OData service.
 - Select any template that fits the OData version (V2 or V4) which is exposed by the Java module.
3. Provide the module name, title, and any other relevant information.
4. In the *Data Connection* step, select the desired Java or Node.js module that exposes an OData service from the *Current Project* list.
5. In the *Template Customization* step, select the relevant application settings and data binding information.

Results

The UI is created based on an OData service in your project.

In the example below, when creating a UI based on a Node.js module, the following content is added:

- The following new route is created in the `xs-app.json` file.

```
"routes": [  
  {  
    "source": "/srv_api/(.*)$",  
    "target": "$1",  
    "authenticationType": "xsuaa",  
    "destination": "srv_api",  
    "csrfProtection": false  
  }  
]
```

- The following data source and model are created in the `manifest.json` file:

```

        "dataSources": {
            "mainService": {
                "uri": "/srv_api/
catalog/",
                "type": "OData",
                "settings": {
                    "odataVersion": "4.0",
                    "localUri": "localService/metadata.xml"
                }
            }
        }

```

```

        "models": {
            "": {
                "dataSource":
                "mainService",
                "preload": true,
                "settings": {
                    "operationMode": "Server",
                    "groupId": "$direct",
                    "synchronizationMode": "None",
                    "autoExpandSelect": true
                }
            }
        }

```

- The HTML5 module name is added to the `mta.yaml` file and the `srv_api` is added to the `requires` section of the `appRouter` module if the HTML5 Application Repository is used.

```

- name: <MTA_name>_appRouter
  type: approuter.nodejs
  path: <MTA_name>_appRouter
  parameters:
    disk-quota: 256M
    memory: 256M
  requires:
    - name: <MTA_name>_html5_repo_runtime
    - name: srv_api
      group: destinations
      properties:
        forwardAuthToken: true
        name: srv_api
        url: '~{url}'

```

7.2.7.1.1.1 Run UI Based on Your OData Service

You can run your HTML5 module on Cloud Foundry based on a running Node.js or Java service.

Prerequisites

You created an HTML5 module based on an OData service in your current project. See [Create UI Based on Your OData Service \[page 493\]](#).

Procedure

1. Right-click the `db` folder and select **▶ Build ▶ Build ▶** to build tables.
2. Right-click the `srv` folder and run the desired application:
 - **▶ Run ▶ Run as Node.js Application ▶**
 - **▶ Run ▶ Run as Java Application ▶**

i Note

When creating UI from a Java module, there is no need to run the Java separately to preview the UI. Right-click the HTML5 module and select **▶ Run ▶ Run as Web Application ▶** to automatically run the Java service in the Cloud Foundry environment.

3. Run the HTML5 module in the Cloud Foundry environment. Right-click the HTML5 module and select **▶ Run ▶ Run as Web Application ▶** or see [Create Run Configurations \[page 394\]](#).

Results

The UI opens in a separate tab, enabling you to preview the application consuming your OData service in the Cloud Foundry environment.

7.2.7.1.1.2 Create UI Based on an SAP BTP Service

Developers can create an HTML5 module in their multitarget application that displays data from an SAP BTP Service.

Prerequisites

Make sure that you have a resource in your `mta.yaml` file. To add a resource, follow the steps in [Adding a Service Dependency to your MTA Project \[page 505\]](#).

Context

You can create a module using the sources described in [Select a Data Source for an HTML5 Module \[page 492\]](#). You can also use an SAP BTP Service as a data source.

Procedure

1. Right-click the MTA folder, and select **New > HTML5 Module**.
2. Select the desired HTML5 module template. For example, use the *List Report Application*.
3. Provide the module name, title, and any other relevant information.
4. In the *Data Connection* step, select **SAP BTP Service**.
5. Select the relevant service from the table.
6. Enter your credentials in the login page and return to the *Data Connection* step.
7. Select the relevant service from the table.
8. In the *Annotation Selection* step, select the required annotation file.
9. In the *Template Customization* step, select the relevant collection.

7.2.7.1.1.3 Create Destinations for Cloud Foundry Services

You can create a destination for a Cloud Foundry service from within the wizard for creating an HTML5 module within a multi-target application. The wizard automatically fills in many of the destination fields, and saves you the need to go to the SAP BTP cockpit to create the destination.

Context

To create a destination within the wizard, the Cloud Foundry service must already be deployed and running. Once you create the destination with the wizard, the service is available in the list of services under [Service URL](#).

Procedure

1. In the [Data Connection](#) step of the HTML5 Module wizard, select [configure a new destination](#).
2. Fill out the following fields:

Field	Description
Destination Name	A name for the destination
Global Subaccount ID	The ID of your global subaccount that contains your Neo environment
Global Subaccount Password	The password of your global subaccount that contains your Neo environment
Cloud Foundry Endpoint	The endpoint you received when signing up for a Cloud Foundry environment
Service Name	The name of your Cloud Foundry service
Cloud Foundry Org	The organization in which your Cloud Foundry service is running
Cloud Foundry Space	The space in which your Cloud Foundry service is running

3. Select [OK](#).

Results

The destination is now available in the list of services under [Service URL](#).

7.2.7.2 Import an HTML5 Module

You can import archived HTML5 modules.

Prerequisites

The `.zip` module archive that you want to import, which was exported from another MTA project, is available in the file system.

Procedure

1. From the root folder of the project, choose **File > Import > File or Project**.
2. Click *Browse* to locate and select your archive, and choose *Open*. The file name appears in the *File* field. The destination folder is displayed in the *Import to* field. To change this folder, choose *Select Folder*, and browse to the required folder, or create a new folder.
The specified folder, containing the artifacts extracted from the archive, is created in the project.
3. To make the imported folder a proper module in your project, you need to convert it into a module of the matching type. From the folder context menu, choose *Convert To*, and then the type of the target module.

i Note

The conversion process does not check whether the imported folder structure matches the selected module type. The process does not generate the module artifacts according to the selected type.

The imported module becomes a part of your MTA project, and the module entry is added to the MTA descriptor.

7.2.7.3 Run an HTML5 Module in the Cloud Foundry Environment

You can choose to run your HTML5 module in the Cloud Foundry environment and test its functionality.

Context

The run functionality is used to test and preview an application before deployment. One of the main limitations of testing an application is that the development environment is most likely different than the productive environment in terms of application dependencies, user roles, and more.

When you develop an HTML5 module in a multi-target application that is using the HTML5 Application Repository, you can choose to run this module on your Cloud Foundry space, in order to receive a true state of the module. For that purpose, when you set the run configuration, select the target environment to be Cloud Foundry.

Additional advantages of running an HTML5 module directly on the Cloud Foundry environment are:

- The HTML5 module contains its own app-router which is also being deployed to Cloud Foundry. This allows a complete user separation and it also means that if you have multiple HTML5 modules in your multi-target application project, it's enough to run one module in order to get the fast re-run capability in all other HTML5 modules.
- All required services are being automatically provisioned for you (e.g. Destination service, Connectivity service, etc.).
- Second run onwards, updates the new HTML5 module content in the running app-router.
- Changes made to key files in your multi-target application will trigger a new deployment and a restart of the app. Key files are:
 - `mta.yaml`
 - `package.json`
 - `xs-security.json`
 - `xs-app.json`
- In case the HTML5 module is dependent on a Java service, the later will be started upon the run of the app, if not already running.

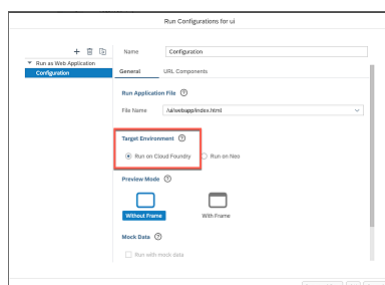
Procedure

1. From the *Preferences* perspective, choose *Cloud Foundry* and verify that you have already configured a space.
2. Create a multi-target application with the option to use the HTML5 Application Repository checked.

For more information, see [Using the HTML5 Application Repository in a Multitarget Application \[page 453\]](#)

3. Add an HTML5 module to the multi-target application.
4. Configure your run settings. In the *Target Environment* section, select *Cloud Foundry*.

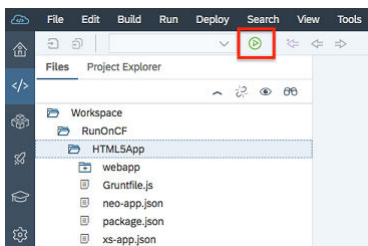
For more info, see [Create Run Configurations \[page 394\]](#)



Note

The default target environment for multi-target applications with HTML5 Application Repository is Cloud Foundry.

5. Select your HTML5 module and click on the green play button in the upper toolbar to run the app.



i Note

The first run of your app deploys it to SAP BTP, Cloud Foundry environment. This action may take up to 1-2 minutes. The next runs will take just a few seconds.

i Note

If you have an MTA extension file with the name `dev.mtaext` that extends the `mta.yaml` file in the MTA project root, its contents will be merged with the `mta.yaml` file when running the module.

When finished, a new browser tab opens, and the HTML5 module is launched from the Cloud Foundry environment!

7.2.7.4 Develop an SAPUI5 Library

You can create, build, deploy, and reuse an SAPUI5 library.

Using an SAPUI5 library enables you to share custom controls between SAPUI5 projects in SAP Web IDE.

The SAPUI5 library that you create includes a sample control and Grunt build capabilities based on the [UI5 Build and Development Tooling](#).

Creating an SAPUI5 Library

You create an SAPUI5 library by adding a new project for the Cloud Foundry environment or by adding an HTML5 module to your MTA project.

1. Add a new project or an HTML5 module.
 - To create an SAPUI5 library by adding a new project to your workspace, from the *File* menu, choose **New > Project from Template**.
 - To create an SAPUI5 library by adding a new HTML5 module to your MTA project, from the MTA project context menu, choose **New > HTML5 Module**.
2. In the *Template Selection* step, from the *Environment* dropdown list, select *Cloud Foundry* (if relevant for your template).
3. From the *Category* dropdown list, select *SAP Fiori Library*, and choose *Next*.
4. In the *Basic Information* step, enter a name for the module and choose *Next*.

5. In the *Template Customization* step, fill in the required fields for the library settings, make sure that the *Add sample control* checkbox is selected, and choose *Next* or *Finish*.

The new project that is created includes a deployer module, an appRouter module, and an SAPUI5 library, which includes a sample control (`Example.js` file).

Building and Deploying the SAPUI5 Library

You can build and deploy your SAPUI5 library as part of the process of building and deploying an MTA project. See [Packaging and Deploying Applications to Production Systems \[page 506\]](#).

The deployer module is mandatory for deploying the content of the SAPUI5 library to the HTML5 Application Repository.

The appRouter module is only required if you develop SAPUI5 applications in the project, in addition to SAPUI5 libraries. If you are only deploying SAPUI5 libraries, you can delete the appRouter module. See [Using the HTML5 Application Repository in a Multitarget Application \[page 453\]](#).

Consuming your SAPUI5 Library

1. Create an SAPUI5 application. See [Create an HTML5 Module \[page 491\]](#).
2. Add a reference to your SAPUI5 library in your SAPUI5 application's `manifest.json` file:
 1. From the workspace, open your SAPUI5 application's `manifest.json` file.
 2. Search for the `"libs"` property in the `"sap.ui5" > "dependencies"` section.
 3. Add your SAPUI5 library to the `"libs"` property by adding the namespace followed by a period (`.`), and then the ID of the library.

```
"sap.ui5": {
  ...
},
"dependencies": {
  "minUI5Version": "1.60.1",
  "libs": {
    "sap.ui.core": {},
    "sap.m": {},
    "sap.ui.layout": {}
    "testnamespace.TestLibrary": {}
  }
}
```

3. Add a reference to a control from your SAPUI5 library to your SAPUI5 application's `view.xml` file:
 1. From the workspace, open your SAPUI5 application's `view.xml` file.
 2. Add a reference to a control from your SAPUI5 library by adding `<xmlns:lib="namespace.libraryID.controls">`.

```
<mvc:View controllerName="TestNameSpace.SAPUI5TestModule.controller.View1"
xmlns:mvc="sap.ui.core.mvc" displayBlock="true" xmlns="sap.m">
xmlns:lib="testnamespace.TestLibrary.controls">
  <Shell id="shell">
    <App id="app">
      <pages>
        <Page id="page" title="{i18n>title}">
```

```

        <content>
            <lib:Example text= "Hello world!"></lib:Example> </
lib:Example>
        </content>
    </Page>
</pages>
</App>
</Shell>
</mvc:View>

```

4. Add the SAPUI5 library to your SAPUI5 application's `index.html` file:

1. From the workspace, open your SAPUI5 application's `index.html` file.
2. Add the SAPUI5 library to the `data-sap-ui-resourceroots` property:

```
"namespace.libraryID": "../namespacelibraryID-libraryversion"
```

```

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>SAPUI5TestModule</title>
    <script id="sap-ui-bootstrap"
      src="https://sapui5.hana.ondemand.com/resources/sap-ui-core.js"
      data-sap-ui-theme="sap_belize"
      data-sap-ui-resourceroots='{"ns.consumingApp": "./",
"testnamespace.TestLibrary": "../testnamespaceTestLibrary-1.0.0"}'
      data-sap-ui-compatVersion="edge"
      data-sap-ui-oninit="module:sap/ui/core/ComponentSupport"
      data-sap-ui-async="true"
      data-sap-ui-frameOptions="trusted">

```

5. Run the consuming SAPUI5 application.

7.2.7.5 Deploy HTML5 Modules

After making changes to HTML5 modules in a multitarget application, you can deploy all HTML5 modules to see the changes. You can see your updated HTML5 modules more quickly than you would if you build and deploy the entire multitarget application.

Prerequisites

You have previously deployed your multitarget application. This ensures that the required service already exist.

Using the HTML5 Repository instance of the `app-host` service plan is a requirement for deploying HTML5 modules. The following is a sample of code from the `mta.yaml` file:

```

resources:
- name: mtalist_html5_repo_host
  parameters:
  service-plan: app-host
  service: html5-apps-repo
  type: org.cloudfoundry.managed-service

```

Procedure

1. Log on to Cloud Foundry through the [Project Settings](#).
2. In the workspace, right-click the project folder and choose **Deploy > Deploy HTML5 Modules**. All HTML5 modules in your multitarget application are deployed as content to the HTML5 Repository, using the app-host instance in the `mta.yaml` file. Even if you only make changes to one HTML5 module, all of the HTML5 modules in your multitarget application are deployed.

The changes made to the UI of your application, deployed on Cloud Foundry, can be viewed from the launchpad or in a separate browser window.

Note

If you have GACD-based content launchpad modules, UI changes in the `manifest.json` and `i18n` files won't occur after HTML5 deployment. If you have older application-based launchpad modules, the UI changes in the `manifest.json` and `i18n` files will occur.

To determine which launchpad module type is in your application, look at the value of the `modules > type` field in the `mta.yaml` file:

Module Type	Value	HTML5 Deployment Supported?
GACD-based content launchpad modules (new)	<code>com.sap.application.portal.content</code>	No
Application-based launchpad module (old)	<code>com.sap.portal.content</code>	Yes

See [Add an SAP Fiori Launchpad Module](#) for more information.

7.2.8 Developing SAP S/4HANA Service Extensions

SAP Web IDE enables you to create S/4 HANA extensions. You implement an application with an OData service that extends an existing S/4HANA service and exposes additional data from a different source, such as an SAP HANA database.

