



PUBLIC
SAP Web IDE
2018-07-23

SAP Web IDE Developer Guide

Content

1	SAP Web IDE.	6
2	What's New in SAP Web IDE - May 2018.	7
3	Overview.	150
3.1	Features and Benefits.	151
	Assisted Development.	153
	Collaborative Development.	154
	User-Centric Customizability and Session Persistence.	155
4	Getting Started.	156
4.1	Open SAP Web IDE.	156
4.2	Connect to ABAP Systems.	158
	Requirements for Connecting to ABAP Systems.	161
5	Setting User Preferences.	163
6	SAP Web IDE Basics.	165
6.1	Navigate SAP Web IDE.	165
6.2	Keyboard Shortcuts.	166
	Customize Keyboard Shortcuts.	169
6.3	Workspace Search Options.	170
	Find and Replace in an Open File.	171
	Search for Files or Content in the Workspace.	172
	Replace Strings Across Multiple Files.	173
	Find References.	175
6.4	Work with Files and Folders.	176
	Work with Multiple Open Files.	177
6.5	Resize Panes.	178
7	Developing.	179
7.1	Creating Projects.	180
	Create Projects from Templates.	181
	Create Projects from Sample Applications.	188
	Add a New Component.	189
7.2	Importing Projects.	190
	Import Projects from an Archive.	190
	Import Applications from SAP Cloud Platform.	191
	Import Applications from the SAPUI5 ABAP Repository.	192

7.3	Customizing Your Project	193
	Configure Annotation Modeler to Use Mock Data.	194
	Customize JavaScript Beautifier Properties.	195
	Customize Code Checking Rules.	196
	Customize Code Checking Triggers.	197
	Define Application Languages.	197
	Use the Translation Hub.	198
	Configure Mock Data Usage.	199
	Set Project Types.	200
	Customize Performance Measures.	201
	Set the SAPUI5 Version.	201
	Configure Run Configurations for the UI Adaptation Editor	202
	Use the Smart Business Service.	203
7.4	Modifying the Application Descriptor Configuration.	205
	Settings Tab Options.	206
	Data Sources Tab Options.	208
	Models Tab Options.	208
	Routing Tab Options.	209
	Navigation Tab Options.	211
7.5	Developing Applications.	213
	Working in the Code Editor.	214
	Using Code Completion.	220
	Checking Code	239
	Developing Apps Using SAP Fiori Elements.	325
	Using the Outline Pane for JavaScript Files.	329
	Creating an HTML5 Application Descriptor File.	330
7.6	Developing Application Tests.	331
	Create Tests.	332
7.7	Layout Editor.	332
	Work with the Layout Editor.	338
	Create a Quick Start Application with the Layout Editor.	348
	SAPUI5 Controls Supported in the Layout Editor.	349
	Try It: Build an Application with the Layout Editor.	362
7.8	Annotation Modeler.	369
	Architecture.	370
	User Interface.	372
	Prerequisites for Working with Annotation Modeler.	374
	Adding Local Annotation Files to the Project.	376
	Working with Annotation Modeler.	377
	Troubleshooting.	393
7.9	UI Adaptation Editor.	395

Change Applications with the UI Adaptation Editor.	397
UI Adaptation Options.	399
7.10 Using Source Control (Git).	402
Set Up Git.	406
Connect to your Corporate Git System.	408
Clone Repositories.	410
Initialize a Local Git Repository.	411
Set a Remote Repository.	412
Fetch Changes.	413
Rebase Changes.	413
Merge Changes.	415
Pull Changes.	415
Stage Files.	416
Commit Changes.	421
Push Changes.	422
Use Multiple Local Branches.	423
Reset a Local Branch.	425
Create a Remote Branch.	425
Git History.	426
Set Up Git to Work with Gerrit.	429
7.11 Running Applications in Development Mode.	432
Run Applications from the Workspace.	433
Create Run Projects.	435
Run Applications in a Frame.	440
Run Applications in the SAP Fiori Launchpad Environment.	441
Run Unit Tests on Applications.	443
Run Applications with Mock Data.	443
7.12 Running Applications with Performance Measurements.	446
7.13 Deploying Applications.	447
Deploy Applications to the SAPUI5 ABAP Repository.	447
Update Existing Applications Residing in the SAPUI5 ABAP Repository.	450
Deploy Applications to SAP Cloud Platform.	451
Register Applications to SAP Fiori Launchpad.	454
Check the Application Status.	456
Application Build.	456
7.14 Extending SAPUI5 Applications.	457
Extend Applications that Reside in the SAPUI5 ABAP Repository.	458
Extend Applications that Reside on SAP Cloud Platform.	459
Create New Extensions.	460
8 Plugins and Add-Ons.	472
8.1 Enable Plugins.	474

9	Extending SAP Web IDE.....	475
10	SAP Web IDE Personal Edition.....	476
10.1	Installation and Setup.....	477
	Install SAP Web IDE Personal Edition.....	477
	Start SAP Web IDE Personal Edition.....	481
	Upgrade SAP Web IDE Personal Edition.....	483
	Uninstall SAP Web IDE Personal Edition.....	484
10.2	Import the Git Server Certificate into the JVM.....	484
10.3	Configure the Orion Application Server.....	485
10.4	Connect Remote Systems in SAP Web IDE Personal Edition.....	486
10.5	Connect to an External Git Repository.....	489
10.6	Run Applications with Multiple SAPUI5 Versions.....	490
11	Security.....	491
11.1	Architectural Overview.....	491
11.2	User Authentication and Authorization.....	494
	Authentication.....	494
	Assign Users Permission in SAP Web IDE.....	495
	Browser Security.....	497
	Transport Security.....	497
11.3	Data Protection and Privacy.....	499
	Export Workspace.....	500
	Delete User Data.....	500
11.4	Secure Programming Guide.....	502
12	Troubleshooting.....	504
12.1	Deployment to SAPUI5 ABAP Repository Troubleshooting.....	504
12.2	Deployment to SAP Cloud Platform Troubleshooting.....	506
12.3	Git Troubleshooting.....	507
13	Known Issues.....	510
14	Third-Party Restrictions.....	517

1 SAP Web IDE

This guide is also available in [Japanese](#) and [Chinese](#).



[What's New in SAP Web IDE - May 2018 \[page 7\]](#)

Learn about SAP Web IDE new features.



[Overview \[page 150\]](#)

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



[Getting Started \[page 156\]](#)

Learn how to set up and start SAP Web IDE.



[Developing \[page 179\]](#)

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



[Plugins and Add-Ons \[page 472\]](#)

Understand how to use additional SAP Web IDE plugins and add-ons.



[Extend SAP Web IDE \[page 475\]](#)

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



[Secure \[page 491\]](#)

Understand the security landscape.



[SAP Web IDE Personal Edition \[page 476\]](#)

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

2 What's New in SAP Web IDE - May 2018

This section gives you an overview of the new features and functions of SAP Web IDE as of **May 2018**.

For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

Project Migration to SAP Web IDE Full-Stack (New)

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

SAP Web IDE Full-Stack is the newest release of SAP Web IDE, based on the Eclipse Che foundation.

For more information on SAP Web IDE Full-Stack, see [Announcing: Switch over to SAP Web IDE Full-Stack](#).

For more information about how to migrate your projects, see [Migrating a project to SAP Web IDE Full-Stack](#).

You can also find all the information concerning migration in the SAP Web IDE Full-Stack Developer guide, see [Migrate your Projects](#).

Related Information

[What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

2.1 What's New in Previous Releases of SAP Web IDE

[May 2017 \[page 9\]](#)

Learn about the new and changed features that are available for SAP Web IDE in May 2017.

[April 2017 \[page 10\]](#)

Learn about the new and changed features that are available for SAP Web IDE in April 2017.

[March 2017 \[page 12\]](#)

Learn about the new and changed features that are available for SAP Web IDE in March 2017.

[February 2017 \[page 14\]](#)

Learn about the new and changed features that are available for SAP Web IDE in February 2017.

[January 2017 \[page 18\]](#)

Learn about the new and changed features that are available for SAP Web IDE in January 2017.

[December 2016 \[page 22\]](#)

Learn about the new and changed features that are available for SAP Web IDE in December 2016.

[November 2016 \[page 31\]](#)

Learn about the new and changed features that are available for SAP Web IDE in November 2016

[October 2016 \[page 36\]](#)

Learn about the new and changed features that are available for SAP Web IDE in October 2016.

[September 2016 \[page 40\]](#)

Learn about the new and changed features that are available for SAP Web IDE in September 2016.

[August 2016 \[page 47\]](#)

Learn about the new and changed features that are available for SAP Web IDE in August 2016.

[July 2016 \[page 51\]](#)

Learn about the new and changed features that are available for SAP Web IDE in July 2016.

[June 2016 \[page 55\]](#)

Learn about the new and changed features that are available for SAP Web IDE in June 2016.

[May 2016 \[page 60\]](#)

Learn about the new and changed features that are available for SAP Web IDE in May 2016.

[April 2016 \[page 64\]](#)

Learn about the new and changed features that are available for SAP Web IDE in April 2016.

[March 2016 \[page 73\]](#)

This section gives you an overview of the new features and functions of SAP Web IDE. It includes release notes that describe what is new, enhanced, changed, or deleted in March 2016.

[February 2016 \[page 79\]](#)

This section gives you an overview of the new features and functions of SAP Web IDE. It includes release notes that describe what is new, enhanced, changed, or deleted.

[January 2016 \[page 86\]](#)

Learn about the new and changed features that became available on January 2016.

[SAP Web IDE 1.19 \[page 90\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.19.