You can do this using the Business Application Development tools for SAP BTP.

Prerequisites

You have enabled the following features (extensions):

- SAP HANA Database Explorer
- Business Application Development tools for SAP BTP

For more information, see [Enable SAP Web IDE Extensions \[page 510\]](#).

Workflow

1. Generate an SAP Business Application in SAP Web IDE. See [The Application Programming Model](#).
2. Set up the SAP S/4HANA Cloud SDK in the Application Programming Model.
3. Define the CDS service model.
4. Add custom handlers to call the SAP S/4HANA System.
5. Build and deploy to Cloud Foundry.

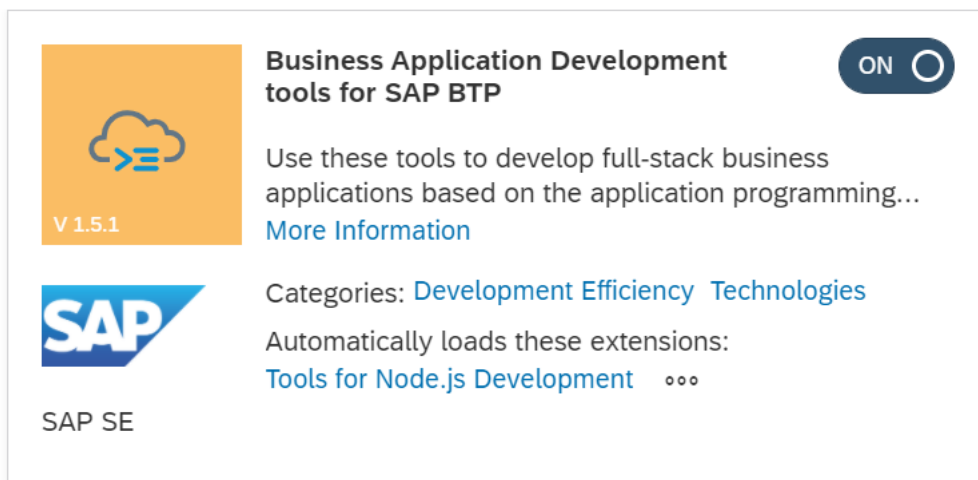
For more information, see [this blog post](#).

7.2.9 Developing Business Applications

SAP Web IDE enables you to develop business applications using the application programming model for SAP BTP.

The Business Application Development tools for SAP BTP contain a new project template, and help you implement data models, services, and UIs to develop your own standalone business applications or extend other cloud solutions, such as SAP S/4 HANA.

To use this extension, you need to enable the tile in the *Preferences* perspective on the *Extensions* page.



The screenshot shows a user interface for managing extensions. At the top, there is a toggle switch labeled 'ON' with a white circle, indicating the extension is active. Below this, the extension is titled 'Business Application Development tools for SAP BTP'. To the left of the title is an orange square icon with a white cloud and a blue '>E' symbol, and the version 'V 1.5.1' is displayed below it. The main text describes the extension: 'Use these tools to develop full-stack business applications based on the application programming...' followed by a 'More Information' link. Below this, it lists categories: 'Development Efficiency Technologies'. Underneath, it states 'Automatically loads these extensions:' followed by 'Tools for Node.js Development' and a three-dot menu icon. At the bottom left, the SAP logo is shown above the text 'SAP SE'.

For more information, see [Business Applications](#) in the SAP BTP documentation.

7.2.10 Adding a Service Dependency to your MTA Project

The Business Application Development tools for SAP BTP extension allows you to add a service dependency to your MTA project.

Prerequisites

Make sure the required service is assigned to your Cloud Foundry space in the SAP BTP cockpit otherwise you will not be able to access it.

Procedure

1. In SAP Web IDE, enable the **Business Application Development tools for SAP BTP** extension. See [Enable SAP Web IDE Extensions \[page 510\]](#).
2. Create a multi-target application that uses the HTML5 Application Repository. See [Using the HTML5 Application Repository in a Multitarget Application \[page 453\]](#).
3. From the context menu of the project folder, select **New > SAP BTP Service**.

The wizard opens.

4. In the *Service Selection* step, from the *Category* dropdown list, select the desired service type.
5. Select the relevant service from the list of available services and choose *Next*.
6. In the *Service Definition* step, use an existing service instance or create a new service instance.

- **Use an existing service instance:**

- a. Select *Reuse instance*.
- b. Select the required service instance from the *Service Name* list.
- c. In the *Resource Name* field, enter a name that will appear in the `mta.yaml` file.

A new resource is added to your `mta.yaml` file.

- **Create a new service instance:**

- a. Select *New Instance*.
- b. Select the required plan from the *Plan* list.
- c. In the *Service Name* field, enter a name that will appear in the `mta.yaml` file and choose *Next*.
- d. (Optional) In the *Parameter Selection* step, add a row for each configuration parameter that you want to add to your service instance. Provide a *Key* and a *Value*. When you've added all of the parameters that you want, choose *Next* or *Finish*.

A new service instance is created in the space assigned to your MTA project and a new resource is added to your `mta.yaml` file.

```
- name: xsuaaNewServiceInstance
  parameters:
    service-name: xsuaaNewServiceInstance
    service: xsuaa
    service-plan: default
  config:
```

```
xsappname: 'my-xsuaa-appname '  
tenant-mode: shared  
type: org.cloudfoundry.managed-service
```

7.2.11 Packaging and Deploying Applications to Production Systems

At the last stage of multi-target application (MTA) development, you need to package your application and deploy it to a target production system. You can only deploy multitarget applications to the SAP BTP, Cloud Foundry environment.

Procedure

1. From your project context menu, choose **Build** > **Build with Cloud MTA Build Tool**. The resulting MTA archive file (<ID>_<version>.mtar file) is generated within the project's *mta_archives* folder.

i Note

If you migrated to the *Build with Cloud MTA Build Tool* from the previous build menu option, make sure you adjusted your project as described [here](#).

2. Right-click the <ID>_<version>.mtar file, and choose **Deploy** > **Deploy to SAP BTP**.
3. In the **Deploy to SAP BTP** dialog box that opens, select the space in the target system where you want to deploy your application.
4. If your project contains MTA extension files, select the relevant checkboxes from the dropdown list.
5. Click *Deploy*.

7.2.11.1 Updating Your MTA Project's Grunt Build to the UI5 Build

This topic provides instructions for updating your MTA project's Grunt build to the UI5 build.

Procedure

1. Perform the steps in [Updating Your Project's Grunt Build to the UI5 Build \[page 408\]](#).
2. Update all of the HTML5 modules in the `mta.yaml` file of the MTA project in the following way:

Before updating your HTML5 module, this is how it appears in the `mta.yaml` file:

```
- name: <modulename>
```

```
type: html5
path: <modulename>
build-parameters:
  builder: grunt
  supported-platforms: []
  build-result: dist
```

After updating your HTML5 module, this is how it appears in the `mta.yaml` file:

```
- name: <modulename>
  type: html5
  path: app/<modulename>
  build-parameters:
    builder: custom
    commands:
      - npm run build
    supported-platforms: []
```

8 SAP Web IDE Extensions

You can enable extensions in SAP Web IDE to extend the existing functionality.

i Note

This is not available in SAP Web IDE personal edition.

Extensions are the building blocks of SAP Web IDE. They are used to group functionality into small units.

Extensions can expose services to provide public APIs. Extension code may include any SAP Web IDE component: a new command, template, editor, pane, or any other contribution to the SAP Web IDE application.

The functionality provided by these extensions can be enabled or disabled from the *Preferences* perspective. For more information, see [Enable SAP Web IDE Extensions \[page 510\]](#).

The list below shows a selection of our most used extensions.

Feature	Description
SAP Mobile Service	You can use these services to build end-to-end mobile applications. See Mobile Services App Development Tools on SAP Web IDE
SAP BUILD	You can create a project in SAP Web IDE that will generate code from an SAP BUILD prototype to build a real application. See Extend Prototypes with SAP Web IDE
Business Application Development tools for SAP BTP	You can develop full-stack business applications based on the application programming model for SAP BTP. See Business Applications
SAP Cloud Portal service	You can create a site template to be used by the SAP Cloud Portal service administrator. See SAP Cloud Portal service .
SAP Workflow	You can model and deploy workflows that help in automating process steps. <div data-bbox="502 1592 593 1628" data-label="Section-Header"><h3>i Note</h3></div> <div data-bbox="502 1646 1342 1713" data-label="Text"><p>The Workflow editor is available only in regions where the SAP Workflow service is offered.</p></div> See SAP Workflow .

Feature	Description
SAP EIM Smart Data Integration Editors	<p>You can create SAP HANA smart data integration replication tasks that access data from supported sources and persist it in SAP HANA database tables. You can also create flowgraphs to perform transformations such as pivoting tables, capturing changed data, and comparing tables.</p> <p>See Modeling Guide for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality</p>
SAP Enterprise App Modeler	<p>You can develop next generation App Modeler applications.</p> <p>See SAP Enterprise App Modeler</p>
SAP Event Management	<p>You can generate your own transactional SAP Fiori apps for SAP Event Management.</p> <p>See SAP Event Management.</p>
SAP Fiori Launchpad Extensibility	<p>You can extend the shell of the SAP Fiori launchpad by creating your own SAP Fiori launchpad plugins.</p> <p>See SAP Fiori Launchpad Extensibility</p>
SAP HANA Database Explorer	<p>The database explorer is integrated into SAP Web IDE and allows you to execute SQL statements and database procedures, query information about the database, and view information about database catalog objects.</p> <p>See SAP HANA Database Explorer</p>
SAP Leonardo IoT	<p>You can build new IoT-related applications and customize them by using predefined components and templates.</p> <p>See SAP Leonardo IoT</p>
SAP MII	<p>You can import Web artefacts from the SAP MII system to SAP Web IDE. You can modify the imported artefacts, create new artefacts and sync them back to the SAP MII system. IntelliSense of SAP MII libraries is available in SAP Web IDE for the JavaScript files.</p> <p>See SAP MII Feature for SAP Web IDE</p>
SAP Translation Hub	<p>You can translate your project's i18n.properties file using SAP Translation Hub.</p> <p>See Use SAP Translation Hub [page 113].</p>
SAP Web IDE Hybrid App Toolkit	<p>You can create hybrid apps (also known as Kapsel apps) using Apache Cordova and the SAP Mobile Platform SDK.</p> <p>See SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments.</p>
Visualization Extension (Vizpacker)	<p>You can use the Visualization Extension plugin (Vizpacker) in SAP Web IDE to create chart extension packages that can be used within SAP Lumira and other products.</p> <p>See Visualization Extension (VizPacker) Plugin for SAP Web IDE</p>

Feature	Description
Fact Sheet Editor	<p>You can create fact sheets from scratch and edit existing fact sheets using a drag-and-drop approach.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>i Note</p> <p>This feature has been deprecated.</p> <p>See Fact Sheet Editor.</p> </div>

i Note

If you want to use the same class for several extensions, you must:

1. Define the desired class as an SAP Web IDE service for the first extension.
2. Consume the class as a service in the other extensions.

For more information about extensions, see the [SAP Web IDE SDK](#).

8.1 Enable SAP Web IDE Extensions

You can enable SAP Web IDE extensions to use in application development.

Context

SAP Web IDE includes extensions that are not enabled by default.

⚠ Caution


An SAP Web IDE extension extends the functionality of SAP Web IDE and provides new capabilities to your IDE. Such extensions) have full privileges to access your browser, your computer, and any data stored in your SAP Web IDE workspace or on SAP BTP, including the ability to read and modify your private and organizational data.

Extensions that are not provided by SAP are under the responsibility of the extension author and may have different privacy policies, terms of use, or quality levels. You can enable extensions that are not provided by SAP and use them at your own risk. It is strongly recommended that you enable only extensions that you trust. At any time and without warning, SAP reserves the right to remove, disable, or uninstall extensions that are not provided by SAP from your environment.

To learn more about extension development, see the [SAP Web IDE SDK](#).

If you would like to use extensions, perform the following steps:

Procedure

1. To open the *Preferences* perspective, in the left sidebar, choose  (Preferences).
2. Under *Workspace Preferences*, choose *Extensions*.
3. In the *Extensions* pane to the right, select the toggle button for the extension you want to enable.
4. Choose *Save*.
5. Refresh your browser.

Results

You can now use the enabled extensions in your projects.

9 Extending SAP Web IDE

SAP Web IDE architecture allows developers to easily extend SAP Web IDE functionality by developing custom extensions and templates.

You can learn about extension development in the SAP Web IDE software development kit (SDK), which you can access from:

- The [Help](#) menu in SAP Web IDE.
- [SAP Web IDE SDK](#).

10 SAP Web IDE Personal Edition

Customers using SAP Web IDE on SAP BTP benefit from the full scope of SAP Web IDE and leverage SAP BTP capabilities with frequent updates of the cloud environment. In addition we offer a personal edition of SAP Web IDE.

SAP Web IDE personal edition is intended as a complementary IDE, to be installed by a single developer on a personal workstation, for offline development (not on a server).

You can use it in one of the following ways:

- Trial - for test and evaluation purposes for anyone.
- Productive
 - For customers with a paid subscription to SAP Web IDE.
 - For customers with a paid subscription to SAP Business Application Studio, only for extending legacy applications which you cannot be extended in SAP Business Application Studio.

i Note

The personal edition is updated periodically and may not include extensions of SAP Web IDE on SAP BTP.

The personal edition includes the possibility to sync with Git on SAP BTP when needed or use a local Git repository.

When switching work modes from the cloud edition to the personal edition (or from the personal edition to the cloud edition), you can use Git or you can export the relevant project and import it in the other edition.

The following features are **not** available if you are using SAP Web IDE personal edition:

- SAP Cloud features:
 - Deploying to SAP BTP
 - Running applications on SAP BTP
 - Extending applications on SAP BTP
 - Importing applications from SAP BTP
 - Registering to SAP Fiori launchpad on SAP BTP
- Proxy Authentication
- Extension development
- Template development
- Use of optional extensions

i Note

The SAP Fiori Overview Page extension is enabled by default in the personal edition.

- Git stash option
- Code check before Git push
- Problem view
- Run configurations advanced settings
- Learning Center perspective

- Notification alerts
- Customizing performance measures
- Client build
- Grunt build
- Translation Hub

10.1 Installation and Setup

The following is an overview of the process for installing and starting SAP Web IDE personal edition.

Context

i Note

SAP Web IDE personal edition is meant for use by an individual user only and should not be used as a server.

Related Information

[Install SAP Web IDE Personal Edition \[page 514\]](#)

[Start SAP Web IDE Personal Edition \[page 518\]](#)

[Upgrade SAP Web IDE Personal Edition \[page 520\]](#)

10.1.1 Install SAP Web IDE Personal Edition

Instructions for installing SAP Web IDE personal edition.

Prerequisites

- You are installing SAP Web IDE personal edition in one of the following operating systems:
 - Microsoft Windows (version 7 or higher)
 - Mac OS (version 9 or higher)
- You have installed Java™ Platform, Standard Edition Runtime Environment (JRE) Version 7 (at least version 1.7) or Version 8 in the 64-bit version. If required, download an installer from [Java SE Download](#) and follow the instructions.

i Note

You can check the version by entering `java -version` in the command shell (Microsoft Windows[®]) or Terminal window (Mac OS[®]).

Output example:

```
java version "1.7.0_55"  
Java(TM) SE Runtime Environment (build 1.7.0_55-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 24.55-b03, mixed mode)
```

If you are using a higher Java version, you can set the personal edition to run with Version 7 or 8 by specifying this in the `orion.ini` file. For example:

≡ Sample Code

```
-vm  
<C:\Program Files\Java\jdk1.8.0_101\bin>  
-vmargs ...
```

Procedure

1. Go to [SAP Development tools](#).
2. Select the [SAPUI5](#) tab and scroll down to **SAP BTP Web IDE personal edition**.
3. Download the installation ZIP file.
4. Extract the zipped files.

i Note

The downloaded ZIP file includes SAP Web IDE personal edition and Orion 8.

Microsoft Windows[®]

Extract the zipped files to `C:\SAPWebIDE`.

i Note

- Due to the long file names, you might have problems extracting the files. Make sure you use a ZIP program that can handle long file names.
- Your workspace will be saved in the `serverworkspace` folder created. Be sure to backup this folder regularly.

Mac OS[®]

Extract the zipped files to `/Applications/SAPWebIDE`.

i Note

If you have installed Mac OS[™] Sierra, you need to perform some post-installation steps. For more information, see [Post Installation Troubleshooting for Mac Sierra Users \[page 516\]](#).

5. Start SAP Web IDE personal edition. For more information, see [Start SAP Web IDE Personal Edition \[page 518\]](#).

Steps required to run SAP Web IDE personal edition on Mac Sierra systems.

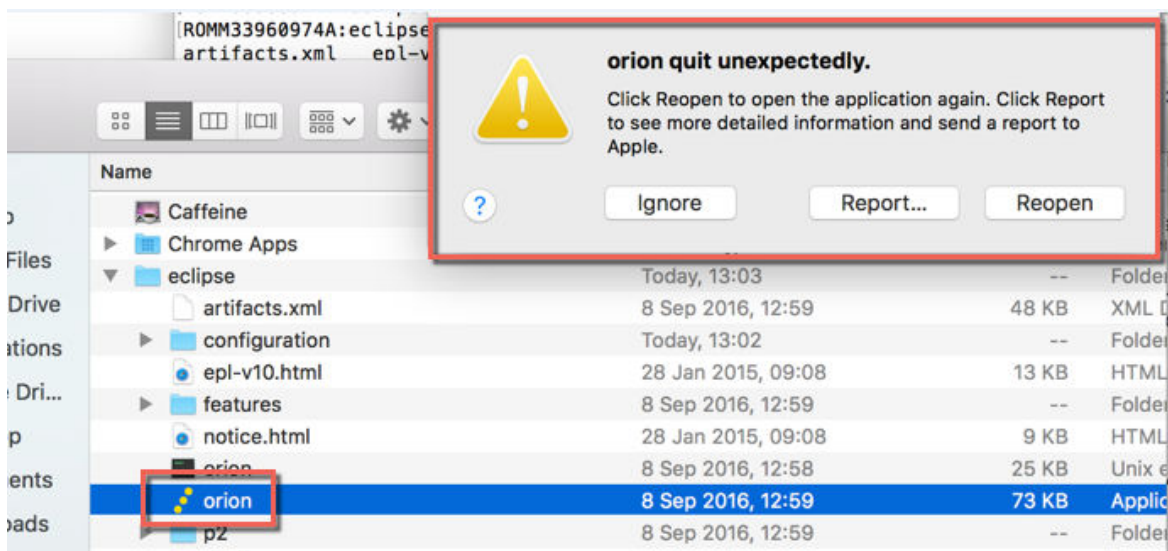
10.1.1.1 Post Installation Troubleshooting for Mac Sierra Users

Steps required to run SAP Web IDE personal edition on Mac Sierra systems.

Context

If you are working on a Mac Sierra system, you might find the following problems:

1. The app crashes when you try to run it by double-clicking the Orion icon. The following message is displayed:



Solution

Run the command `xattr -r -c *` inside the `eclipse` folder.

```

eclipse — -bash — 136
[ROMM33960974A:eclipse i045523$ ls -la
total 224
drwxr-xr-x@ 13 i045523 1694527156 442 Sep 23 13:27 .
drwx----- 17 i045523 1694527156 578 Sep 23 13:20 ..
-rw-r--r--@ 1 i045523 1694527156 6148 Sep 23 13:27 .DS_Store
-rw-r--r--@ 1 i045523 1694527156 47603 Sep 8 12:59 artifacts.xml
drwxr-xr-x@ 5 i045523 1694527156 170 Sep 23 13:19 configuration
-rw-r--r--@ 1 i045523 1694527156 12638 Jan 28 2015 epl-v10.html
drwxr-xr-x@ 17 i045523 1694527156 578 Sep 8 12:59 features
-rw-r--r--@ 1 i045523 1694527156 9013 Jan 28 2015 notice.html
-rwxrwxrwx@ 1 i045523 1694527156 25160 Sep 8 12:58 orion
drwxr-xr-x@ 3 i045523 1694527156 102 Sep 8 12:59 orion.app
drwxr-xr-x@ 4 i045523 1694527156 136 Sep 8 12:59 p2
drwxr-xr-x@ 190 i045523 1694527156 6460 Sep 23 13:20 plugins
drwxr-xr-x@ 3 i045523 1694527156 102 Sep 23 13:19 readme
[ROMM33960974A:eclipse i045523$ xattr -r -c *
[ROMM33960974A:eclipse i045523$ ls -la
total 224
drwxr-xr-x@ 13 i045523 1694527156 442 Sep 23 13:27 .
drwx----- 17 i045523 1694527156 578 Sep 23 13:20 ..
-rw-r--r--@ 1 i045523 1694527156 6148 Sep 23 13:27 .DS_Store
-rw-r--r--@ 1 i045523 1694527156 47603 Sep 8 12:59 artifacts.xml
drwxr-xr-x@ 5 i045523 1694527156 170 Sep 23 13:19 configuration
-rw-r--r--@ 1 i045523 1694527156 12638 Jan 28 2015 epl-v10.html
drwxr-xr-x@ 17 i045523 1694527156 578 Sep 8 12:59 features
-rw-r--r--@ 1 i045523 1694527156 9013 Jan 28 2015 notice.html
-rwxrwxrwx@ 1 i045523 1694527156 25160 Sep 8 12:58 orion
drwxr-xr-x@ 3 i045523 1694527156 102 Sep 8 12:59 orion.app
drwxr-xr-x@ 4 i045523 1694527156 136 Sep 8 12:59 p2
drwxr-xr-x@ 190 i045523 1694527156 6460 Sep 23 13:20 plugins
drwxr-xr-x@ 3 i045523 1694527156 102 Sep 23 13:19 readme
[ROMM33960974A:eclipse i045523$

```

2. Problems closing Orion.

Currently, you can only close Orion using *Force Quit*.

Solution for closing using the command line

1. Edit the `orion.ini` file located under the folder `orion.app/Contents/MacOS/`.
2. Add the following code just after the `-vmargs` statement.

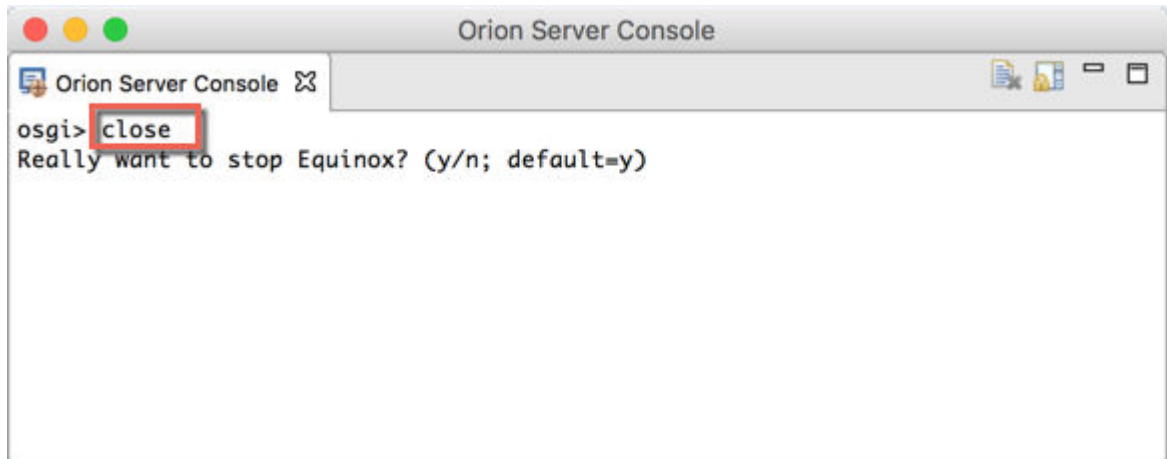
Sample Code

```

<pre class="language-javascript"><code>-product
org.eclipse.orion.server.ui.console -application
org.eclipse.orion.server.ui.consoleApp </code></pre>

```

3. Save the file and restart Orion.
4. Close Orion, by typing the command *Close* directly in the Orion console.



Task overview: [Install SAP Web IDE Personal Edition \[page 514\]](#)

10.1.2 Start SAP Web IDE Personal Edition

Follow the procedure below to open SAP Web IDE personal edition.

Procedure

1. Since the server is part of the installation, you must first open Orion. Start the Orion Application Server as follows:

i Note

The default port is 8080. If you want to use a different port, see [Configure the Orion Application Server \[page 522\]](#).

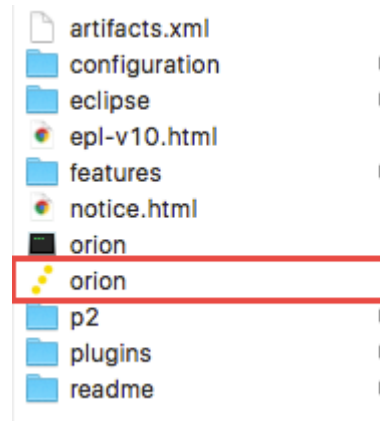
Microsoft Windows®

1. Go to `C:\SAPWebIDE` and open the `eclipse` folder.
2. Double-click the `orion.exe` file.

A command shell opens.

Mac OS®

1. In the Finder, go to `/Applications/SAPWebIDE/eclipse`
2. Hold the `Ctrl` key and click the `orion` file.



3. From the context menu, select *Open*.

i Note

If prompted, you should confirm the use of Orion.

i Note

If you have installed macOS™ Sierra, you need to perform some post-installation steps. For more information, see the blog post: [Post-Installation tips for SAP Web IDE Personal Edition on MAC](#).

2. Access SAP Web IDE personal edition via the URL `http://localhost:8080/webide/index.html`. The default port is 8080. If you configured a different port, you must change it in the URL accordingly.

i Note

When you start the Orion Application server for the first time (after initial installation or upgrade), you have to create a new account for it. (In the future this might be done automatically.)

1. Choose *Create a new account*.
2. Enter a user name and password.

SAP Web IDE personal edition does not support more than one account.

3. On the Orion Server Application logon page, enter the user and password that you defined.

10.1.3 Upgrade SAP Web IDE Personal Edition

You can manually upgrade to the newest version of SAP Web IDE personal edition.

Procedure

1. In your current SAP Web IDE personal edition version, backup the following files and folders:
 - `orion.ini` file: This file holds proxy information and has to be recreated during each installation.
 - `orion.conf` file: This file is used when the Orion configuration is changed. For example, when setting up SSL or changing the port.
 - `destinations` folder: This folder contains the destination you have set up.
 - `serverworkspace` folder: This is the folder where all the users and their workspaces are stored.
2. Remove the SAP Web IDE personal edition folders and files in `C:\SAPWebIDE` in Windows systems or in `/Applications/SAPWebIDE` in Mac systems.
3. Install the latest version of SAP Web IDE personal edition according to the instructions in the topic [Installation and Setup \[page 514\]](#).

i Note

If you have installed Mac Sierra, you need to perform some post-installation steps. For more information, see [Post Installation Troubleshooting for Mac Sierra Users \[page 516\]](#).

4. Restore the following files and folders:
 - `orion.ini` file. If you made changes to this file since your first installation, you must append them to the new version of the file.
 - `orion.conf` file.
 - `destinations` folder. Restore the folder to the following path: `config_master/service.destinations/destinations`.
 - `serverworkspace` folder. Restore to the `eclipse` folder.

Next Steps

If not previously defined, you can perform the optional settings that are described in the following sections:

- [Configure the Orion Application Server \[page 522\]](#)
- [Connect Remote Systems in SAP Web IDE Personal Edition \[page 523\]](#)
- [Import the Git Server Certificate into the JVM \[page 521\]](#)

10.1.4 Uninstall SAP Web IDE Personal Edition

Instructions for uninstalling SAP Web IDE, personal edition.

Procedure

1. In your current SAP Web IDE personal edition version, backup the following files and folders:
 - `orion.ini` file: This file holds proxy information and has to be recreated during each installation.
 - `orion.conf` file: This file is used when the Orion configuration is changed. For example, when setting up SSL or changing the port.
 - `destinations` folder: This folder contains the destination you have set up.
 - `serverworkspace` folder: This is the folder where all the users and their workspaces are stored.
2. Remove the SAP Web IDE personal edition folders and files in `C:\SAPWebIDE` in Windows systems or in `/Applications/SAPWebIDE` in Mac systems.

10.2 Import the Git Server Certificate into the JVM

If the server certificate is based on a company internal root certificate, you have to import the root certificate into the JVM.

Context

If you are connecting to a Git server in your company via HTTPS, the Java Virtual Machine (JVM) on which the Orion installation is running has to trust the server certificate. If the server certificate is not issued by a public agency, but is based on a company internal root certificate, you have to import the root certificate into the JVM. Otherwise the verification of the Git server's certificate fails and prevents you from performing Git operations.

To import your certificate into the JVM, fetch the certificate * .cer file from your company's IT department and store it on your machine, as follows:

Microsoft Windows*

1. In the command shell, enter the following command:
`cd C:\Program Files\Java\jre<VERSION>\lib\security`

Example:

```
C:\Program Files\Java\jre7\lib
\security
```

2. Next, enter the following command:
`keytool -import -file <PATH OF THE .cer FILE> -keystore cacerts -alias <ANY ALIAS>`

Mac OS*

1. In the Terminal window, enter the following command:
`cd /Library/Java/JavaVirtualMachines/jdk<VERSION>.jdk/Contents/Home/jre/lib/security`

Example:

```
/Library/Java/JavaVirtualMachines/
jdk1.7.0_65.jdk/Contents/
Home/jre/lib/security
```

2. Next, enter the following command:
`sudo keytool -import -file <PATH OF THE .cer FILE> -keystore cacerts -alias <ANY ALIAS>`

Note

If the keystore is protected by a password, ask your company's IT department for the password.

10.3 Configure the Orion Application Server

You may wish to configure the Orion Application Server for your system landscape.

Context

By default, the Orion Application Server runs on port 8080. You can change the port, for example, if there are conflicts with other servers running on the same machine.