[SAP Web IDE 1.18 \[page 96\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.18.

[SAP Web IDE 1.17 \[page 99\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.17.

[SAP Web IDE 1.16 \[page 103\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.16.

[SAP Web IDE 1.15 \[page 106\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.15.

[SAP Web IDE 1.14 \[page 109\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.14.

[SAP Web IDE 1.13 \[page 111\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.13

[SAP Web IDE 1.12 \[page 114\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.12.

[SAP Web IDE 1.11 \[page 118\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.11.

[SAP Web IDE 1.10 \[page 121\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.10.

[SAP Web IDE 1.9 \[page 124\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.9.

[SAP Web IDE 1.8 \[page 127\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.8.

[SAP Web IDE 1.7 \[page 131\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.7.

[SAP Web IDE 1.6 \[page 136\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.6.

[SAP Web IDE 1.5 \[page 138\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.5.

[SAP Web IDE 1.4 \[page 140\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.4.

[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.3 for trial use on hanatrial.ondemand.com.

[SAP Web IDE 1.2 \[page 146\]](#)

Learn about the new and changed features that are available for SAP Web IDE 1.2.

2.1.1 May 2017

Learn about the new and changed features that are available for SAP Web IDE in May 2017.

SAP Web IDE Full-Stack (New)

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 Platform cockpit services, you will see two different tiles, SAP Web IDE and the new SAP Web IDE Full-Stack.

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

For more information, see [SAP Web IDE Full-Stack](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.2 April 2017

Learn about the new and changed features that are available for SAP Web IDE in April 2017.

For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

Annotation Modeler (Changed)

The annotation modeler now supports annotating entity type properties.

For more information, see [Annotation Modeler \[page 369\]](#).

Creating Project from a Template (Changed)

You can now explore the services hosted by [SAP API Business Hub](#) when selecting a data source. For more information, see [Select a Data Source \[page 183\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)

[March 2017 \[page 12\]](#)

[February 2017 \[page 14\]](#)

[January 2017 \[page 18\]](#)

[December 2016 \[page 22\]](#)

[November 2016 \[page 31\]](#)

[October 2016 \[page 36\]](#)

[September 2016 \[page 40\]](#)

[August 2016 \[page 47\]](#)

[July 2016 \[page 51\]](#)

[June 2016 \[page 55\]](#)

[May 2016 \[page 60\]](#)

[April 2016 \[page 64\]](#)

[March 2016 \[page 73\]](#)

[February 2016 \[page 79\]](#)

[January 2016 \[page 86\]](#)

[SAP Web IDE 1.19 \[page 90\]](#)

[SAP Web IDE 1.18 \[page 96\]](#)

[SAP Web IDE 1.17 \[page 99\]](#)

[SAP Web IDE 1.16 \[page 103\]](#)

[SAP Web IDE 1.15 \[page 106\]](#)

[SAP Web IDE 1.14 \[page 109\]](#)

[SAP Web IDE 1.13 \[page 111\]](#)

[SAP Web IDE 1.12 \[page 114\]](#)

[SAP Web IDE 1.11 \[page 118\]](#)

[SAP Web IDE 1.10 \[page 121\]](#)

[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)
[What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

2.1.3 March 2017

Learn about the new and changed features that are available for SAP Web IDE in March 2017.

Beta Release of SAP Web IDE Innovation Version (New)

SAP Web IDE, innovation version is a new beta version of SAP Web IDE that is based on an open-source server-side foundation called Eclipse Che. In SAP Web IDE, innovation version, you can import your existing SAP Web IDE projects and create and run Grunt builds. At this time, the innovation version is being offered in addition to the regularly released version of SAP Web IDE.

Gerrit Pane Lets You Submit Change (New)

From within the *Gerrit* pane, you can now submit changes that are ready to be merged. You can also see the code review and verified status for each change.

For more information, see [Viewing Changes in the Gerrit Pane \[page 430\]](#).

Create Remote Branches in Git (New)

You can now create remote branches for your Git projects. Right-click your project and choose *Git* *Create Remote Branch*.

i Note

To create a local branch, you can choose the menu *Git* or click the plus sign in the *Git* pane.

For more information, see [Create a Remote Branch \[page 425\]](#).

Viewing Live Data (Changed)

When selecting an OData service, you can get a preview of live production data associated with an entity set. Apart from the preview, you can see a maximized view that displays a limited set of records and properties of that entity set. With this release, you can see all the records and properties of an entity set through pagination. You can further customize this view by sorting, filtering and selecting specific properties.

For more information, see [Viewing Live Data \[page 187\]](#).

OData Model Editor (Changed)

The OData Model Editor now provides an outline view of the OData model in a tree structure. This view makes navigating through the models, especially large ones spanning hundreds of lines of code, much easier.

For more information, see [OData Model Editor plugin](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)
- [SAP Web IDE 1.17 \[page 99\]](#)
- [SAP Web IDE 1.16 \[page 103\]](#)
- [SAP Web IDE 1.15 \[page 106\]](#)
- [SAP Web IDE 1.14 \[page 109\]](#)

[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

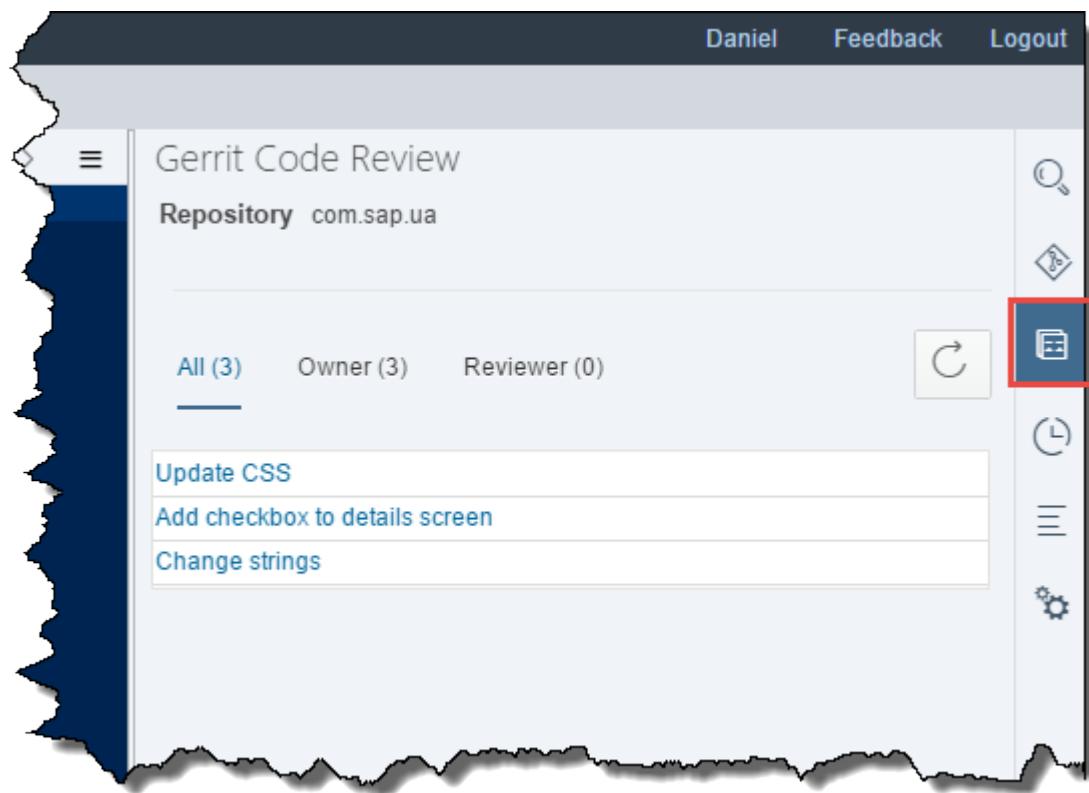
2.1.4 February 2017

Learn about the new and changed features that are available for SAP Web IDE in February 2017.

For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

Gerrit Pane (New)

The new *Gerrit* pane lets you view all your changes related to your SAP Web IDE projects that are open in Gerrit, both the changes you own and the changes for which you were added as a reviewer. In the pane, click on a change and the change opens in your Gerrit system.



There is also a new Git setting that lets you enable/disable the *Gerrit* pane.

For more information, see [View Changes in the Gerrit Pane \[page 430\]](#).

i18n Code Completion (New)

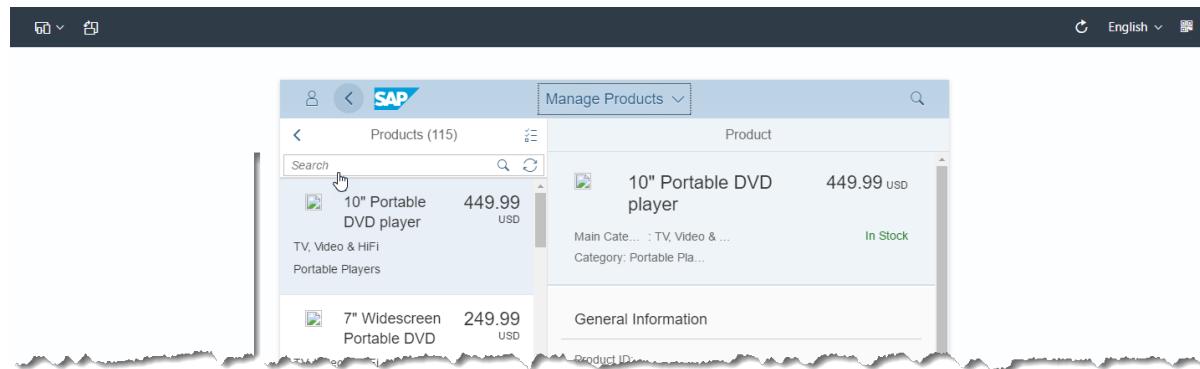
You can now get code completion when you need to select a string to add to an XML or JavaScript file. Put your cursor inside quotation marks and press **Ctrl** + **Space**, and get a list of strings defined in your i18n properties files (the properties files must be defined as i18n models in your manifest).