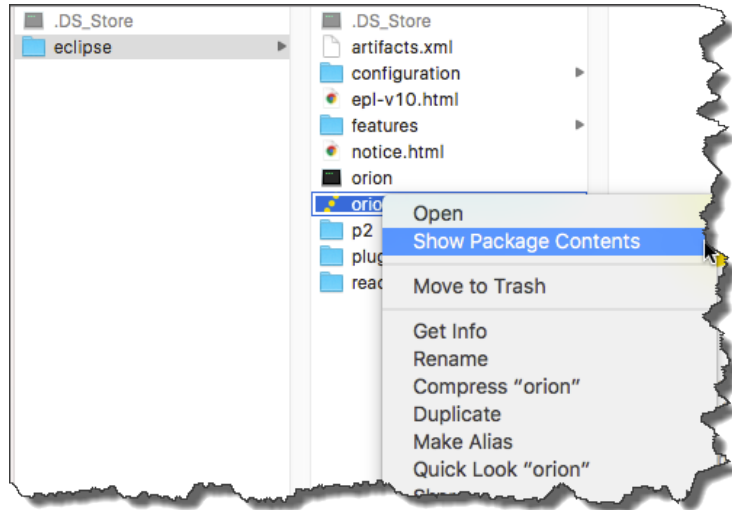
In your landscape, you may need to access remote systems via a proxy server. In this case, you also need to configure the proxy server settings.

Procedure

1. Open the `orion.ini` file:

Open the `orion.ini` file with a text editor.

1. In the Finder go to `orion`, right-click, and select *Show Package Contents*.



2. Go to the terminal and enter `cd /Applications/SAPWebIDE/eclipse/orion.app/Contents/MacOS`
3. Right-click `orion.ini` and open it with a text editor.

2. To change application port server, change the port in the following line:

```
-Dorg.eclipse.equinox.http.jetty.http.port=8080
```

3. To configure the proxy settings, add the following lines at the end of the file, replacing the placeholders with values specific to your landscape:

```
-Dhttp.proxyHost=<Proxy address>
-Dhttp.proxyPort=<Proxy port>
-Dhttps.proxyHost=<Proxy address>
-Dhttps.proxyPort=<Proxy port>
-Dhttps.nonProxyHosts=localhost|<host1>|<host2>
```

4. Save your changes.

10.4 Connect Remote Systems in SAP Web IDE Personal Edition

You can define service destinations to access remote systems as required.

Prerequisites

If you want to connect to an ABAP system, make sure you have checked the requirements in the topic: [Requirements for Connecting to ABAP Systems \[page 60\]](#).

Context

You must create a destination for each remote system to which you want to connect in SAP Web IDE personal edition. For example, when creating a project, in the Data Connection step, you can select an OData service from configured destinations.

Procedure

1. Create a file with the same name as your remote system in the following location:

i Note

Do not use an extension for the file.

Microsoft Windows®

```
C:\SAPWebIDE\eclipse
\config_master
\service.destinations
\destinations
```

Mac OS®

1. Go to the terminal and enter `cd /Applications/SAPWebIDE/eclipse/orion.app/Contents/MacOS/config_master/service.destinations/destinations/`
2. Create a new file by running the following command `touch <your system name>`

2. Open the file you created and add the following configuration, replacing the placeholders with values specific to your system.

```
Description=<mysystem> description
Type=HTTP
TrustAll=true
Authentication=NoAuthentication
Name=<mysystem>
ProxyType=Internet
URL=https\:\/\/<host>\:<port>
WebIDEUsage=<add a value according to the table below>
WebIDESystem=<mysystem>
WebIDEEnabled=true
sap-client=<SAP client number for ABAP systems only. Delete this line if you
are not using an ABAP system.>
```

Key	Value
WebIDEUsage	<p>Enter one or more of the following possible values:</p> <ul style="list-style-type: none"> ○ odata_abap: For the OData functionality of Gateway (corresponds to URL path /sap/opu/odata) ○ odata_gen: For generic OData functionality (service URL must be provided manually in the New Project wizard) ○ ui5_execute_abap: For executing SAPUI5 applications from the SAPUI5 ABAP repository (corresponds to URL path /sap/bc/ui5_ui5) ○ dev_abap: For extensibility scenarios and developing or deploying to the SAPUI5 ABAP repository (corresponds to URL path /sap/bc/adt) ○ bsp_execute_abap: For working with fact sheets (corresponds to URL path /sap/bc/bsp) ○ odata_xs: For SAP HANA XS OData services (corresponds to URL path/sap/hba)

i Note

When you enter multiple usages for a destination, separate them by commas without spaces (for example, `odata_abap,ui5_execute_abap`).

3. Save and close the file.

i Note

If your landscape requires the use of a proxy server to reach your systems, you can configure the Orion Application Server to use a proxy server.

Related Information

[Configure the Orion Application Server \[page 522\]](#)

10.5 Connect to an External Git Repository

Configure SAP Web IDE personal edition to access an external Git repository through a proxy.

Context

Modify the configuration file `orion.ini` to allow SAP Web IDE personal edition requests to pass through a proxy. Then, configure the account settings for the SAP Web IDE personal edition Git client.


Procedure

1. Make sure that the Eclipse Orion console is closed.
2. In the `C:\SAPWebIDE\eclipse` folder, locate and open the `orion.ini` file in the editor.
3. Insert the following entries after the `-vmargs` row:

```
-Dhttp.proxyHost=<proxy>
-Dhttp.proxyPort=<port>
-Dhttps.proxyHost=<proxy>
-Dhttps.proxyPort=<port>
-Dhttps.nonProxyHosts=<hosts to be excluded from proxy>
```

For example:

```
-Dhttp.proxyHost=myproxy.mycompany
-Dhttp.proxyPort=8080
-Dhttps.proxyHost=myproxy.mycompany
-Dhttps.proxyPort=8080
-Dhttps.nonProxyHosts=*.mycompany|localhost
```

4. Save the `orion.ini` file, and restart SAP Web IDE personal edition.
5. From the left sidebar, choose  (Preferences) and select [Git Settings](#).
6. Enter your e-mail and user name, and choose [Update](#).

Results

You have configured your SAP Web IDE personal edition Git client to use the proxy. You can test this by cloning an external Git repository.

Related Information

[Clone Repositories \[page 371\]](#)

10.6 Run Applications with Multiple SAPUI5 Versions

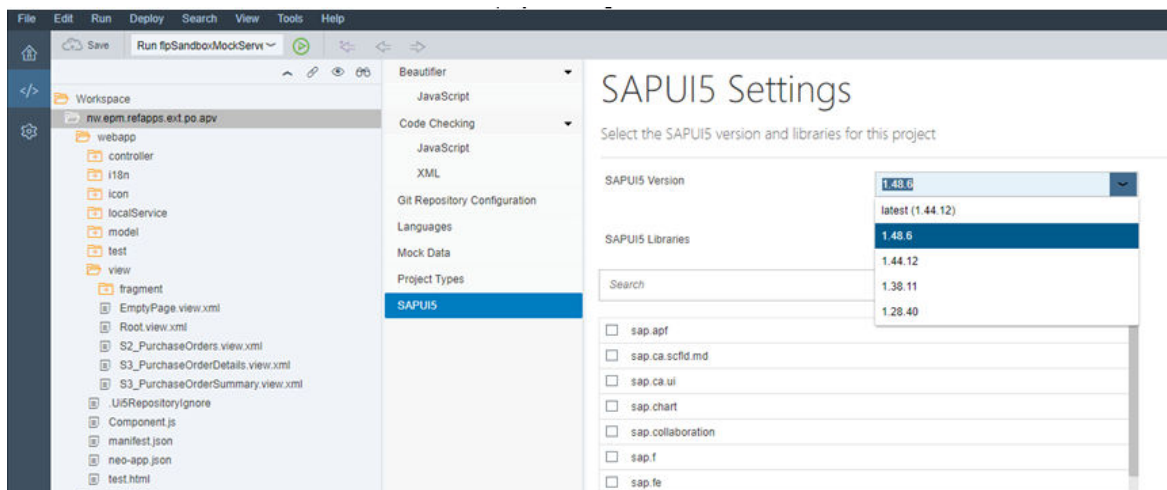
SAP Web IDE personal edition comes with three default SAPUI5 versions you can choose from. If you want to run your application using a different SAPUI5 version you must make the version available in SAP Web IDE personal edition.

1. In your file system, go to `C:\SAPWebIDE\eclipse\plugins\com.sap.webide.orionplugin_1.53.1\ui5` and create a new folder with the SAPUI5 version as the name. For example, 1.48.6.
2. Go to <http://openui5.org/download.html> and download the *Download OpenUI5 SDK* version.
3. Unzip the downloaded file to the new folder you created.
4. Update the `neo-app.json` file in the new folder (`C:\SAPWebIDE\eclipse\plugins\com.sap.webide.orionplugin_1.53.1\ui5\neo-app.json`) with the new added version- for example

```
Sample Code

{
  "path": "/1.48.6",
  "target": {
    "type": "service",
    "name": "sapui5",
    "version": "1.48.6",
    "preferLocal": true
  },
  "description": "SAPUI5 1.48.6"
}
```

5. Go to SAP Web IDE personal edition and refresh your application.
6. Go to **Project Settings** > **SAPUI5** and make sure you can see the new version in the *SAPUI5 Version* dropdown list.



11 Security

When using SAP Web IDE, make sure that your data and processes support your business needs and prevent unauthorized access to critical information.

Errors due to the application users' actions, negligence, or any attempted malicious operation on your system should not result in loss of information or processing time. You must guarantee and comply with the legal regulations regarding protection of users' personal data.

In addition to using this documentation, refer to the other security documentation below:

- [Security Information for SAPUI5](#)
- [SAP HANA Security Guide](#)
- [Identity Service](#)
- [Cloud Connector](#)
- [Cloud Connector Operation Guide](#)
- [Principal Propagation \[page 548\]](#)
- [Destinations](#)
- [HTML5 Applications Development](#)

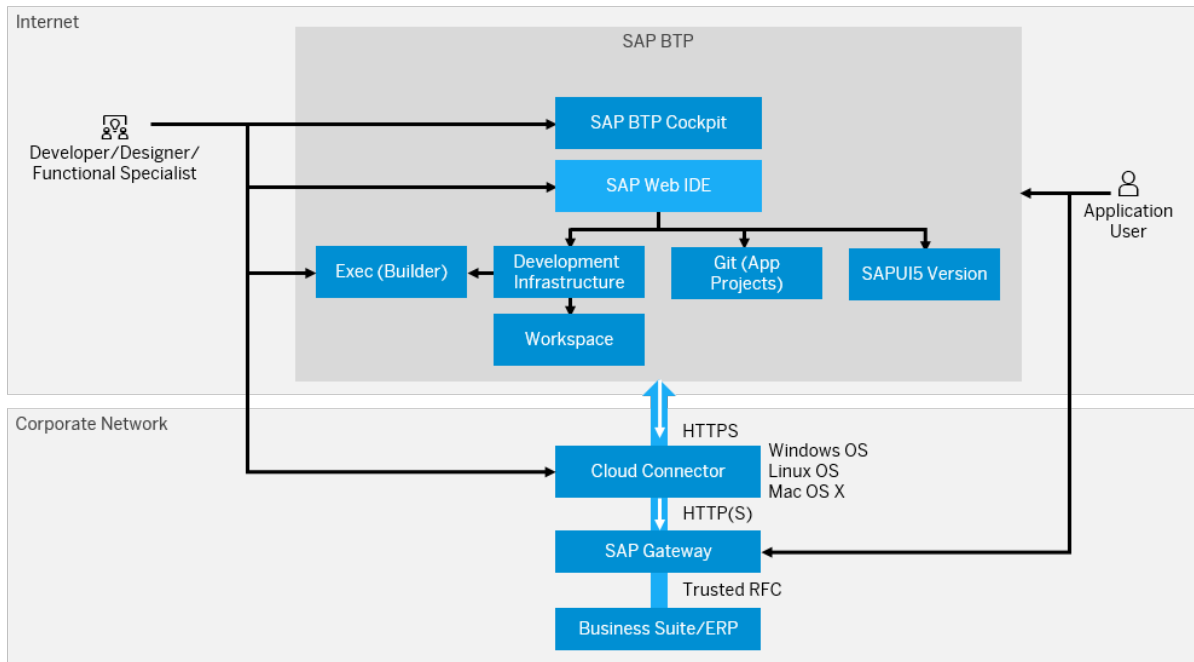
11.1 Architectural Overview

SAP Web IDE is a browser-based IDE consisting of integrated parts that interact with each other and with an SAP system.

SAP Web IDE is integrated with the dispatcher, a mechanism that manages access to the application and various services, and enables you to connect to your SAP systems through REST services.

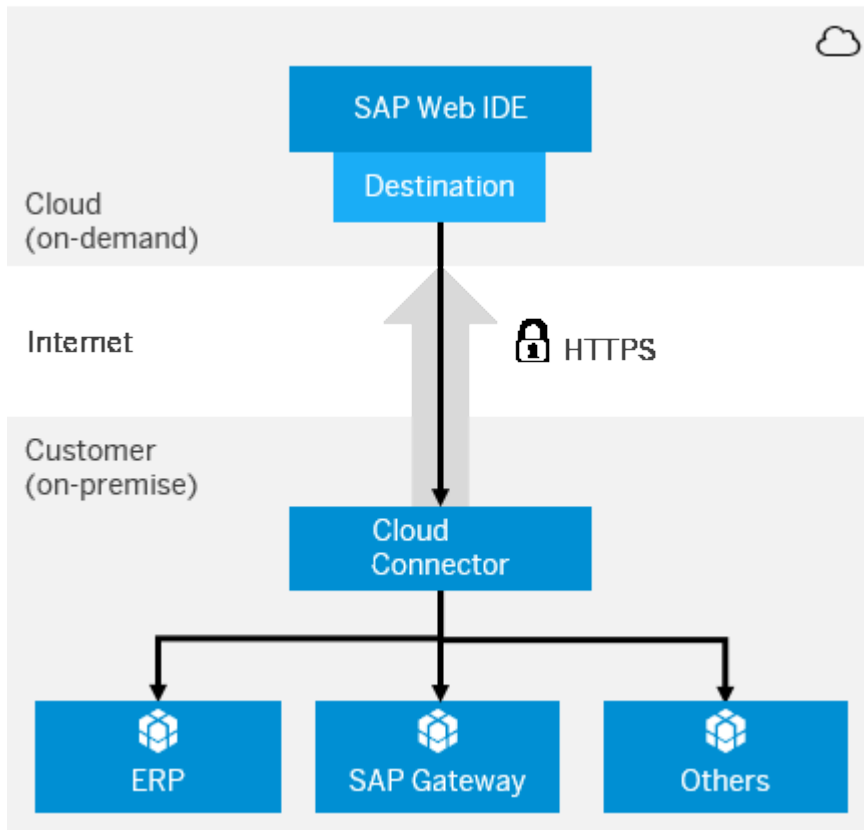
The Development Infrastructure (DI) is a server-side set of development infrastructure services, such as workspace management, build, run, and so on. The services are exposed as a rich, standard REST API, based on Eclipse Che Open-Source Software (OSS).

The image below illustrates the high level typical architecture for SAP Web IDE Full-Stack. For more information about the dispatcher, see [HTML5: Development](#).



SAP Web IDE implements the following features of SAP Connectivity Service to provide a secure connection to on-premise systems:

- Establishes a secure SSL tunnel between SAP BTP and on-premise systems.
- Creates connectivity through an on-premise agent by the reverse-invoke process.
- Safeguards against forgeries by supporting a preconfigured destination API and certificate inspection.
- Is appropriate for both on-premise and cloud landscapes.



11.2 User Authentication and Authorization

User authentication and authorization processes in SAP Web IDE ensure that users can access only the resources to which they have the required permissions.

When developing in SAP Web IDE, you need permissions for the following software components:

- Git: Manages revisions and provides source code control. You must provide user credentials in the SAP ID service and an account to log on to the cloud.
- DI (Development Infrastructure): Used for development infrastructure, workspace management, build, run, and so on. DI leverages SSO as part of the user logon to SAP Web IDE.
- Dispatcher (on SAP BTP): Runs the SAP Web IDE framework. Use your user credentials in the SAP ID service; you must be a cloud account member to manage your source code versions and revisions.

You can log on to SAP Web IDE with the credentials configured for your customer account on SAP BTP. However, to access the Git server on the cloud, you must provide your credentials (user and password) as configured in the SAP ID service.

Related Information

[Authentication \[page 531\]](#)

[Assign Users Permission for SAP Web IDE \[page 532\]](#)

11.2.1 Authentication

Authentication and user propagation in SAP Web IDE occurs in the Dispatcher and consists of two authentication processes: IdP (Identity Provider) configuration and Git authentication.

IdP Configuration

IdP configuration is required to access SAP Web IDE.

You must configure an SAP IdP or a custom IdP as the identity provider for SSO.

i Note

When using SAP Cloud Identity Services as the SAML IdP, you need to configure the SCI trusted domain configuration for the SCI tenant. For more information, see [Configure Trusted Domains](#).

For information, see [SAP ID Service – Single Sign-On for Cloud Applications](#) .

Git Authentication

When you push changes to the Git server, the SAP ID service requests your SCN credentials.

Related Information

[Identity and Access Management \(SAP BTP documentation\)](#)

[Principal Propagation \[page 548\]](#)

[Register, Create and Manage Your Profile \(SCN document\)](#) 

[Cloud Connector \(SAP BTP documentation\)](#)

11.2.2 Assign Users Permission for SAP Web IDE

To develop using SAP Web IDE Full-Stack or manage data stored by the tool, the relevant role needs to be assigned.

Assigning Developer Permissions

Note

You may skip this step if the following conditions are met:

- The developers are members of an account with the `Developer` predefined platform role.
- The platform identity provider and the application identity provider in the account are both SAP ID services. This is the default configuration.

To develop with SAP Web IDE Full-Stack, the `DiDeveloper` role needs to be assigned.

Individual User

The `DiDeveloper` role can be assigned to an individual user as follows:

1. In the SAP BTP cockpit, choose **Services** > **SAP Web IDE Full-Stack** > **Configure Service**.
2. On the **Configure Service** tab, in the table under **Shared HTML5 Roles**, select **DiDeveloper**.
3. In the **Individual Users** area, choose **Assign**, then in the popup window, enter the user ID of the user to whom you want to assign the `DiDeveloper` role, and choose **Assign**.

User Group

Prerequisite

You have created one or more user groups in SAP BTP cockpit. For more information, see [Managing Roles](#).

The `DiDeveloper` role can be assigned to a user group as follows:

1. In the SAP BTP cockpit, choose **Services** > **SAP Web IDE Full-Stack** > **Configure Service**.
2. On the **Configure Service** tab, under **Shared HTML5 Roles**, select **DiDeveloper**.
3. In the **Groups** area underneath, choose **Assign**, then in the popup window, enter the group to which you want to assign the `DiDeveloper` role, and choose **Assign**.

Assigning Administrator Permissions

To be able to manage user data, a user has to be assigned to the `DiAdministrator` role.

This role enables you to export user workspaces and delete user data. For more information, see [Export Workspaces \[page 538\]](#) and [Delete User Data \[page 539\]](#).

Individual User

The `DiAdministrator` role can be assigned to an individual user as follows:

1. In the SAP BTP cockpit, choose [Services](#) > [SAP Web IDE Full-Stack](#) > [Configure Service](#).
2. On the [Configure Service](#) tab, under [Shared HTML5 Roles](#), select [DiAdministrator](#).
3. In the [Individual Users](#) area, choose [Assign](#), then in the popup window, enter the user ID of the user to whom you want to assign the `DiAdministrator` role, and choose [Assign](#).

User Group

Prerequisite

You have created one or more user groups in SAP BTP cockpit. For more information, see [Managing Roles](#).

The `DiAdministrator` role can be assigned to a user group as follows:

1. In the SAP BTP cockpit, choose [Services](#) > [SAP Web IDE Full-Stack](#) > [Configure Service](#).
2. On the [Configure Service](#) tab, under [New Role](#), select [DiAdministrator](#).
3. In the [Groups](#) area underneath, choose [Assign](#), then in the popup window, enter the group to which you want to assign the `DiAdministrator` role, and choose [Assign](#).

Assigning Git Permissions

When using the SAP BTP Git service, users by default have the `accessGit` permissions.

To be able to manage Git repositories, a user has to be assigned to the `DiScpGitAdministrator` role.

This role enables you to use the `manageGit` permissions which will allow you in SAP Web IDE to push commits made by other users (forge committer identity). For more information, see [Platform Scopes](#).

Individual User

The `DiScpGitAdministrator` role can be assigned to an individual user as follows:

1. In the SAP BTP cockpit, choose [Services](#) > [SAP Web IDE Full-Stack](#) > [Configure Service](#).
2. On the [Configure Service](#) tab, under [New Roles](#), select [DiScpGitAdministrator](#).
3. In the [Individual Users](#) area, choose [Assign](#), then in the popup window, enter the user ID of the user to whom you want to assign the `DiScpGitAdministrator` role, and choose [Assign](#).

User Group

Prerequisite

You have created one or more user groups in SAP BTP cockpit. For more information, see [Managing Roles](#).

The `DiScpGitAdministrator` role can be assigned to a user group as follows:

1. In the SAP BTP cockpit, choose [Services](#) > [SAP Web IDE Full-Stack](#) > [Configure Service](#).
2. On the [Configure Service](#) tab, under [New Role](#), select [DiScpGitAdministrator](#).
3. In the [Groups](#) area underneath, choose [Assign](#), then in the popup window, enter the group to which you want to assign the `DiScpGitAdministrator` role, and choose [Assign](#).

Related Information

[Account Member Roles](#)

[Managing Roles and Permissions](#)

[Managing Roles](#)

[ID Federation with the Corporate Identity Provider](#)

11.2.2.1 Maintain an IdP Mapping Rule for a User Group

How to maintain an IdP mapping rule for a large user base.

Prerequisites

You have configured a corporate SAML identity provider (IdP) for your account. For more information, see [ID Federation with the Corporate Identity Provider](#).

Context

This approach requires minimum ongoing effort to maintain an IdP mapping rule, especially where there is a large user base. When this IdP mapping rule is assigned, the required SAP Web IDE authorization is automatically derived from the IdP, making it unnecessary to assign it to each user separately.

Procedure

1. In SAP BTP cockpit, choose **Security > Authorizations**, and select the *Groups* tab. Define a new group, for example, `DiDeveloperGroup`.
2. Choose **Security > Trust**, and in the *Application Identity Provider* tab, click the link with the relevant IdP name.
3. In the *Trusted Identity Provider* dialog box that opens, select the *Groups* tab and click the link, *Add Assertion-Based Group*.
4. In the *Group* dropdown list, select the group created in the first step and define a mapping rule based on the SAML attribute issued by the custom IdP that will automatically assign users to the group if the mapping rule is matched.
5. In SAP BTP cockpit, again choose **Security > Authorizations** and select the *Groups* tab. Select the group you created in the first step and assign it with the `DiDeveloper` role.

Related Information

[Managing Roles](#)

[Account Member Roles](#)

11.2.2.2 Authorization Migration

As an account administrator, you can easily migrate SAP Web IDE user role assignments to SAP Web IDE Full-Stack.

In SAP Web IDE, users have access to the system only if they have been granted the `WebIDEPermission`. For SAP Web IDE Full-Stack, the user needs to be assigned the `DiDeveloper` role instead. The authorization migration will allow you to migrate user permissions easily.

i Note

Only developer permissions (`WebIDEPermission`) are migrated. Administrator permission migration needs to be done manually.

To migrate the user role assignments:

1. In the SAP BTP cockpit, select the [Services](#) tab, then open the SAP Web IDE Full-Stack tile.
2. Select [Configure Service](#) > [Authorization Migration](#).
3. Choose [Migrate](#).
4. Select the relevant target for the authorization migration.

i Note

If the SAP Web IDE `WebIDEPermission` permission is assigned to the `Everyone` role, only the [Grant permissions to a group](#) option is available.

- If you want to migrate the authorizations granting permissions to all users individually, choose [Grant permissions to all users individually](#). According to the current user role assignment in SAP Web IDE, a corresponding role is assigned to each individual user in SAP Web IDE Full-Stack.
- If you want to migrate the authorizations granting permissions to a group of users:
 1. Choose [Grant permissions to a group](#). According to the current user role assignment in SAP Web IDE, a corresponding role is assigned in SAP Web IDE Full-Stack to an existing or newly created group, and then the users are added to the group.
 2. If you are granting permissions for a new group, provide a name for the group. You can either choose the default value (`WebIDEDevelopers`) or enter a new name.
 3. If you are granting permissions for an existing group, enter the name in the [Group Name](#) field and clear the [Create new group](#) checkbox.

If the SAP Web IDE `WebIDEPermission` permission is assigned to the `Everyone` role, once the migration is complete, the SAP Web IDE Full-Stack `DiDeveloper` role is assigned to the specified group and the group is assigned to the default `IdP Group`. As a result, all `IdP` users get development access to Web IDE Full-Stack. To remove an `IdP` default group configuration, you need to delete the created group.

5. Choose *Grant Permissions*. The migration process might take a few minutes.

i Note

Do not close your browser until you get a confirmation message that the migration was successful.

After migration is complete the action status will remain *Idle*.

The authorization assignment results can be viewed in the Roles tab.

11.2.3 Browser Security

Your browser may be subject to potential attacks. SAP Web IDE has a Cross Site Request Forgery (CSRF) protection mechanism to ensure that your data stays secure all the time.

HTML5

Although certain aspects of HTML5 functionality may generally increase security risks, SAP Web IDE and Orion mitigate these risks as follows:

- All browsers offer a session storage API that stores a limited amount of data on the browser. The data can be accessed using JavaScript code in the domain where it is stored. The session storage in SAP Web IDE does not store any confidential information.
- *postMessage* allows inter-window communication between different domains. Basically, this poses a risk in the same origin policy currently implemented in the browser. When you subscribe to the *onMessage* event, you can receive messages from any other browser window. SAP Web IDE uses *postMessage* in its extensibility window; however, it checks the originating domain and only processes messages that are sent by trusted domains.

11.2.4 Transport Security

All cloud applications use only HTTPS, which ensures that communication channels use encrypted connections. In addition, you should use session handling either through cookies or URL rewriting to associate the set of information with the specific user.

Encryption

SAP applications must send HTTP protocol over an SSL secured connection. SAP Web IDE fully supports the use of HTTPS.

→ Recommendation

We recommend that you enable or test SSL connections at an early stage of application development.

Session Security

SAP Web IDE supports Cross Site Request Forgery (CSRF) prevention implemented by target systems or SAP systems, using a CSRF token that is read from the server and used for subsequent write requests.

11.3 Data Protection and Privacy

Overview

SAP Web IDE in trial accounts is governed by the [SAP Web IDE Trial Privacy Statement](#) and SAP Web IDE Trial personal edition is governed by the [SAP Web IDE Trial Personal Edition Privacy Statement](#).

SAP Web IDE stores a user's projects and files as well as his or her preference settings. User data stored by SAP Web IDE can be exported or deleted by an administrator. For more information, see [Export Workspaces \[page 538\]](#) and [Delete User Data \[page 539\]](#).

Destinations created from SAP Web IDE may include user personal data such as user ID and name. The list of destinations can be found in SAP BTP cockpit, in the area for configuring destinations. There, you can see, manage, and delete destinations. For more information, see [Configure Destinations from the Cockpit](#).

For a resource defined in the MTA descriptor (`mta.yaml`) of a user's application, SAP Web IDE may generate a service instance in Cloud Foundry, adding the corresponding user ID as a prefix to the service instance name. An administrator can manage user service instances for MTA projects. For more information, see [Manage User Service Instances \[page 540\]](#).

If an application built using SAP Web IDE includes person-related data, the application must comply with the data protection laws of its target countries. This includes the usage of proper authentication, authorization, and encryption, such as SSO and usage of HTTPS, as well as properly securing and logging access to person-related data.

Git

SAP Web IDE is integrated with Git for source and version control. To work with Git, users need to provide their user name and e-mail in the SAP Web IDE [Preferences](#) perspective and in the [Project Settings](#) of each project. These settings are optional in SAP Web IDE, but if users don't provide them, they will not be able to work with Git.

i Note

If users choose to provide their user name and e-mail, these settings are sent to and stored on the remote Git server of their choice. These settings cannot be deleted.

The settings that are configured in the *Preferences* perspective and the *Git Settings* area of the *Project Settings* are copied to each new project and stored in the Git repository configuration. A user can change these settings for each project because the credentials used for Git might be different for each project.

Related Information

[Using Source Control \(Git\) \[page 363\]](#)

[Setting User Preferences \[page 70\]](#)

11.3.1 Export Workspaces

An administrator can export a `.zip` file containing the projects from all of a user's SAP Web IDE workspaces.

Prerequisites

- Users must be assigned to the `DiAdministrator` role. See [Assign Users Permission for SAP Web IDE \[page 532\]](#).
- The user ID of the user whose workspaces you want to export.

i Note

- If the user ID contains special characters, it should be encoded in URL format.
- The user ID should be in the format defined in the identity provider (IdP) that is configured for the account. For the SAP IdP, the user ID should be uppercase.

- The SAP Web IDE service URL.

i Note

To get the SAP Web IDE service URL, follow these menu options and links:

SAP BTP Cockpit > account > Services > Web IDE Full-Stack tile and then click the [Go to Service](#) link.

In your browser address bar, enter:

```
https://{SAP Web IDE service URL}/di/workspace/export-all/{userid}
```

Result

A `.zip` file containing the contents of all the user's workspaces is downloaded to your computer.

11.3.2 Delete User Data

An administrator can delete a user's workspaces and data from SAP Web IDE.

⚠ Caution

The deletion procedure that is described below permanently removes a user's workspaces and data from SAP Web IDE and is irreversible.

Prerequisites

- You must be assigned to the `DiAdministrator` role. For more information, see [Assign Users Permission for SAP Web IDE \[page 532\]](#).
- The user ID of the user whose workspaces and projects you want to delete.

i Note

- If the user ID contains special characters, the entire ID should be encoded in URL format.
- The user ID should be in the format defined in the identity provider (IdP) that is configured for the account. For the SAP ID service, which is the default identity provider of SAP BTP, the user ID should be uppercase.

- The SAP Web IDE service URL.

i Note

To get the SAP Web IDE service URL, follow these menu options and links:

SAP BTP Cockpit > account > Services > Web IDE Full-Stack tile and then click the [Go to Service](#) link.

- Any REST client that works with your browser.

i Note

The instructions below may vary slightly in different REST clients.

To delete a user's data, you need to follow these basic steps as described in detail in the subsections below:

1. Get certain header information using your browser and REST client.
2. In your REST client, enter the header information you obtained and remove the user's data using the `DELETE` method.