For more information, see [What's New in SAP Web IDE - May 2018 \[page 7\]](#)[Working in the Code Editor \[page 214\]](#).

Redesigned Preview Frame (Changed)

The window in which applications are previewed (when selecting a run configuration with a frame) has been redesigned for easier use. All the same functionality is provided.



Name of SAP Fiori Run Configurations (Changed)

The SAP Fiori run configuration type is now called [SAP Fiori Launchpad Sandbox](#) (formerly, [SAP Fiori Component on Sandbox](#)).

Annotation Modeler (Changed)

- The annotation modeler now always uses the up-to-date metadata by retrieving it from the backend. To avoid inconsistencies, you will be informed if the backend is not accessible rather than continue working with the potentially outdated data from the local copy.
- The annotation modeler UI now provides vocabulary descriptions of terms and properties in tooltips. Please note that tooltips are shown only for terms and properties that have description in the corresponding OData vocabulary.
- The action [Clone for overriding](#) has a new icon: 

For more information, see [Annotation Modeler \[page 369\]](#).

Project Template (Deprecated)

The *SAP Web IDE Plugin* project template has been deprecated. It is no longer available as of this release of SAP Web IDE. Instead, use the *SAP Web IDE Feature* project template for creating new plugins within a feature.

For more information, see *Creating Our First Plugin* and *Adding a Plugin to an Existing Feature* in the *Tutorial* section of the *SAP Web IDE SDK*.

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)
- [SAP Web IDE 1.17 \[page 99\]](#)
- [SAP Web IDE 1.16 \[page 103\]](#)
- [SAP Web IDE 1.15 \[page 106\]](#)
- [SAP Web IDE 1.14 \[page 109\]](#)
- [SAP Web IDE 1.13 \[page 111\]](#)
- [SAP Web IDE 1.12 \[page 114\]](#)
- [SAP Web IDE 1.11 \[page 118\]](#)
- [SAP Web IDE 1.10 \[page 121\]](#)
- [SAP Web IDE 1.9 \[page 124\]](#)
- [SAP Web IDE 1.8 \[page 127\]](#)
- [SAP Web IDE 1.7 \[page 131\]](#)
- [SAP Web IDE 1.6 \[page 136\]](#)
- [SAP Web IDE 1.5 \[page 138\]](#)

- [SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

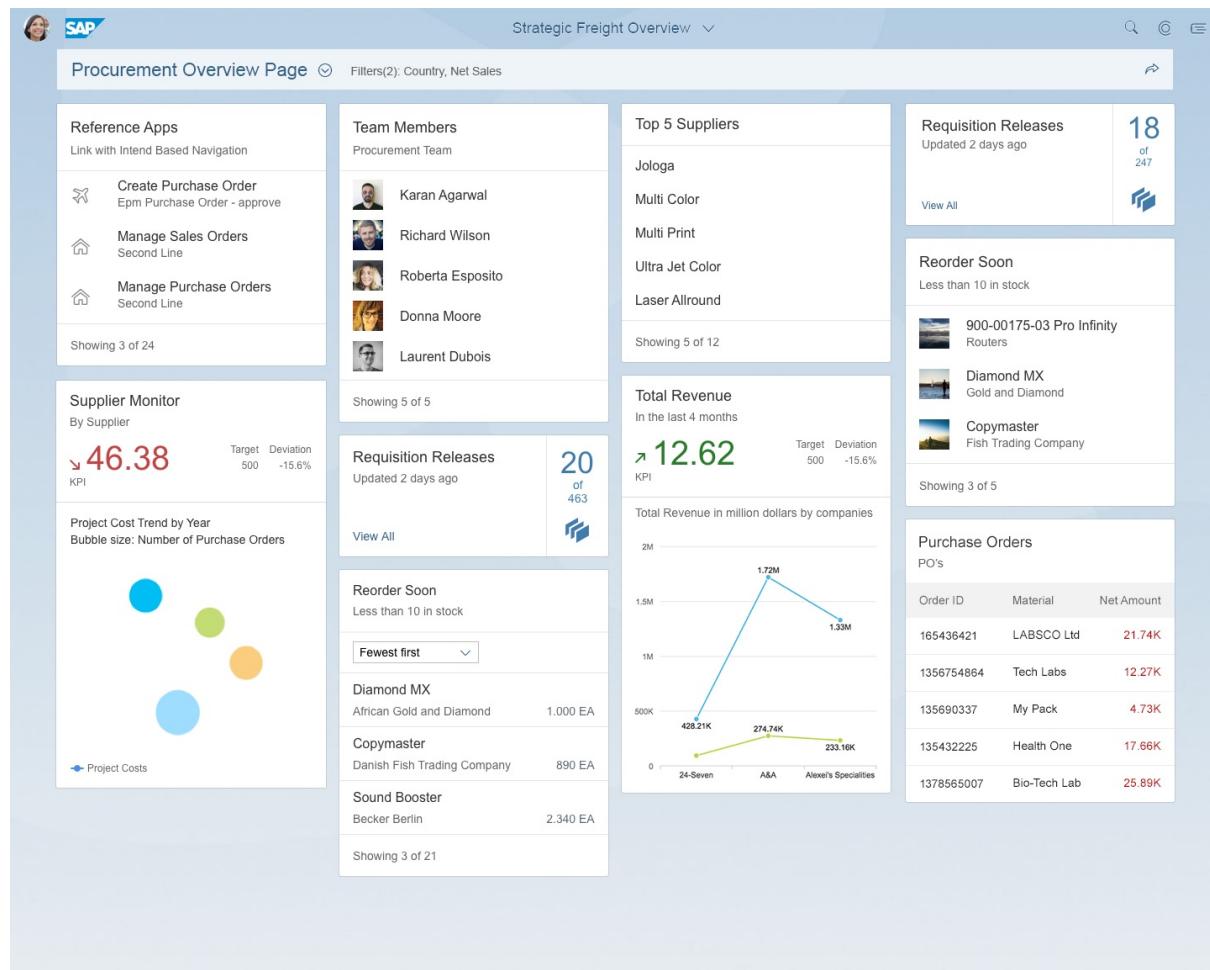
2.1.5 January 2017

Learn about the new and changed features that are available for SAP Web IDE in January 2017.

Overview Page Application Template (New)

The Overview Page application template is now enabled by default in SAP Web IDE.

Use the Overview Page application template to create an overview page for your existing applications. It provides an interactive overview of specific areas, where you can interact with business data and perform key actions.



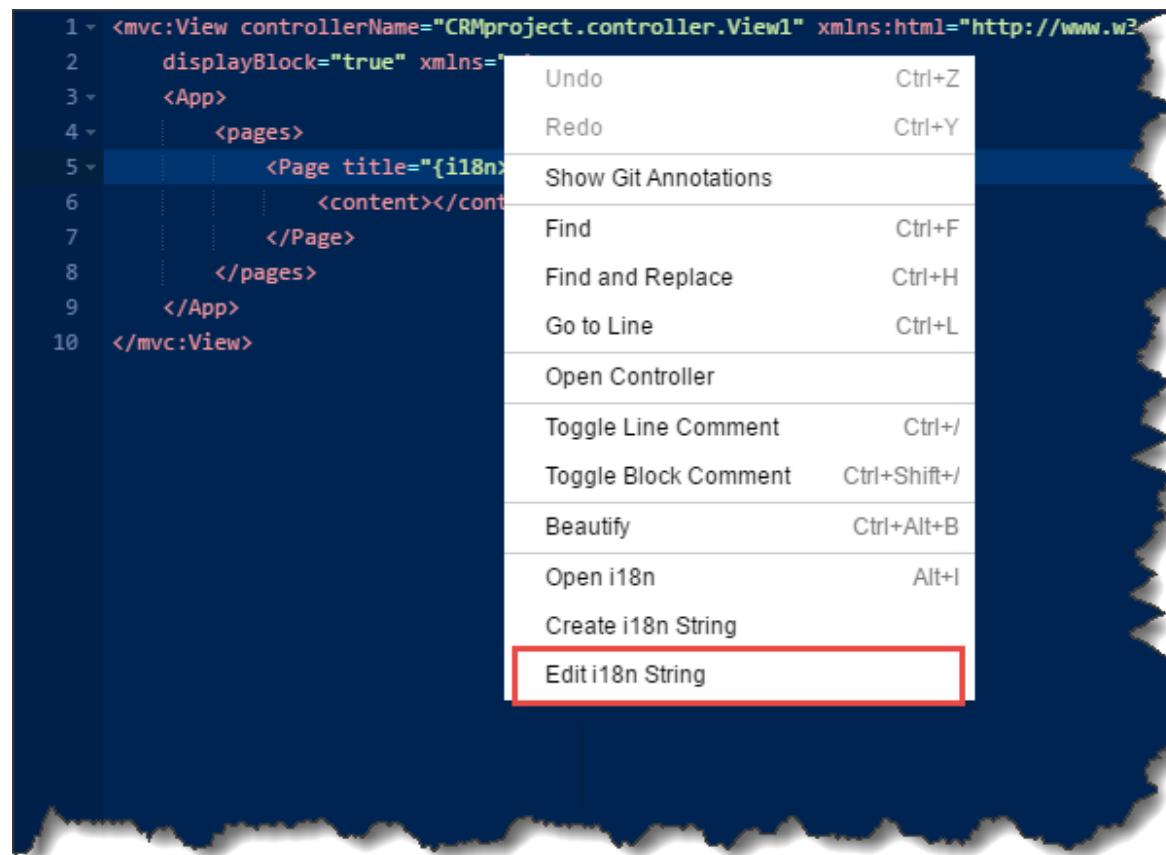
The screenshot displays the SAP Fiori Overview Page application template. At the top, there's a header bar with the SAP logo and a search icon. Below the header, the title "Strategic Freight Overview" is shown, along with a dropdown menu and a refresh button. The main content area is divided into several cards:

- Procurement Overview Page**: A card showing "Filters(2): Country, Net Sales". It includes a "Reference Apps" section with links to "Create Purchase Order" and "Manage Sales Orders".
- Supplier Monitor**: A card showing a KPI of **46.38** (down from 500, deviation -15.6%). It also shows a bubble chart for "Project Cost Trend by Year" with bubbles of varying sizes representing the number of purchase orders.
- Team Members**: A card for the "Procurement Team" showing five team members: Karan Agarwal, Richard Wilson, Roberta Esposito, Donna Moore, and Laurent Dubois.
- Top 5 Suppliers**: A card listing the top suppliers: Jologa, Multi Color, Multi Print, Ultra Jet Color, and Laser Allround. It shows 5 of 12 results.
- Requisition Releases**: A card showing 20 of 463 releases, updated 2 days ago. It lists three releases: 900-00175-03 Pro Infinity Routers, Diamond MX Gold and Diamond, and Copymaster Fish Trading Company.
- Total Revenue**: A card showing revenue in million dollars over the last 4 months. The total is **12.62** (up from 500, deviation +15.6%). It includes a line chart with data points: 428.21K, 274.74K, 1.72M, and 1.33M.
- Reorder Soon**: A card showing items with less than 10 in stock. It lists three items: Diamond MX, Copymaster, and Sound Booster.
- Purchase Orders**: A card showing PO's with columns for Order ID, Material, and Net Amount. It lists six entries: LABSCO Ltd (21.74K), Tech Labs (12.27K), My Pack (4.73K), Health One (17.66K), and Bio-Tech Lab (25.89K).

For more information, see [Developing Apps Using SAP Fiori Elements \[page 325\]](#).

Editing i18n Strings (New)

To edit an existing string, you can now right-click on an i18n binding (for example, {i18n>fileType_title}) and select *Edit i18n String*. Edit the string in the dialog without needing to open the i18n properties file.



For more information, see [Working in the Code Editor \[page 214\]](#).

SAP Web IDE Developer Guide Translation (New)

The documentation is now translated into Japanese and Chinese.

SAP HANA Cloud Platform, Web IDE 開发人员指南

SAP HANA Cloud Platform, Web IDE 開發者ガイド

SAP HANA Cloud Platform、Web IDE

For more information, see [SAP Web IDE Developer Guide \(Japanese\)](#) and [SAP Web IDE Developer Guide \(Chinese\)](#).

SAP Enterprise Portal Plugin (Changed)

The Role Viewer sample application is provided by the SAP Web IDE Enterprise Portal plugin and is presented as a standalone tile in the SAP Fiori launchpad on the Portal. The tile represents the hierarchical view of a specific role.

Sample Application Selection Confirmation

New Role Viewer Sample Application

Sample Application Selection

Search Category Sort By

SAP Enterprise Portal App Sort by name

Role Viewer Sample Application

Sample application used for training and development.

The sample app can be edited in the SAP Web IDE and deployed to the respective SAP system.

For more information, see [SAP Enterprise Portal Plugin for SAP Web IDE](#).

Parent topic: What's New in Previous Releases of SAP Web IDE [page 7]

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)
- [SAP Web IDE 1.17 \[page 99\]](#)
- [SAP Web IDE 1.16 \[page 103\]](#)
- [SAP Web IDE 1.15 \[page 106\]](#)
- [SAP Web IDE 1.14 \[page 109\]](#)

[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.6 December 2016

Learn about the new and changed features that are available for SAP Web IDE in December 2016.

For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

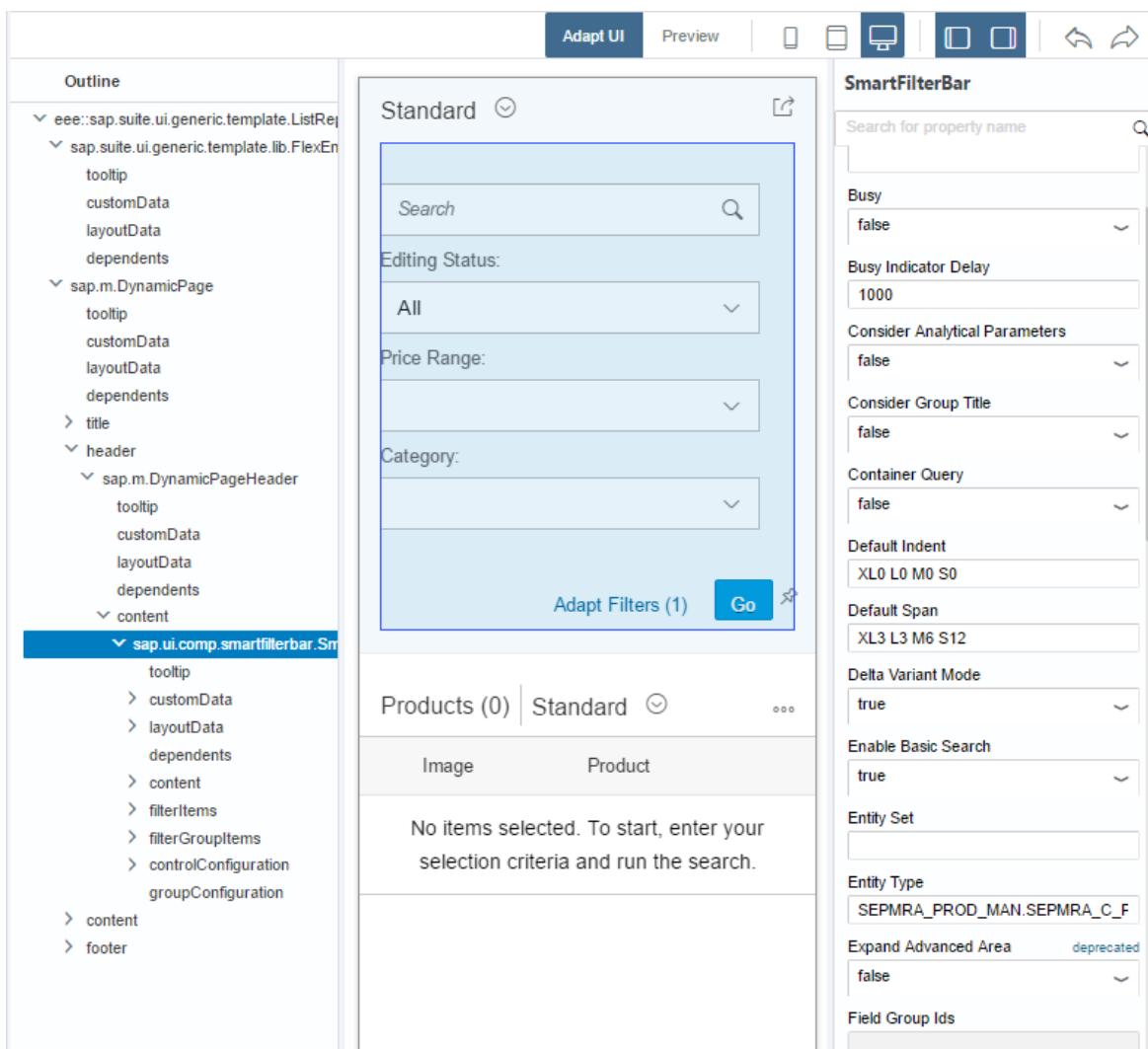
SAP Web IDE No Longer Freezes While Debugging (Changed)

SAP Web IDE no longer freezes when debugging an application in a preview tab. That is, when previewing an application, you can switch back and interact with your SAP Web IDE project (in the original tab) even when debugging the application in the preview tab and stopping at breakpoints.

The first time you preview an application, a popup blocker might appear. Make sure to set the pop-up blocker to let the preview tab open.

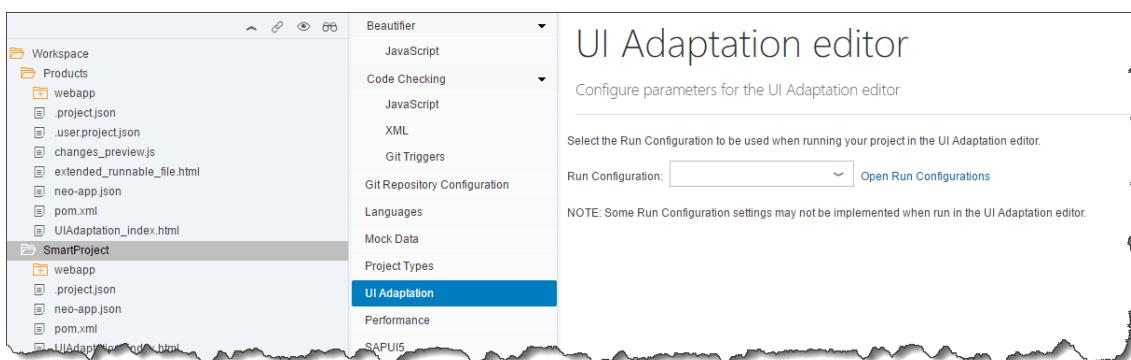
UI Adaptation (New)

- The UI Adaptation editor provides an intuitive user interface to make changes to SAP Fiori element applications at runtime. For example, you can add, remove, or move fields and group. You can also view the properties of the controls in the application.