Get Header Information

Cookie

1. In your browser address bar, enter: `https://{SAP Web IDE service URL}` using the URL you saved in the prerequisite.

2. Press the `F12` key and go to the *Network* tab.
3. In the same browser address bar, enter: `https://{SAP Web IDE service URL}/di/healthcheck`
4. Using the *Search* field, search for `healthcheck`.
5. Go to the headers of the response and copy the value of the header `cookie`.

X-CSRF-Token

In your REST client program, do the following:

1. Choose the `GET` method.
2. Enter the request URL: `https://{SAP Web IDE service URL}/di/healthcheck`
3. Add a new request header named `cookie` with the value that you copied.
4. Add another new header for the `X-CSRF-Token` and set the value to `Fetch`.
5. Send the request.
6. Copy the `X-CSRF-Token` value from the response headers.

Delete User Data

1. In your REST client program, enter the URL: `https://{SAP Web IDE service URL}/di/termination/user/{user-id}`
2. Choose the `DELETE` method.
3. Set the `cookie` header with the value you copied previously.
4. Set the `X-CSRF-Token` header with the value you copied previously.
5. Click *Send*.

Result

All user data, including workspaces and projects, is deleted from SAP Web IDE.

i Note

The next time the user whose data you deleted enters SAP Web IDE, a new empty workspace is created automatically for him or her.

11.3.3 Manage User Service Instances

An administrator can manage a user's service instances.

Prerequisites

- `SpaceDeveloper` role for the user Cloud Foundry space.
- The user ID of the user whose service instances you want to manage.

Context

To manage the service instances created for a particular user, perform the following steps:

Procedure

1. Open SAP BTP cockpit.
2. Navigate to your Cloud Foundry account, subaccount and space, and then ► [Services](#) ► [Service Instances](#) ►.
3. Filter the list of services by entering the relevant user ID in the [Search](#) field.
4. Delete any or all of the services, as required.

11.3.4 Data Protection and Privacy for Previous Versions

Overview

As of December, 31st 2018, the SAP Web IDE version is not available anymore and we offer SAP Web IDE Full-Stack as the only cloud version. This section is relevant for users who worked in the SAP Web IDE version before this date.

SAP Web IDE in trial accounts is governed by the [SAP Web IDE Trial Privacy Statement](#) and the SAP Web IDE Trial personal edition is governed by the [SAP Web IDE Trial Personal Edition Privacy Statement](#).

SAP Web IDE stores a user's projects and files as well as his or her preference settings. User data stored by SAP Web IDE can be exported or deleted by an administrator. For more information, see [Export Workspace in Previous Versions \[page 542\]](#) and [Delete User Data in Previous Versions \[page 543\]](#).

Destinations created from SAP Web IDE may include user personal data such as user ID and name. The list of destinations can be found in SAP BTP cockpit, in the area for configuring destinations. There, you can see, manage, and delete destinations. For more information, see [Configure Destinations from the Cockpit](#).

If an application built using SAP Web IDE includes person-related data, the application must comply with the data protection laws of its target countries, including the usage of proper authentication, authorization, and encryption, such as SSO and usage of HTTPS, as well as properly securing and logging access to person-related data.

Git

SAP Web IDE is integrated with Git for source and version control. In order to work with Git, users were required to provide their user name and email in the SAP Web IDE *Preferences* perspective and in the *Project Settings* of each project. These settings were optional.

i Note

If users chose to provide their user name and email, these settings were sent to and stored on the remote Git server of their choice. These settings cannot be deleted.

11.3.4.1 Export Workspace in Previous Versions

An administrator can export a `.zip` file containing projects from the user's SAP Web IDE workspace.

Prerequisites

- You must be assigned to the SAP Web IDE administrator role (SAP Web IDE Admin). For more information, see [Assign User Permissions in Previous Versions \[page 544\]](#).
- You have the user ID of the user whose workspace you want to export.

i Note

- If the user ID contains special characters, it should be encoded in URL format.
- The user ID should be in the format defined in the identity provider (IdP) that is configured for the account. For the SAP IdP, the user ID should be uppercase.

- You have the SAP Web IDE service URL.

i Note

To get the SAP Web IDE service URL, choose:

SAP BTP Cockpit > account > Services > SAP Web IDE tile and then click the [Go to Service](#) link.

In the browser address bar, enter:

```
https://{SAP Web IDE service URL}/orion/export/user/{userid}
```

Result

A `.zip` file containing the contents of the user's workspace is downloaded to your computer.

11.3.4.2 Delete User Data in Previous Versions

An administrator can delete a user's workspaces and data from SAP Web IDE.

⚠ Caution

The deletion procedure that is described below permanently removes a user's workspaces and data from SAP Web IDE and is irreversible.

Prerequisites

- You must be assigned to the `SAP Web IDE Admin` role. For more information, see [Assign User Permissions in Previous Versions \[page 544\]](#).
- The user ID of the user whose workspaces and projects you want to delete.

i Note

- If the user ID contains special characters, the entire ID should be encoded in URL format.
- The user ID should be in the format defined in the identity provider (IdP) that is configured for the account. For the SAP ID service, which is the default identity provider of SAP BTP, the user ID should be uppercase.

- The SAP Web IDE service URL.

i Note

To get the SAP Web IDE service URL, follow these menu options and links:

SAP BTP Cockpit > account > Services > Web IDE tile and then click the [Go to Service](#) link.

- Any REST client that works with your browser.

i Note

The instructions below may vary slightly in different REST clients.

To delete a user's data, you need to follow these basic steps as described in detail in the subsections below:

1. Get certain header information using your browser and REST client.
2. In your REST client, enter the header information you obtained and remove the user's data using the `DELETE` method.

Get Header Information

Cookie

1. In your browser address bar, enter: `https://{SAP Web IDE service URL}` using the URL you saved in the prerequisite.

2. Press the `F12` key and go to the *Network* tab.
3. In the same browser address bar, enter: `https://{SAP Web IDE service URL}/orion/healthcheck`
4. Using the *Search* field, search for `healthcheck`.
5. Go to the headers of the response and copy the value of the header cookie.

X-CSRF-Token

In your REST client program, do the following:

1. Choose the `GET` method.
2. Enter the request URL: `https://{SAP Web IDE service URL}/orion/healthcheck`
3. Add a new request header named `cookie` with the value that you copied.
4. Add another new header for the `X-CSRF-Token` and set the value to `Fetch`.
5. Send the request.
6. Copy the `X-CSRF-Token` value from the response headers.

Delete User Data

1. In your REST client program, enter the URL: `https://{SAP Web IDE service URL}/orion/termination/user/{user-id}`
2. Choose the `DELETE` method.
3. Set the `cookie` header with the value you copied previously.
4. Set the `X-CSRF-Token` header with the value you copied previously.
5. Click *Send*.

Result

All user data, including workspaces and projects, is deleted from SAP Web IDE.

i Note

The next time the user whose data you deleted enters SAP Web IDE, a new empty workspace is created automatically for him or her.

11.3.4.3 Assign User Permissions in Previous Versions

To develop using SAP Web IDE or manage data stored by the tool, the relevant role needs to be assigned.

Prerequisites

- To assign a role to developers or administrators, you must be an administrator on the SAP BTP account.

- To assign a role to a user group, you need to have at least one existing user group or to create one in SAP BTP cockpit.
For more information, see [Managing Roles](#).

Assigning Developer Permissions

1. In the SAP BTP cockpit, in the hierarchy tree on the left, choose [Services](#), then in the tab on the right, choose the [Web IDE](#) tile.
2. Click [Configure Service](#), then in the hierarchy tree, choose [Roles](#).
3. On the [Service Configuration](#) tab, under [Shared HTML5 Roles](#), check if the role you want to assign appears in the table. If not, create one as described below:

i Note

To create a new role, choose [New Role](#) and then in the popup window, enter the name of a new role and choose [Save](#).

4. Assign the SAP Web IDE permission (`WebIDEPermission`) to a developer role:

i Note

This permission is mandatory for developers who want to use SAP Web IDE.

1. In the hierarchy tree above [Roles](#), select [Destinations & Permissions](#) (for the SAP Web IDE service that you selected in step 1).
 2. In the tab, under [Application Permissions](#), choose [Edit](#).
 3. In the table, in the [WebIDEPermission](#) row, in the [Assigned Role](#) column, select the role to which you want to assign the SAP Web IDE permission (`WebIDEPermission`), then choose [Save](#).
5. Assign a role to an individual user or group:
 - **Individual Users**
 1. In the hierarchy tree, select [Roles](#).
 2. In the [Individual Users](#) area, choose [Assign](#).
 3. In the popup window, enter the user ID of the user to whom you want to assign the role, then choose [Assign](#).
The user you assigned now appears in the [Individual Users](#) table.
 - **Groups**
 1. In the hierarchy tree, select [Roles](#).
 2. In the [Groups](#) area, choose [Assign](#).
 3. In the popup window, select the group to which you want to assign the role, then choose [Assign](#).
The group you assigned now appears in the [Groups](#) table.

Assigning Administrator Permissions

To be able to manage user data, a user must be assigned with the `SAP Web IDE Admin` role. This role enables an administrator to export the workspace content of a user or to delete user data.

For more information, see [Export Workspace in Previous Versions \[page 542\]](#) or [Delete User Data in Previous Versions \[page 543\]](#).

You can assign the SAP Web IDE administrator role (SAP Web IDE Admin) as follows:

1. In the SAP BTP cockpit, in the hierarchy tree on the left, select *Services*, then in the tab on the right, choose the *Web IDE* tile.
2. Create a role named SAP Web IDE Admin (if it does not already exist in the table):
 1. In the hierarchy tree, select *Roles*.
 2. In the tab, under *Shared HTML5 Roles*, choose *New Role*.
 3. In the popup, enter **SAP Web IDE Admin** and choose *Save*.
The SAP Web IDE Admin role now appears in the *Shared HTML5 Roles* table.
3. Assign the SAP Web IDE Admin role to an individual user or group:
 - **Individual Users**
 1. In the table under *Shared HTML5 Roles*, select *SAP Web IDE Admin*.
 2. In the *Individual Users* area, choose *Assign*.
 3. In the popup window, enter the user ID of the user to whom you want to assign the SAP Web IDE Admin role, and choose *Assign*.
The user you assigned the SAP Web IDE Admin role now appears in the *Individual Users* table.
 - **Groups**
 1. In the table under *Shared HTML5 Roles*, select *SAP Web IDE Admin*.
 2. In the *Groups* area, choose *Assign*.
 3. In the popup window, select the group to which you want to assign the SAP Web IDE Admin role, and choose *Assign*.
The group you assigned the SAP Web IDE Admin role now appears in the *Groups* table.

11.4 Connection to External Systems

Access to SAP systems and other external systems is based on the Dispatcher and a defined destination, at least one destination per system.

By default, the SAP BTP Identity Authentication service is configured in your account, and can be used for connecting to an external system.

SAP BTP uses destinations as connection properties for accessing target systems. For more information on destinations, see [HTTP Destinations](#).

⚠ Caution

It is strongly recommended to consume remote services from a separate development (test) system and to avoid using a production system.

Back-End Authentication Methods Supported in SAP Web IDE

Authentication Type	Description
No Authentication	<p>Any challenge request by the target system is returned to an end user and results in a credentials dialog box popup.</p> <p>For more information, see the <i>No Authentication</i> section in Client Authentication Types for HTTP Destinations.</p>
Basic	<p>Credentials used for target system access are embedded into the destination configuration. This results in all requests using the same hard-coded credentials.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p>⚠ Caution</p> <p>This option is not applicable for production environments. If you select <i>BasicAuthentication</i>, access to the SAP system is not secured, since it uses the service user credentials for all requests, regardless of the authenticated user's identity.</p> </div> <p>For more information, see the <i>Basic Authentication</i> section in Client Authentication Types for HTTP Destinations.</p>
Certificate-based	<p>For client certificate authentication.</p> <p>For information, see the <i>Client Certificate Authentication</i> section in Client Authentication Types for HTTP Destinations.</p>
Certificate-based (principle propagation)	<p>Based on SAP Cloud Connector, which generates the certificate.</p> <p>For more information, see Principal Propagation Authentication and SAP BTP Connectivity.</p>
OAuth Client Credential Grant	For information, see OAuth Client Credentials Authentication .
OAuth SAML Bearer Assertion	For information, see SAML Bearer Assertion Authentication .
App2App	<p>For information, see Application-to-Application SSO Authentication.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p>i Note</p> <p>For SAP BTP, Neo only. Application-to-application SSO authentication is used in the scenario of SAP BTP application-to-application communication, assuming that both applications are deployed on SAP BTP.</p> </div>
SAP Assertion SSO	<p>For information, see SAP Assertion SSO Authentication.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p>i Note</p> <p>For SAP NetWeaver only. Relevant for SAP systems only, based on SAP assertion tickets.</p> </div>

Related Information

[Managing Destinations](#)

[Remote Connectivity Troubleshooting \[page 562\]](#)

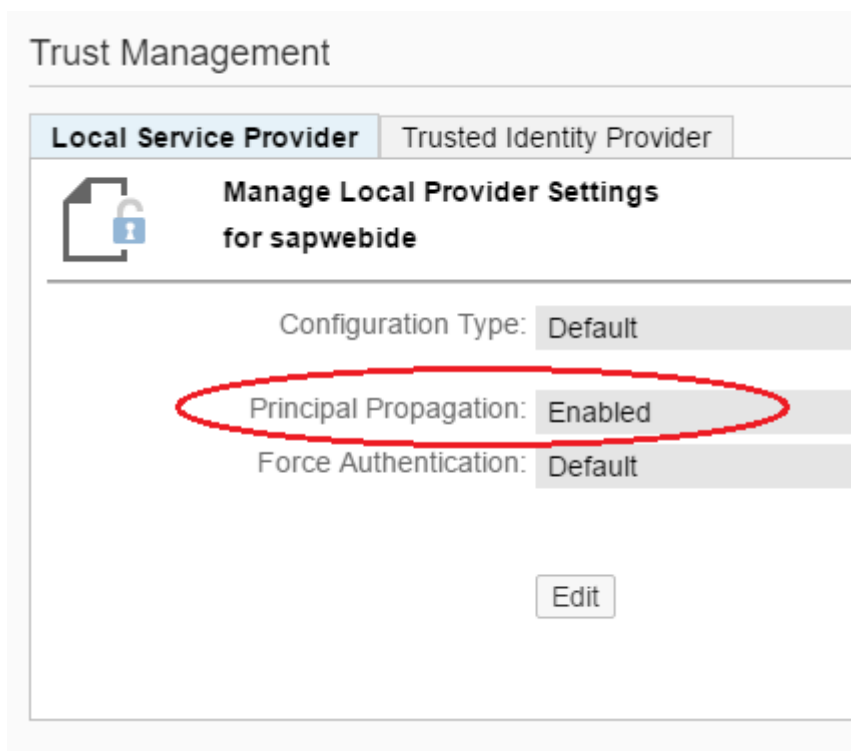
11.5 Principal Propagation

In order to support the SSO solution of SAP Web IDE Full-Stack, you need to configure your account to allow principal propagation.

i Note

The principal propagation setting is configured per account, which means that once it is enabled, applications propagate principal information to each other. If you do not want to enable application-to-application single sign-on, set this option to *Disabled*.