For more information, see [UI Adaptation Editor \[page 395\]](#)

- You can now select the Run Configuration to be used when running your project in the UI Adaptation editor.

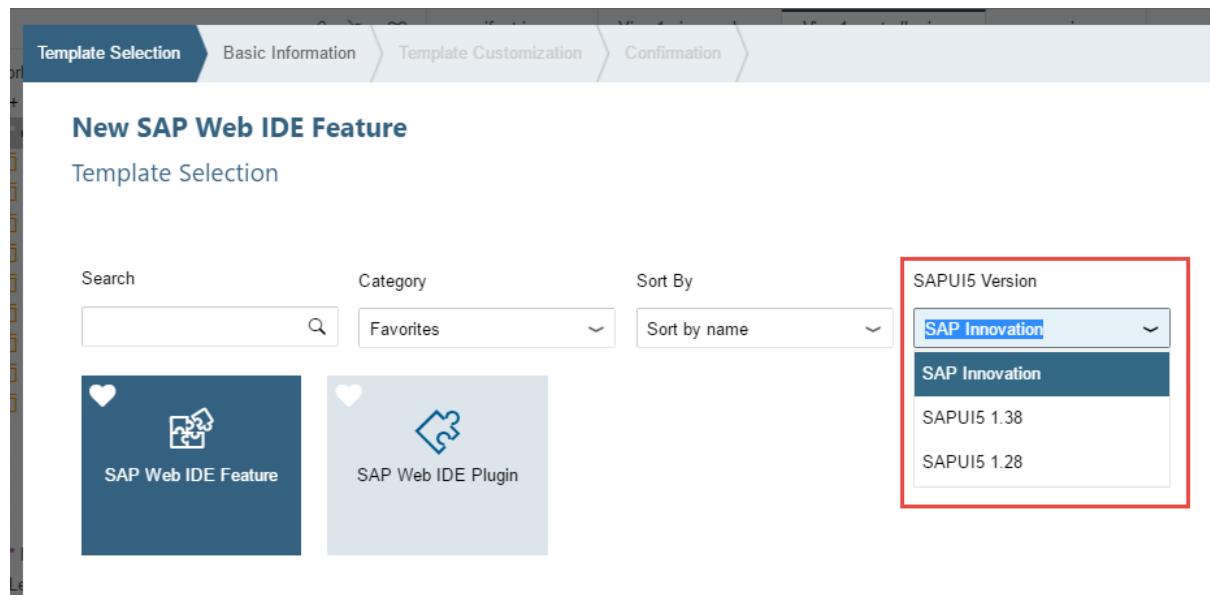


For more information, see [Configure Run Configurations for the UI Adaptation Editor \[page 202\]](#).

SAPUI5 Filter for Templates (New)

You can now use the filter *SAPUI5 Version* in the *New Project from Template* wizard to show only those SAPUI5-based templates that work with that selected version. Templates that do not depend on SAPUI5 are shown no matter the value of this filter.

The selected value will be used as the SAPUI5 version of the project for design time and runtime. If the innovation version is chosen, the default innovation version is used.



For more information, see [Create Projects from Templates \[page 181\]](#).

Code Completion Based on the Project's SAPUI5 Version (New)

Code completion is now dependent on the SAPUI5 version and libraries selected in the project settings. Previously, code completion was based on all the libraries in the latest SAPUI5 version.

The libraries defined in the project settings reflect the SAPUI5 dependencies in the manifest.

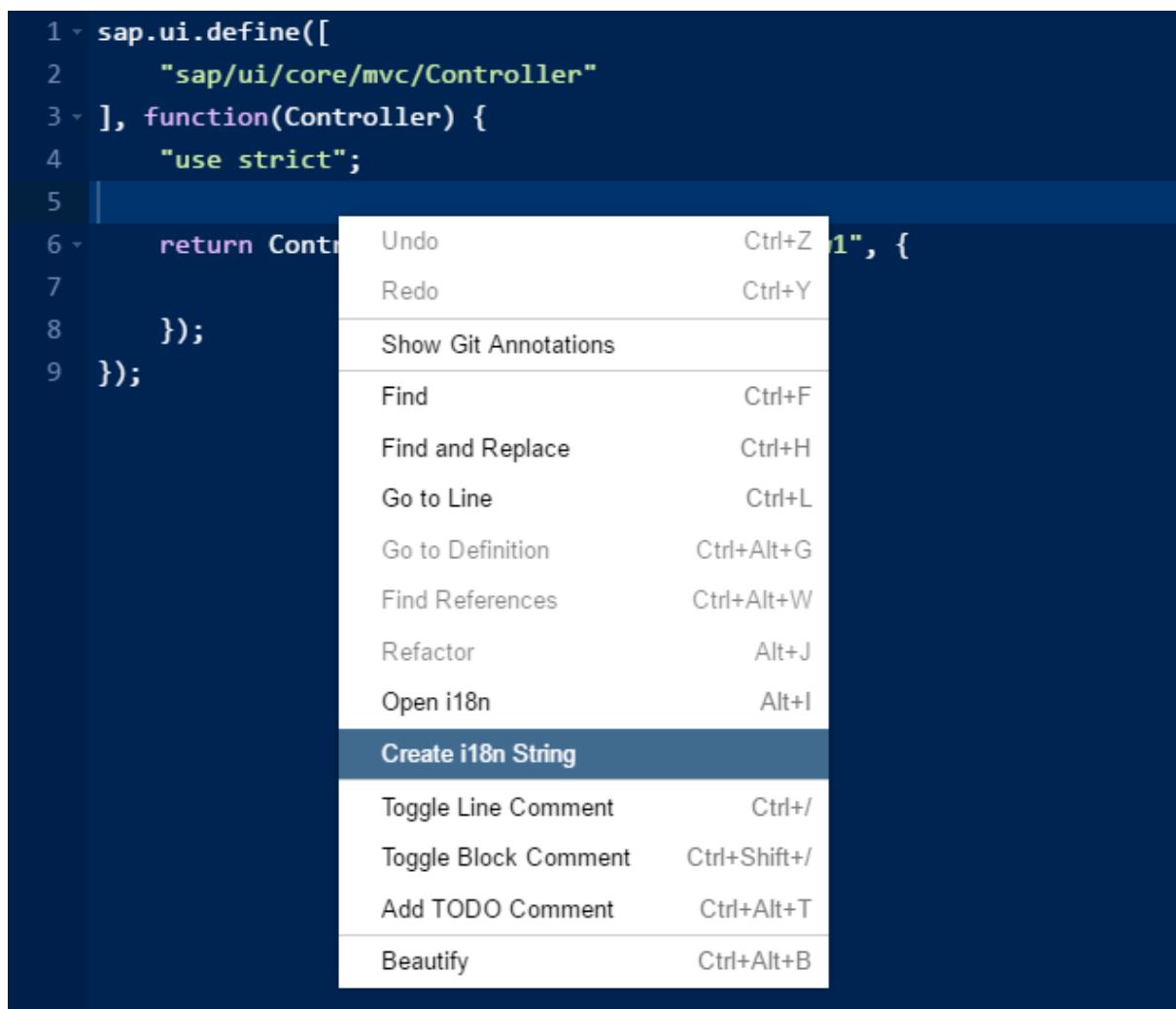
The screenshot shows the SAPUI5 Settings page. On the left, a sidebar menu lists various settings categories: Beautifier, JavaScript (selected), Code Checking, JavaScript (selected again), XML, Git Triggers, Git Repository Configuration, Languages, Mock Data, Project Types, Performance, and SAPUI5 (selected). The main content area is titled "SAPUI5 Settings" and contains a sub-header "Select the SAPUI5 version and libraries for this project". It features a dropdown for "SAPUI5 Version" set to "latest (1.44.2)" and a search bar for "SAPUI5 Libraries" with a magnifying glass icon. A list of libraries is displayed below, with "sap.m" checked.

SAPUI5 Library
<input type="checkbox"/> sap.apf
<input type="checkbox"/> sap.ca.scfld.md
<input type="checkbox"/> sap.ca.ui
<input type="checkbox"/> sap.chart
<input type="checkbox"/> sap.collaboration
<input type="checkbox"/> sap.f
<input type="checkbox"/> sap.fiori
<input type="checkbox"/> sap.gantt
<input type="checkbox"/> sap.landviz
<input checked="" type="checkbox"/> sap.m

For more information, see [Using Code Completion \[page 220\]](#).

Creating i18n String (New)

You can now create a new i18n string by selecting from the context menu *Create i18n String*, and entering the string, its key and other information about the string. The string is added to the selected i18n file. You no longer have to find and open the i18n file and manually add the string.



For more information, see [Working in the Code Editor \[page 214\]](#).