To enable the principal propagation setting, in the SAP BTP cockpit, choose ► [Security](#) ► [Trust Management](#) ► and set *Principal Propagation* to *Enabled*.



For more information about configuring the principal propagation settings, see:

- [Application-to-Application SSO Authentication](#)
- [ID Federation with the Corporate Identity Provider](#)

11.6 Secure Programming Guide

Software security includes improvements of software development process along the entire development life cycle, and not just a one-time event, or simple code review.

For more information, see [Secure Programming Guide](#) in the *SAPUI5 Developer Guide*.

11.7 Auditing and Logging Information

Here you can find a list of the security events that are logged by SAP Web IDE.

Security events written in audit logs

Event grouping	What events are logged	How to identify related log events	Additional information
Workspace-related events	Log a configuration change	<ul style="list-style-type: none"> • Create-member: {WORKSPACE_ID} changed the configuration for the UserId • Add-member-to-workspace: Workspace member updated: successful update of user {USER_ID} at workspace {WORKSPACE_ID} • Remove-user: {USER_ID} changed the configuration for the {UserID}, Removed from workspace {WORKSPACE_ID} • Set-workspace-version: {WORKSPACE_ID} changed the configuration for the {workspace-Version} of {WORKSPACE_ID} 	
	Delete a branch	User {USER_ID} deleted branch {BRANCH_NAME}	
Git-related events	Create Stash	Extracting User {USER_NAME} with email {EMAIL} from Git Configuration to build stash messages	

Event grouping	What events are logged	How to identify related log events	Additional information
	Log a configuration change	User {USER_NAME} changed the configuration for the {attributeName} of {object-Name}	

The following information is described in the table columns:

- *Event grouping* - Events that are logged with a similar format or are related to the same entities.
- *What events are logged* - Description of the security or data protection and privacy related event that is logged.
- *How to identify related log events* - Search criteria or key words, that are specific for a log event that is created along with the logged event.
- *Additional information* - Any related information that can be helpful.

Related Information

[Audit Logging in the Cloud Foundry Environment](#)

[Audit Logging in the Neo Environment](#)

12 Troubleshooting

Here are some common troubleshooting issues in SAP Web IDE.

[Archive Import Troubleshooting \[page 551\]](#)

If you are having trouble importing an archive (.zip file) into SAP Web IDE Full-Stack, you can try the following possible solutions.

[Deployment to SAPUI5 ABAP Repository Troubleshooting \[page 552\]](#)

Steps you can take if you have trouble deploying your app to the SAPUI5 ABAP Repository.

[Deployment to SAP BTP Troubleshooting \[page 554\]](#)

Steps you can take if you have trouble deploying your application to SAP BTP.

[Git Troubleshooting \[page 555\]](#)

Steps you can take if you have trouble using Git.

[Grunt Build Troubleshooting \[page 558\]](#)

If you are having trouble running the Grunt build for your project, you can try the following possible solutions.

[Running Applications Troubleshooting \[page 559\]](#)

If you are having trouble running an application, you can try the following possible solutions.

[Remote Connectivity Troubleshooting \[page 562\]](#)

Options for troubleshooting SAP Web IDE remote connectivity issues.

12.1 Archive Import Troubleshooting

If you are having trouble importing an archive (.zip file) into SAP Web IDE Full-Stack, you can try the following possible solutions.

- **UTF-8 Encoding**

Make sure that the archive files and folder names in your project contain only characters that are encoded with UTF-8 encoding. In other words, these names must not contain any special characters such as a question mark (?) or an ampersand (&). In addition, make sure to use only Latin characters. Non-Latin characters such as Chinese, Japanese, or Hebrew cannot be used.

- **Check File Size**

The .zip file you want to import must be less than 100 MB.

- **Import .zip Files Only**

Make sure that the archive you want to import is a .zip file. Archives with other extensions, such as .rar, cannot be imported.

Multitarget Application (MTA) Project Troubleshooting

- Check the correctness of the `mta.yaml` file.
For more information, see [Inside an MTA Descriptor \[page 440\]](#).
- Make sure that the path element of each module in the `mta.yaml` file points to a folder that actually exists in the archive.
- Make sure that the path value specified in the `mta.yaml` file is a relative path; in other words, that it does not start with a period (`.`), slash (`/`), or backslash (`\`).

12.2 Deployment to SAPUI5 ABAP Repository Troubleshooting

Steps you can take if you have trouble deploying your app to the SAPUI5 ABAP Repository.

The following list includes error messages you may receive when deploying, possible causes of the error, and possible solutions.

Note

The technical messages are displayed in the language selected for your back-end system. Here we provide the English version of these errors.

HTTP Status 504 - An internal application error occurred

This can happen if the application contains a rather large file, and the HTML5 dispatcher (which is on the route from SAP Web IDE to the SAP system) gets a timeout when trying to dispatch it.

Sometimes, the HTML5 dispatcher has a shorter timeout (30s) than the ABAP system itself (5m). When this happens, the ABAP system is actually able to handle the upload of the large file, but the HTML5 dispatcher raises a timeout in the process.

The solution is to increase the timeout of the HTML5 Dispatcher. To do this, follow its instructions under *Destination Properties* in [Accessing REST Services](#) and increase the timeout to the maximum of 300s.

If you're still experiencing this error after raising the timeout, you may need to perform an ICM hard shutdown. ICM (Internet Communication Manager) handles all inbound HTTP/HTTPS/SMTP connections.

Use `SMICM` transaction to view the status of these connections. The hard shutdown may be necessary if you are pushing an app from SAP Web IDE, as the connection may be held open for some reason. The action is triggered from the menu by selecting **Administration > ICM**.

If the problem persists, locate the large file and upload it manually via the `SE80` transaction.

Namespace Errors

There are a variety of namespace errors:

- **Test objects cannot be created in foreign namespaces**
- **Remote creation in customer namespace not possible in SAP systems**
- **SAP object <X> cannot be assigned to package <Y>**

These errors are caused by the namespace you provided.
The target system runs in either SAP or Customer mode.
If it is running in SAP mode, you can only use the SAP namespace in the given application name.
If it is running in Customer mode, you can only the Customer namespace in the given application name.
Another option would be to change the system mode, if possible.
In addition, the namespace given to the application must match the selected package namespace.
- **A dynpro popup has been opened during processing**

This error indicates that the server tried to open a transport selection dialog, which won't work if called via HTTP. A possible reason would be that the given application namespace does not match the namespace of the selected package.

Authorization Errors

There are a variety of authorization errors:

- **No development license for user**
- **No license to edit object**
- **You are not authorized to create**

To deploy, the user has to be registered as a developer in the SAP system and acquire the necessary licenses and authorizations.

Resource <X> does already exist

SAP Web IDE is trying to create a new resource instead of updating an existing one. But the real problem is caused by the `UI5RepositoryPathMapping.xml` file.

This file contains a list of all files in the application and their paths and it should be valid. SAP Web IDE examines this file in order to get the app's structure. If the file is not valid, for example if manual changes have been made to it, you may encounter errors.

Make sure this file is valid and that it depicts the true structure of the application.

Virus Scan Errors

There are a variety of virus scan errors:

- **Virus scan server error**
- **No virus scan profile is selected as the default**

Virus scan errors should not block the deployment process. The virus scan should be configured in such a way so as not to disturb the deployment process.

Also, a default virus scan profile should be selected in the system or switched off entirely.

User <X> is currently editing <Y>

This error means there is an editor lock on the object. Go to the SAP system to release it.

Request <X> is not a local request

Both the package and the transport request have a transport layer assigned to them. In this case, the package has a local transport layer assigned to it, but the transport request created is not a local request.

See [SAP Note 2121673](#) that deals with inconsistencies in the transport handling, and how such inconsistencies might result in this error. Make sure you have the latest release of this note and that your package is defined as described in the note.

12.3 Deployment to SAP BTP Troubleshooting

Steps you can take if you have trouble deploying your application to SAP BTP.

The following list includes error messages you may receive when deploying, possible causes of the error, and possible solutions.

Other artifacts found for the same ID

When an application is deployed to SAP BTP, a build process is automatically triggered in the background. The build process, among other things, flattens the structure of the application, so that its `manifest.json` file resides directly under its root folder in the runtime environment (SAP BTP).

Then, in SAP BTP, the application index service can locate the new application and index it. The application index service identifies an application by its `sap.app/id` attribute in its `manifest.json` file, so this attribute must be unique.

If there is already an application in the account with the same `sap.app/id` attribute, you will get the following error message:

```
Other artifacts found for the same ID
```

To solve this, you can do one of the following:

- Delete the application containing the same `sap.app/id` attribute from the SAP BTP cockpit. For the application index to become aware that the application has been deleted, the administrator needs to open the Fiori Configuration Cockpit (FCC) and go to [App Resources](#). A full replication including a clean-up is triggered.
- Rename the ID of the application you're currently trying to deploy. For example, if the duplicate ID is `hcm.emp.myleaverequests` you might want to change it to `com.mycompany.hcm.emp.myleaverequests`.
 - In your project (all files and folders) search for `hcm.emp.myleaverequests` and replace it with `com.mycompany.hcm.emp.myleaverequests`
 - In your project (all files and folders) search for `hcm/emp/myleaverequests` and replace it with `com/mycompany/hcm/emp/myleaverequests`
- Rename the ID of the deployed application that contains the duplicate ID. For example, if the duplicate ID is `hcm.emp.myleaverequests` you might want to change it to `com.mycompany.hcm.emp.myleaverequests`.
 - Import your customer SAPUI5 application from your SAP BTP account to the SAP Web IDE workspace via **File > Import > Application from SAP BTP**.
 - In your project (all files and folders) search for `hcm.emp.myleaverequests` and replace it with `com.mycompany.hcm.emp.myleaverequests`.
 - In your project (all files and folders) search for `hcm/emp/myleaverequests` and replace it with `com/mycompany/hcm/emp/myleaverequests`.
 - Deploy the application as an update to the existing application.

12.4 Git Troubleshooting

Steps you can take if you have trouble using Git.

The following lists error messages you may receive when using Git, possible causes of the error, and possible solutions.

Cannot convert folders to modules

This error occurs when folders of MTA applications cannot be converted to modules according to the `mta.yaml` file.

- Check the correctness of the `mta.yaml` file. For more information, see [Inside an MTA Descriptor \[page 440\]](#).
- Make sure that the path element of each module in the `mta.yaml` file points to a folder that actually exists in the archive.

- Make sure that the path value specified in the `mta.yaml` file is a relative path; in other words, that it does not start with a period (`.`), slash (`/`), or backslash (`\`).
- Make sure types of modules are one of the following values: `html5`, `hdb`, `nodejs`, `java`, `cds`, `sitecontent`, `dwf`.
- After you have identified and fixed the root cause of the failure:
 - Create a new empty module of the required type.
 - Copy the content from the problematic folder to the new module
 - Update the `mta.yaml` file so that the corresponding `path` property in the module points the new module root folder.
 - Delete the obsolete folder.

Invalid committer

The e-mail listed in the Git repository is not the same as the e-mail assigned to you in Gerrit.

1. Right-click on your project, select **Project Settings** > **Git Repository Configuration** and change the e-mail address in the `user.email` field.
2. Open the **Preferences** perspective, choose **Git Settings**, and verify that the **Git Email Address** value is correct.
3. Commit your changes again, this time by selecting the **Amend** checkbox and then selecting **Commit**. You can then try to push your changes again.

NON FAST FORWARD

Someone else pushed new changes to the remote repository.

Sync your repository (either **Fetch** and then **Rebase**, or **Pull**).

Prohibited by Gerrit

If you did not select the **Add configuration for Gerrit** checkbox when cloning your project, and the Git repository is connected to a Gerrit server, your remote repository will not work.

Right-click on your project, select **Project Settings** > **Git Repository Configuration**, and add an entry with the key `gerrit.createchangeid` and set the value to `true`.

Cannot upload review

You do not have permission to push changes to Gerrit. Request from your administrator permissions on Gerrit.


Checkout failed

One of the files listed in the `.gitignore` file is preventing you from checking out a different branch.

1. Open `.gitignore` file (original branch).
2. Remove lines containing files blocking checkout.
3. Click *Discard* for the file if it appears in the staging table in the Git pane.
4. Stage the `.gitignore` file and commit it.
5. Check out your branch.
6. Create `.gitignore` file and save it. (If the file already exists, then make a small change and save it.)
7. In staging table, right-click the files blocking checkout and select *Untrack and Ignore*.
8. Stage, commit, and push your changes.
9. Merge your change in Gerrit.
10. Check out the original branch.
11. Select *Reset*.

Not authorized

You are not authorized in the Git system. This error may occur simply because the password was incorrect, for one of the following reasons:

- You entered the wrong password.
- The wrong password was cached in the browser. Clear the browser cache of passwords.
- The Git server requires a [personal access token](#)  authorization.

Clone request failed

A clone request may fail for a variety of reasons. Check the error message for the specific reason.

- **Invalid Git repository URL:** You entered the wrong clone URL.
- **Cannot open git-upload-pack:** You are using a Git system within your corporate network, and you did not set up the configuration properly.
- **502 Bad Gateway:** You are using a Git system within your corporate network. The channel to the Git system, which you opened in the cloud connector, is either disconnected or contains the wrong URL to your Git system.
- **503 Service not available:** You are using a Git system within your corporate network. The system is either temporarily not available, or the channel to your system, which you opened in the cloud connector, is disconnected. To check your Git system availability, try to clone from it using Git Bash.
- **SAP Web IDE is not configured to trust the security certificate provided by the Git server:** You are using a Git system within your corporate network, and the Git system certificate is either expired or missing, or the channel to your Git system, which you opened in the cloud connector, is disconnected. Another possible reason for this error is when you are using a custom identity provider that is not properly configured. Make sure that you have configured the required properties, especially the **assertion-based attributes**.

For information about configuring a custom identity provider, see [Configure Trust to the SAML Identity Provider](#).

For more information about configuring an internal Git system, see [Connect to your Corporate Git System \[page 367\]](#).

Authentication not supported

This error occurs in SAP BTP when you are using a custom identity provider that is not properly configured. Make sure that you have configured the required properties, especially the **assertion-based attributes**.

For information about configuring a custom identity provider, see [Configure Trust to the SAML Identity Provider](#).

Fetch request failed (wrong remote URL)

This error occurs after you initialize a local Git repository from a project ([Initialize Local Repository](#)) and then you set a remote repository with the wrong URL. When setting a remote repository, a fetch is automatically performed, and if the wrong URL is entered, the fetch will fail. The error message includes *Git repository not found*.

If you continue and do other actions with the remote repository, these actions will also fail.

To fix the URL, right-click your project and go to [Project Settings](#) [Git Repository Configuration](#) and then change the *remote.origin.url* field to the correct URL.

12.5 Grunt Build Troubleshooting

If you are having trouble running the Grunt build for your project, you can try the following possible solutions.

Running Automated Tests with the Grunt Build's Test Task

You can execute unit and integration tests by triggering the `@sap/grunt-sapui5-bestpractice-test` Grunt task from the CLI.