Navigating to i18n File (New)

From the context menu in the JavaScript and XML code editors, you can now select [Open i18n](#) and navigate to the i18n file. If there are more than one i18n file, a dialog asks which one you want to open. If you right-click on an i18n binding and select [Open i18n](#), the corresponding i18n file opens and the specific string is selected.

```

<mvc:View controllerName="LeaveRequest.controller.View1" xmlns:html="http://www.w3.org/1999/xhtml">
  <App>
    <pages>
      <Page title="{i18n>title}">
        <content></content>
      </Page>
    </pages>
  </App>
</mvc:View>

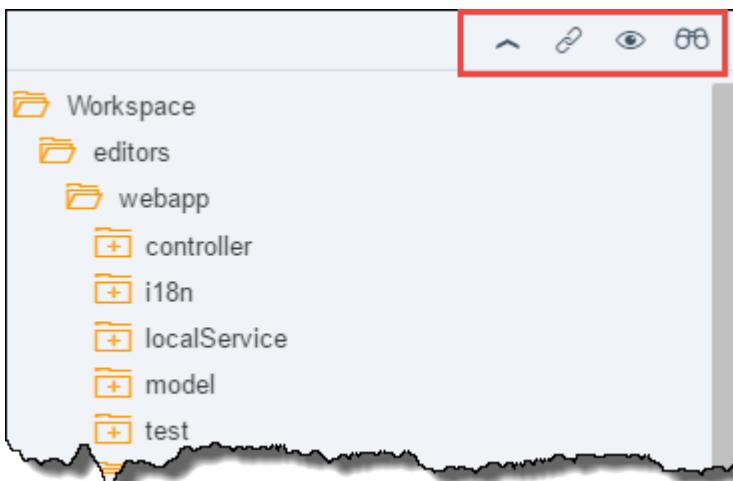
```

Workspace Buttons (New)

The top of the workspace has two new buttons for doing the following:

🔗: Linking/unlinking the workspace with the editor. For more information, see [Work with Files and Folders \[page 176\]](#).

👁️: Showing/hiding hidden files



SAP Web IDE Plugins (New/Changed)

- **SAP Global Track and Trace Plugin for SAP Web IDE (New)**

The SAP Global Track and Trace plugin allows you to create, modify, and delete metadata models for SAP Global Track and Trace, and save them either in a temporary workspace or in a GIT repository you define. After finishing the metadata definition, you can download the metadata model to a local environment and then upload it to the runtime repository. This is a microservice on the target runtime environment and provides data access APIs to other components or microservices.

For more information, see [SAP Global Track and Trace Plugin for SAP Web IDE](#).

- **SAP Web IDE IoT Application Enablement (New)**

The IoT Application Enablement plugin enables you to create IoT applications from scratch using a wizard-based approach. The wizard also includes the IoT Application Enablement UI reuse components that can also be used in your own code without the wizard.

For more information, see [IoT Application Enablement Reuse Components and Templates](#).

- **SAP MII Content Management Plugin for SAP Web IDE (New)**

The SAP MII Content Management plugin helps to 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.

For more information, see [SAP MII Plugin for SAP Web IDE](#).

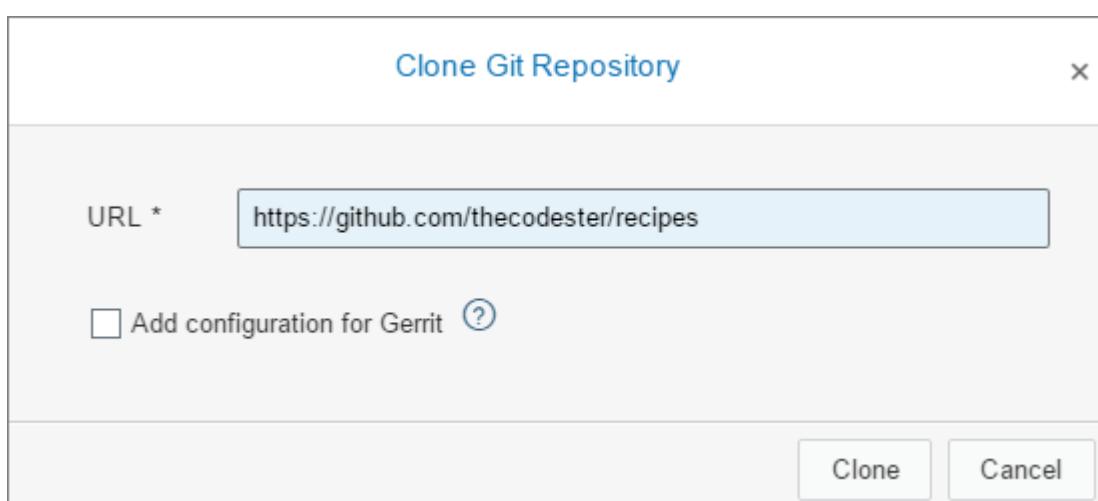
- **OData Model Editor Plugin (Changed)**

When creating an OData model using the Service Catalog, you must now select a product to see the details of a service hosted by an API Management system.

For more information, see [Creating an OData Model from Service Catalog](#).

Git (New/Changed)

- When cloning a Git repository, the dialog is simplified so that you only need to supply the clone URL



- You can now specify that your Git system works with Gerrit, and have any push go through Gerrit. When cloning a repository or when setting a remote repository for a local repository, you can select the [Add configuration for Gerrit](#) checkbox to have Git work with Gerrit.

Annotation Modeler (Changed)

- In the annotation modeler you can now change the sequence of items within collections (complex collection records) using the *Move Up* or the *Move Down* button. See, [Change the Order of Annotations and Collection Items \[page 387\]](#).
- The function *Copy to Local Annotations* is now called *Clone for Overriding*. See, [Edit External Annotations \[page 390\]](#).
- Editing of the terms *Units and Currencies* does now match the standard pattern.
- To better match the annotation structure of the XML, the annotation structure in the *Annotation Structure* pane includes additional nodes for some terms in the tree.

The screenshot shows the SAP Annotation Modeler interface with the 'Annotation Structure' pane open. The pane lists annotations for the 'SEPMRA_C_PD_Product' entity. It shows a hierarchy of nodes like 'Local Annotations', 'UI.Facets', 'UI.CollectionFacet', 'Facets', 'UI.ReferenceFacet', 'UI.ReferenceFacet', 'UI.FieldGroup', and 'Data'. Some nodes have their children expanded, and specific nodes ('Facets' and 'Data') are highlighted with red boxes.

- Adaptations in the annotation modeler design.
- Option to set the annotation modeler as a default editor for annotation files, so that they are opened in the annotation modeler by default (on double-click on file name).
- New *Code Editor* tab for easy switching between the relevant editors.
- Improved distinction between local and external annotations in the *Annotation Structure* pane.

For more information, see [Annotation Modeler \[page 369\]](#).

SAP Web IDE SDK (Changed)

As of this release, the topics that had been in the *Extend SAP Web IDE* chapter in this developer guide are now located in the SAP Web IDE SDK.

- Under *Client-Side Plugin Programming Guidelines*:
 - *Plugin Concept*
 - *Template Concept* and related subtopics
- Under *Tutorials*:
 - *Creating a New Template from Scratch*

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

Layout Editor (Changed)

You can now define a data entity set for a view from the catalog viewer by clicking the data binding button.

This enables you to see all available data sets together with their details.

Language Files Are Hidden (Changed)

Language files, which contain the translated texts for specific languages, are now hidden by default. You can see them by showing hidden files, either with the new  icon at the top of the workspace, or by choosing

► [View > Show Hidden Files](#) ▶.

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)
- [SAP Web IDE 1.17 \[page 99\]](#)
- [SAP Web IDE 1.16 \[page 103\]](#)
- [SAP Web IDE 1.15 \[page 106\]](#)
- [SAP Web IDE 1.14 \[page 109\]](#)
- [SAP Web IDE 1.13 \[page 111\]](#)
- [SAP Web IDE 1.12 \[page 114\]](#)
- [SAP Web IDE 1.11 \[page 118\]](#)

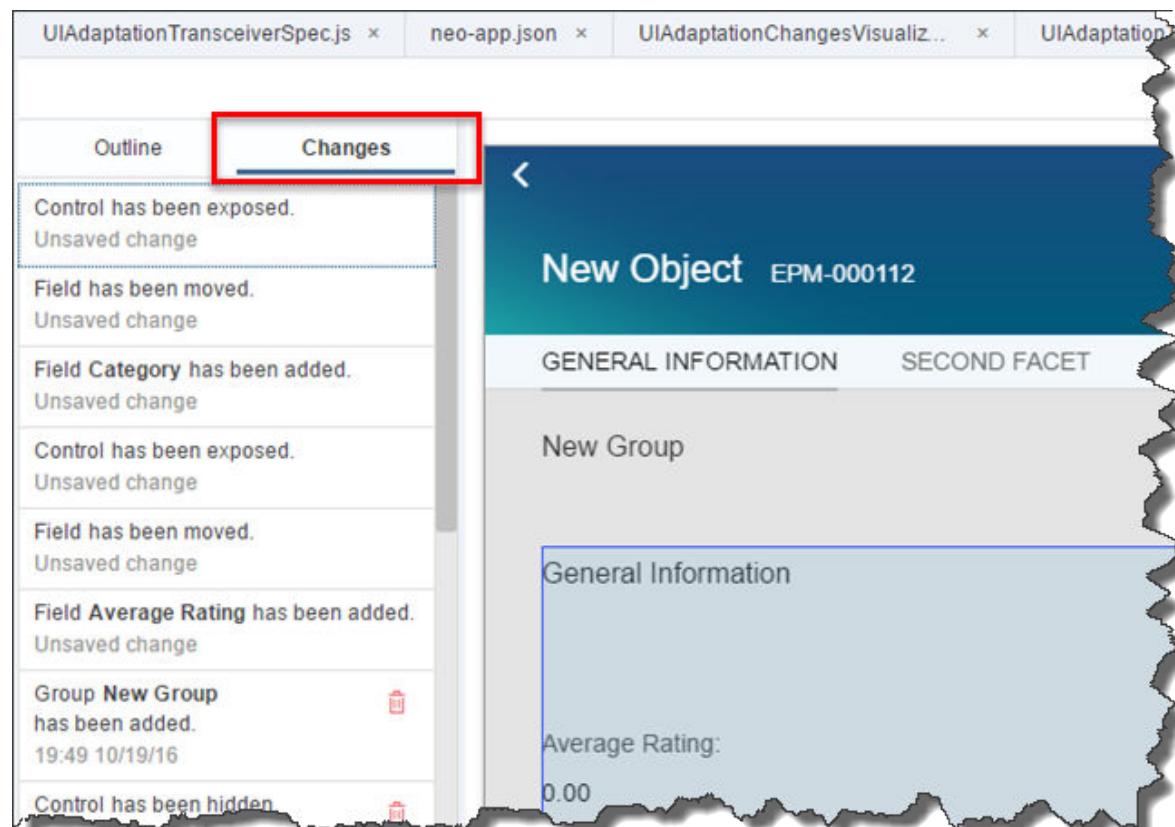
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.7 November 2016

Learn about the new and changed features that are available for SAP Web IDE in November 2016

Changes Tab in the UI Adaptation Editor (New)

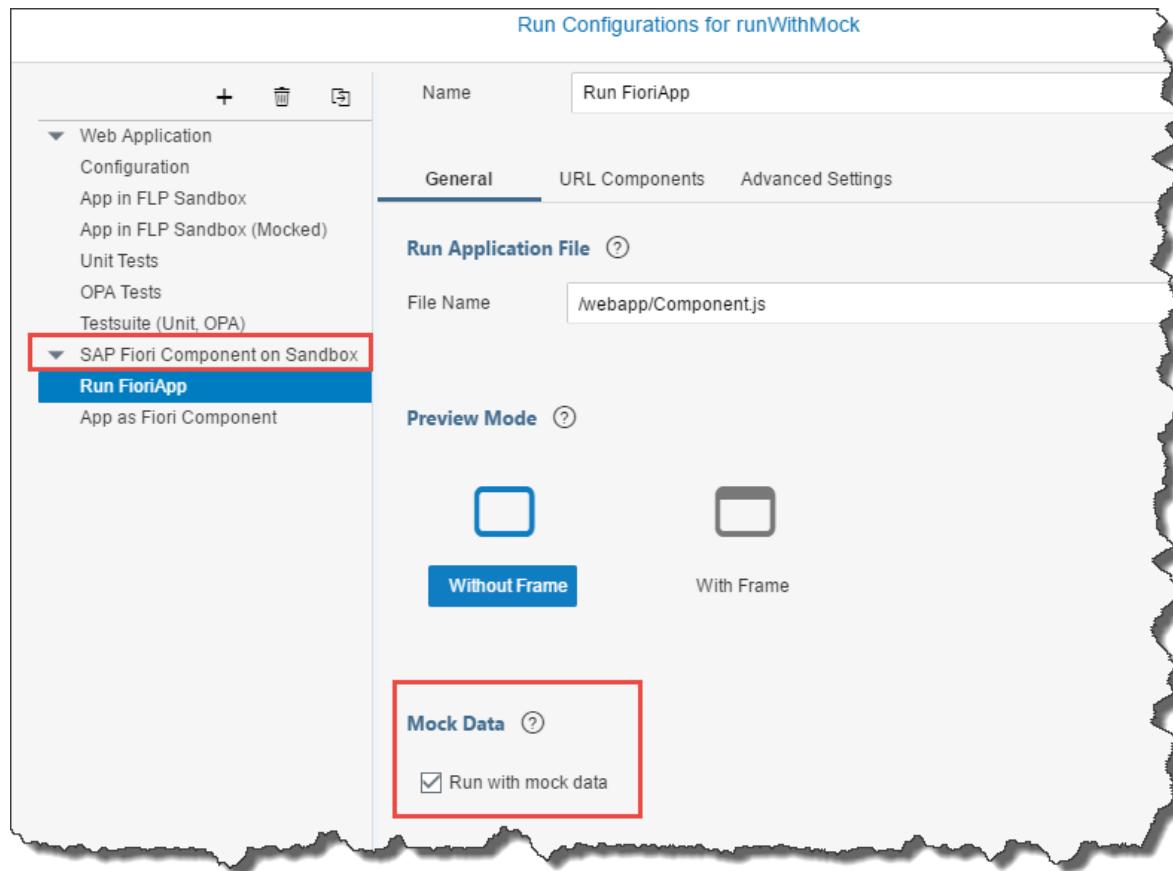
You can now open the [Changes](#) tab to see the changes made to your project using the UI Adaptation editor. You can also remove the changes by deleting them from the [Changes](#) tab.



For more information, see [Change Applications with the UI Adaptation Editor \[page 397\]](#).

Run SAP Fiori Component on Sandbox with Mock Data (Changed)

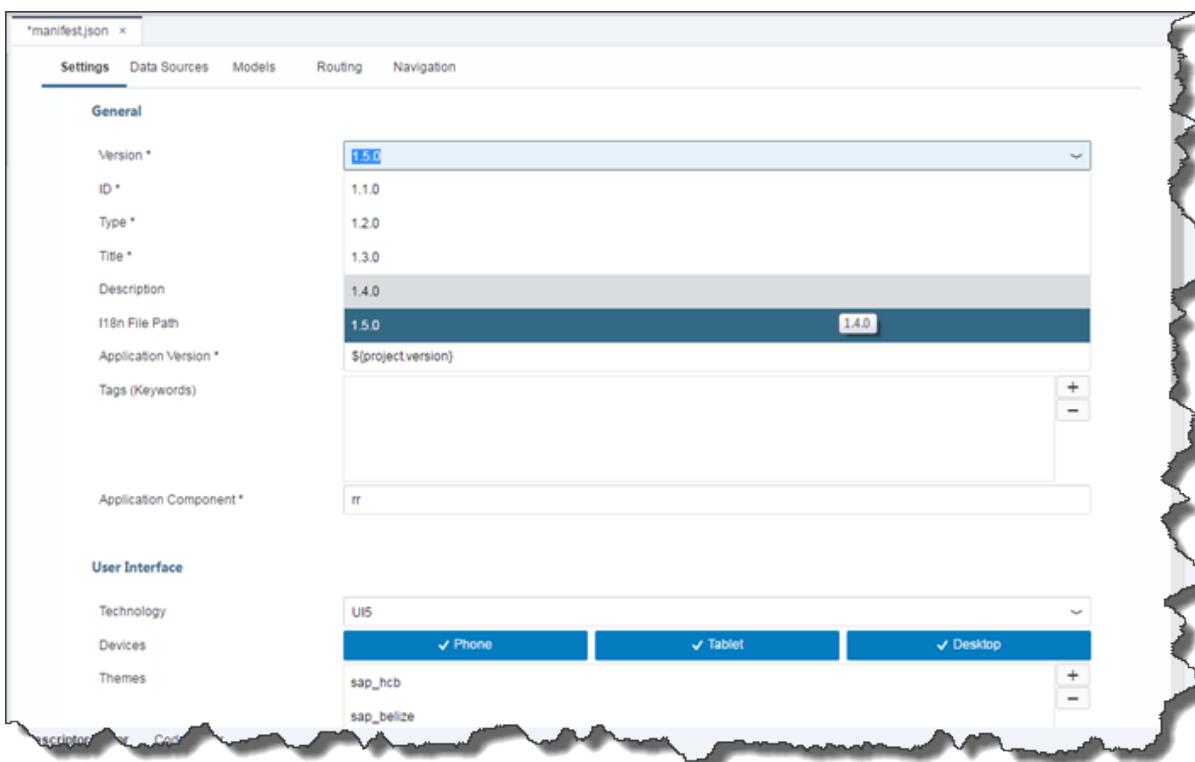
You can now run applications with the SAP Fiori Component on Sandbox configuration using mock data by selecting the relevant checkbox from the Run Configurations page.



For more information, see [Run Applications with Mock Data \[page 443\]](#).

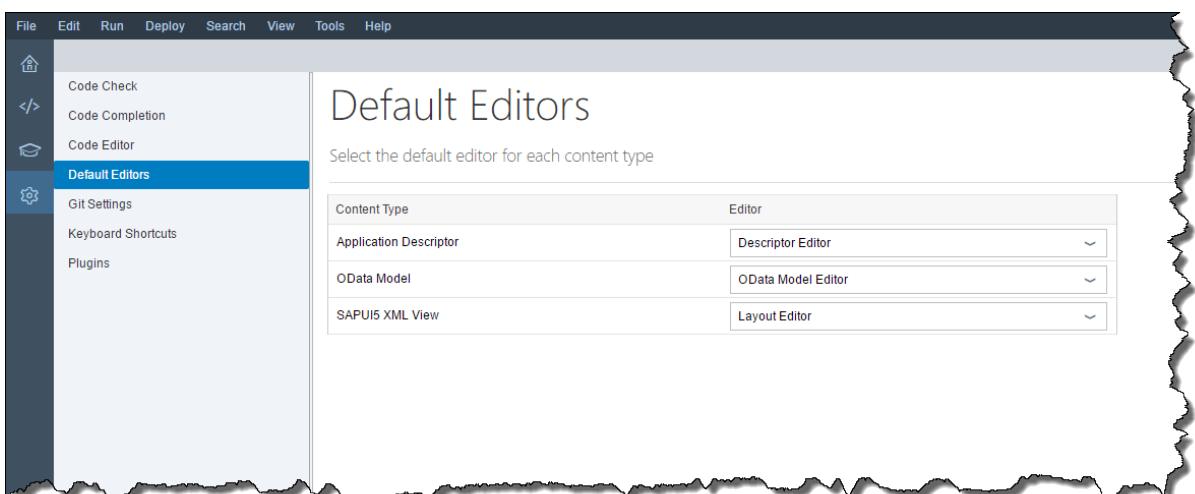
Selecting the Application Description Version (New)

You can now select the desired version of the Application Description schema from a list of supported versions. We recommend you select the latest version.