1. Enter the following command to add the `grunt-sapui5-bestpractice-test` to your `package.json` file:

```
npm install --save-dev @sap/grunt-sapui5-bestpractice-test
```
2. Add the following script to your `package.json` file to enable running the unit and integration tests via npm:

```
"scripts": { "test": "grunt unit_and_integration_tests" }
```

3. Add the following test task to your `Gruntfile.js` file:

```
grunt.loadNpmTasks("@sap/grunt-sapui5-bestpractice-test");
grunt.registerTask("unit_and_integration_tests",
  [
    "test"
  ]
);
grunt.config.merge({
  coverage_threshold: {
    statements: 0,
    branches: 100,
    functions: 0,
    lines: 0
  }
});
```

4. Run the following command from the CLI to execute the unit and integration tests:

```
npm test
```

Running a Grunt Build with an Older SAPUI5 Version

When building a project using the `grunt-sapui5-bestpractice-build`, a build task is performed.

If you are using an SAPUI5 version below 1.52, include the following tasks in the build task:

```
grunt.registerTask('build', [
  'copy:copyToTmp',
  'devxUpdateManifest',
  'cssmin',
  'openui5_preload:preloadTmp',
  'copy:copyToDbg',
  'copy:copyDbgToDist',
  'copy:copyProjectFilesToDist',
  'uglify:uglifyPreload',
  'copy:copyTmpToDist',
  'devxUpdateNeoApp',
  'createManifestBundle',
  'cleanTmp',
  'createsCachebusterInfoJson',
  'cleanupChanges',
  'createResourcesJson'
]);
```

12.6 Running Applications Troubleshooting

If you are having trouble running an application, you can try the following possible solutions.

Running Applications

When running an application, the preview tab is blank. No errors appear in the console.

- This is caused by a missing destination to the DI application.
For more information, see [Running Applications in Development Mode](#).

When running an application, the preview tab opens with 404 Not Found or 500 internal error errors.

- The URL property in the destination to the DI application is not configured correctly.
For more information, see [Running Applications in Development Mode](#).

When running an application, the preview tab does not open.

- If this is your first time running your application in a browser, there might be a popup blocker. Make sure to disable it in your browser preferences.
For more information, see [this blog post](#).

Preview In Frame

When running an application with the [Preview Mode](#) [With Frame](#) option selected, the preview is disabled.

- Check if the application is protected against clickjacking.
 1. In your application, search for the file containing the `sap-ui-bootstrap` section.

Sample Code

```
<script id="sap-ui-bootstrap"
  src="../../resources/sap-ui-core.js"
  data-sap-ui-async="true"
  data-sap-ui-libs="sap.m"
  data-sap-ui-theme="sap_belize"
  data-sap-ui-compatVersion="edge"
  data-sap-ui-resourceroots='{ "demo.demo": "../" }'>
</script>
```

2. Check if the section contains the line `data-sap-ui-frameOptions="deny"`. This indicates that the application will be disabled when displayed in an iframe.
3. Right click the on the project and select [Run](#) [Run Configurations](#).
4. Select the configuration you are using and, in the *General* tab, select [Preview Mode](#) [Without Frame](#).
5. Run the application. The application opens without a frame and can be edited.

Optionally, you can change the `data-sap-ui-frameOption` line to `"allow"`. However, this will remove the clickjacking protection from your application.

For more information, see [Frame Options](#).

When running an application with the [Preview Mode](#) [With Frame](#) option selected, the preview displays the SAP BTP logon screen.

- Make sure the [Accept 3rd-party cookie](#) checkbox is selected in your Chrome settings.

When running an application with the [Preview Mode](#) [With Frame](#) option selected in MAC Safari private mode, the text in the buttons is not properly displayed.

- This is a known limitation of the preview.

AppCacheBuster

When running an application, the preview does not reflect the changes made to the source code.

- In the `neo-app.json` file, search for the `cacheControl` entry and delete it.

When running an application, the application `appcachebuster` settings are overwritten by the SAP Web IDE `appcachebuster`.

1. In your application, search for the file containing the `sap-ui-bootstrap` section.

Sample Code

```
<script id="sap-ui-bootstrap"
  src="../../resources/sap-ui-core.js"
  data-sap-ui-async="true"
  data-sap-ui-libs="sap.m"
  data-sap-ui-theme="sap_belize"
  data-sap-ui-compatVersion="edge"
  data-sap-ui-resourceroots='{ "demo.demo": "../" }'>
</script>
```

2. Check if the section contains the line `data-sap-ui-appCacheBuster="../../<path to external library appcachebuster>"` . For example, `data-sap-ui-appCacheBuster="../../sap/bc/ui5_ui5/ARTEC/PPCUTIL/,/sap/bc/ui5_ui5/ARTEC/PPSUTIL/"`
3. Delete the `sap-ui-appCacheBuster` parameter from the URL.

Security

Cannot run the application after configuring the Active Directory Federation Services (ADFS).

- Set the ACS (Assertion Consumer Service) in the Trusted Identity Providers settings to [Assertion Consumer Service](#).

Other Issues

Header request parameter removed when using SAP Web IDE.

- To send header parameter from SAP Web IDE, add the header parameter names in the `neo-app.json` file in the `'headerWhiteList'` attribute.
See [Run Applications from the Workspace](#).

12.7 Remote Connectivity Troubleshooting

Options for troubleshooting SAP Web IDE remote connectivity issues.

Whenever there is a need to troubleshoot an SAP Web IDE remote connectivity issue, whether it is for a newly developed SAP Web IDE extension or because of a software failure, it is highly recommended to start checking **without** using:

- **SAP Web IDE logic**
Avoid superfluous logic and simplify reproducibility.
- **Web browser XHR** (XMLHttpRequest, such as Ajax, Asynchronous JavaScript, and XML):
Avoid the Same Origin Policy (SOP) web browser security mechanism, which stops scripts from accessing data from a different origin (domain).

Accordingly, check the connectivity with a web browser (enter the URL in the web browser address bar) and HTTP destination with the following two alternatives:

- Based on an HTML5 app:
`https://<app-url>/<route-path>/<resource>`
For more information, see [Accessing REST Services](#).
- Based on SAP Web IDE routing:
`https://<webide-url>/destinations/<destination-name>/<resource>`
In the SAP BTP cockpit destination settings, under *Additional Properties*, add the `WebIDEEnabled` property and set the value to `true`.

If connectivity is still not working properly, continue with deeper analysis options:

- Check the connectivity using a web browser without an HTTP destination.
It is still preferable to proceed without XHR; however, if XHR is used and there is a Cross-Origin Resource Sharing (CORS) failure, then disable the web browser security feature (just during troubleshooting) by using a web browser argument or a web browser add-on.
- Check the connectivity without a web browser and without an HTTP destination; for example, based on a tool such as Postman.

Related Information

[Connection to External Systems](#)

13 Known Issues

Locate the symptom and follow the recommended analysis and resolution steps for it.

Login to Cloud Foundry

SAP Web IDE supports only the SAP ID service for Cloud Foundry login.

Solution

SAP Web IDE was developed to support the SAP ID service for Cloud Foundry login. It does not support the use of custom ID. The additional support of custom IDP is available in SAP Business Application Studio. It supports custom IDP and is fully equipped for developing application for Cloud Foundry. For more information, see this [blog post](#) and the SAP Business Application Studio [community page](#).

Untracked files are Lost in the Git Pane

Performing [Apply Stash](#) may result in the loss of untracked files if there was a conflict when merging stashed files to the current branch.

Solution

You can recover the lost files by resetting the branch and performing Apply Stash. This eliminates the conflicts and allows all stashed files (both tracked and untracked) to be merged.

Cannot Run Application When Workspace Browser is Closed

Running an application from the [Run](#) menu when the workspace browser is closed results in an error.

Solution

Make sure that the workspace browser is open when you run an application.

Unable to Delete a Remote Branch

If your remote branch name contains a slash character (/), you will be unable to delete this remote branch from SAP Web IDE.

Solution

Delete the branch using the command line.

Cannot Clone the Corporate Git System

When trying to clone the corporate Git system, I get the following error message:

Error

Clone request failed

```
https://i[redacted].git: 502 Host not found
```

Solution

Make sure that the internal host entered in the *WebIDEAdditionalData* property of the destination is the same as the URL in the destination itself.

i Note

When you define the cloud connector, there are 2 types of hosts: Internal and Virtual. Make sure the internal and external hosts have the same name.

SAP Web IDE Freezes while Debugging

Make sure you allow popups in your browser the first time run your app.

Importing Project under a Subdirectory

SAP Web IDE does not support projects in subdirectories. The import will fail.

SAPUI5 Elements Might Not Display Correctly in Preview Mode

When you open an application in preview mode, SAPUI5 elements in the iPad or iPhone view might not display as expected.

Parallel Instances of SAP Web IDE Can Cause Issues

Opening several instances of SAP Web IDE in parallel can cause issues and lead to unhandled exceptions. We recommend that you work on a single SAP Web IDE instance.

SAP Web IDE Fails to Load

- SAP Web IDE fails to load after an update because of an error originating from one of the open files in the editor.

Solution

Add the URL parameter `settings=ignore` to the SAP Web IDE URL and refresh your browser. This will force SAP Web IDE to ignore all settings configured, including which files are currently open in the editor. This will force the closing of all open files and allow SAP Web IDE to load.

If this does not work, add the URL parameter `settings=delete` to the SAP Web IDE URL and refresh your browser. SAP Web IDE should now load successfully. Note that this will delete every special configuration made in the settings.

Once SAP Web IDE loads, remove the URL parameter from the SAP Web IDE URL and refresh your browser.

- SAP Web IDE fails to load due to an error originating from a plugin.

Solution

1. Add the URL parameter `settings=ignore` to the SAP Web IDE URL and refresh your browser. SAP Web IDE should now load successfully.
2. Go to **Preferences > Plugins**.
3. Clear the checkbox of the problematic plugin and choose *Save*.
4. Remove the URL parameter `settings=ignore` from the SAP Web IDE URL and refresh your browser.

Web Authorization Changes Cause 403 Error Message

When an identity provider is specified for your system and you did not change the default authorization settings accordingly, you will receive the following error message:

HTTP Status 403 - You are not authorized to access this resource

Solution

Set (or ask your administrator to set) the proper authorization settings for your system. See [Assign Users Permission for SAP Web IDE \[page 532\]](#).

Error Message When Logging On

You may receive an error message *Unhandled Error: Unexpected token* when logging on to SAP Web IDE.

Solution

Close the message and refresh the browser.

Issues with Logging On Again to SAP Web IDE

You may have issues logging on again to SAP Web IDE, for example, after logging out.

Solution

Empty your browser cache and log on again.

Browser Issues

Mozilla Firefox Browser Support

SAPUI5 does not support smartphone simulation in the Mozilla Firefox browser using the `fakeOS` parameter in the URL (which is what the preview service requires).

When you preview your application in the Mozilla Firefox browser, it will always run as a tablet or desktop.

Safari Browser Support

- **Basic Authentication**

If you defined a destination to a remote system that requires a *Basic Authentication* popup, this popup may not appear for synchronous requests when running or extending an application with SAP Web IDE in the Safari browser.

For more information on defining a destination to a remote system, see [Connect to ABAP Systems \[page 57\]](#)

Solution

If you cannot change the application coding, you can use a different authentication type in the destination maintenance, such as *Principle Propagation*.

For your own application, ensure that you load the OData metadata asynchronously:

```
For example, new sap.ui.model.odata.ODataModel(sServiceUrl, {json: true, loadMetadataAsync: true});
```

- **Exported Files**

Exported files of projects or folders do not have the .zip extension. (To export a file, choose **File** > *Export Project or Folder* from the menu bar.)

Solution

Manually add **.zip** as an extension to the exported file.

- **Layout Editor**

SAP Web IDE layout editor is not supported on Safari.

- **Running with Frame**

The Running with Frame feature does not work properly when using Private Browsing.

Internet Explorer Issues

When previewing an application in regular preview or via the Extensibility pane, you might receive an error message saying the content was blocked because it was not signed by a valid security certificate.

Solution

Add SAP Web IDE as a trusted site as follows:

1. Open your Microsoft Internet Explorer and select *Internet Options*.
2. Choose the *Security* tab.
3. Choose *Trusted sites*.
4. Make sure the *Enable Protected Mode* checkbox is deselected.
5. Choose *Sites*.

6. To add the SAP Web IDE web site to the list of trusted sites, enter the website URL into the provided field.
7. Click [Add](#).
8. Close the [Trusted sites](#) dialog box and choose [OK](#).

If this issue occurs only in the Extensibility pane, run a regular preview of the extension project first and then try the pane again.

Cannot hide a control within a fragment

If you hide a control residing within a fragment that is loaded dynamically during the application's runtime application, it may still appear in the application's UI, even if it is marked as [hidden](#) in the Extensibility pane.

Solution

You can try to hide these controls by replacing the view or extending the controller that hides it and override the method.

Issues in the Layout Editor

- **Data Binding**
Data binding supports only 1 OData service.
- **Unsupported Controls**
Some controls are not supported by the layout editor. These unsupported controls are marked with the label [unsupported](#) when selected in the canvas or in the **Outline** pane.
The following aspects apply to unsupported controls:
 - They are not displayed in the palette.
 - You cannot edit their properties.
 - You can delete them.
 - You cannot delete, move, or change the properties of controls that are children of an unsupported control.

i Note

Use the XML code editor to perform operations on the view layout that you cannot achieve with the layout editor.

- **Only ResponsiveGridLayout for SimpleForm Control Supported**
The layout editor only supports the [ResponsiveGridLayout](#) layout for the [SimpleForm](#) control. Other layouts might cause issues with the graphical display of the XML view in the canvas.
- **Problems on Safari**
SAP Web IDE layout editor is not supported on Safari.

Blank Page When Comparing Code in Git

When opening the compare editor in Git, by double-clicking the file in the staging table row, you may find that a blank page opens.

Solution:

Close the tab of the file that you want to compare and double-click it again in the staging table row. This time the compare editor opens properly.

Deletion of the .Project.json File Content

When manually editing a project's `.project.json` file, or if the file has syntax errors, its content might be automatically deleted.

Solution:

Create a copy of the `.project.json` file before you edit and revert to it if needed.

Duplication of Projects

A project might appear multiple times in the SAP Web IDE workspace.

The project is not physically duplicated. The data has one occurrence in the data storage.

Browser Issue When Running Reference Applications with Mock Data

Running reference applications with mock data will not work on the Microsoft Internet Explorer[®] browser.

Solution:

You can try one of the following:

- Reduce length of the mock data files to less than 1,000 lines.
- Use another browser.

Refactor and Find References of Class Variables within Function

When trying to find references for a class variable (that is, properties of the `this` object) from within a function, only references from within the function are shown in the *References* tab of the *Search* pane. This also affects refactoring: when refactoring a class variable from within a function, only the references of the class variable within the function are changed.

Replace Data Source enhancements

When using the *Replace with Node* option, if the user tries to adjust the mapping via the *Remove mapping* context menu option, the mapping might get corrupted.

Solution:

Avoid using the *Remove mapping* context menu option together with the *Replace with Node* option. We recommend you adjust the mapping in the mapping pane instead.

Synonym Support on SAP BTP

There is not sufficient authorization to view the data preview of Synonyms, which consumes the Calculation View from other containers.

QR Code in Preview Frame

When running an application in a frame, long application URLs can cause the application's QR code not to work. Remove long URL parameters and try running the application again.

14 Third-Party Restrictions

Customer applications developed in SAP Web IDE must:



- Use SAP BTP only as a platform and connect only to official SAP BTP APIs.
- Make no direct usage of third-party components within SAP BTP.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2021 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.