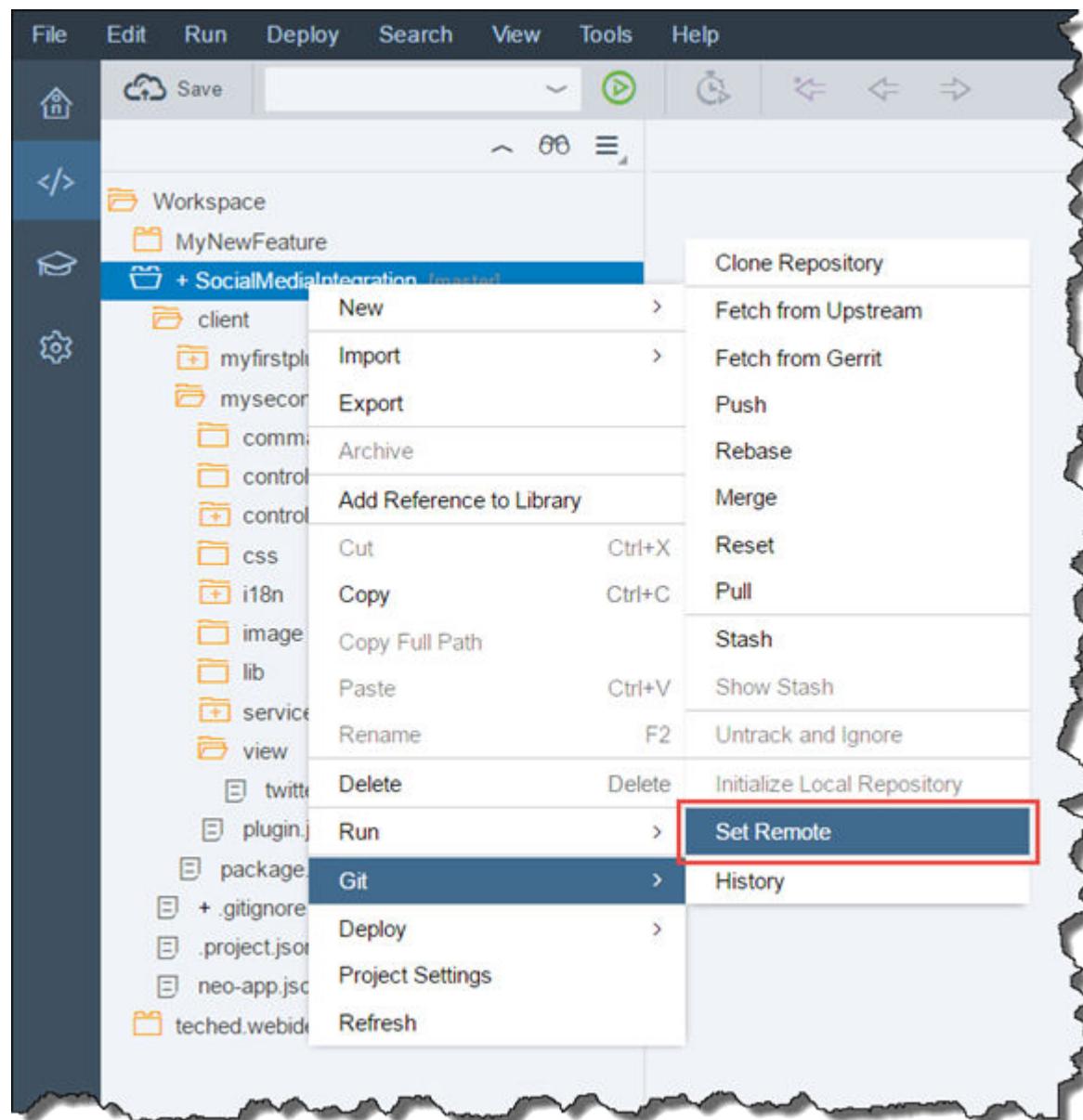
Selecting Default Editors (New)

You can now select the default editor that will be opened when double-clicking on a selected file type, for example, whether to open a view with the code editor or the layout editor. Go to [Tools > Preferences > Default Editors](#).



Setting a Remote Git Repository (Changed)

After initializing a local Git repository, select **File > Git > Set Remote** to connect the project to a remote Git repository. This is now a separate action from initializing the local Git repository.

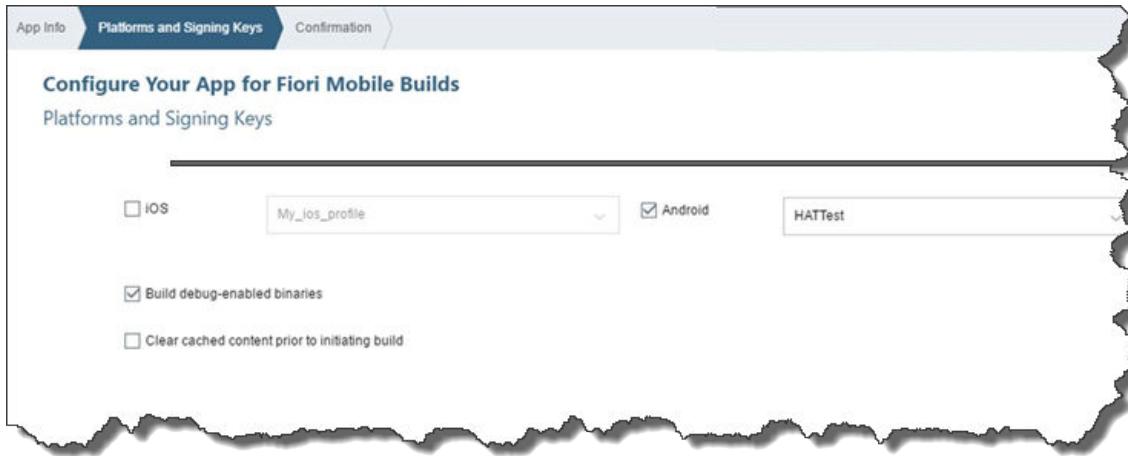


For more information, see [Using Source Control \(Git\) \[page 402\]](#).

New Hybrid App Toolkit Features

- Support for Building and Deploying Apps Using SAP HANA Cloud Platform, Mobile Service for SAP Fiori (New)

Hybrid App Toolkit is now part of the Fiori Mobile developer experience. The hybrid apps that you develop using SAP Web IDE can now be built and deployed through Fiori Mobile build. You can then download and install these apps from SAP Mobile Place.



You can still build your hybrid apps locally if the local build-related features are enabled. To enable these features, go to ► *Tools* ► *Preferences* ▾ and select the *Enable Local Add-On features* check box.

- **Support for Building List Report Applications (formerly known as Smart Template applications)**
Using SAP HANA Cloud Platform, Mobile Service for SAP Fiori (Fiori Mobile), you can now create applications using the List Report application template and then build them using Fiori Mobile build.
- **Support for New Cordova Plugin**
The Code Completion function, Mobile Development Pane, and Cordova Facade preview support a new Cordova plugin called *Printer* plugin.

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)

[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.8 October 2016

Learn about the new and changed features that are available for SAP Web IDE in October 2016.

Support for Building and Deploying Apps Using SAP HANA Cloud Platform, Mobile Service for SAP Fiori (New)

Hybrid App Toolkit is now part of the Fiori Mobile developer experience. The hybrid apps that you develop using SAP Web IDE can now be built and deployed through Fiori Mobile build. You can then download and install these apps from SAP Mobile Place.

For more information, see [Developing Apps through SAP HCP, Mobile Service for SAP Fiori \(Fiori Mobile\)](#).

You can still build your hybrid apps locally, but it is optional and has to be enabled first. To enable the local build related features, go to and select the *Enable Local Add-On features* check box.

Toggle Buttons for Bottom Pane (New)

Now you can toggle the bottom pane in SAP Web IDE to display the SAP Web IDE console or the Problems view.

- Console toggle button 
- Problems View toggle button 

For more information, see [Navigate SAP Web IDE \[page 165\]](#) and [Keyboard Shortcuts \[page 166\]](#).

Central Rules for JavaScript Validation (New)

An administrator can now upload ESLint rules (`.eslintrc` and `.eslintrc.ext` files) for JavaScript validation to the SAP HANA Cloud Platform account, and those rules will be the default validation rules for all SAP Web IDE projects for all users on that account. Users can still modify the rules for their own SAP Web IDE.

For more information, see [Set JavaScript Rules for All Users \[page 296\]](#).

Connecting to On-Premise Git (New)

You can now work with Git within SAP Web IDE and connect to a remote Git repository within your corporate landscape.

For more information, see [Connect to your Corporate Git System \[page 408\]](#).

Terminology Update (Changed)

The term Smart Template has been changed to SAP Fiori element. From this release, the term *SAP Fiori element* is used in the UI and documentation.

Code Editor (Changed)

- **Go to Line (New)**

In the code editor, you can now select *Go To Line* in the context menu to navigate directly to a specific line number in the code.

For more information, see [Working in the Code Editor \[page 214\]](#).

- **Find and Replace (New)**

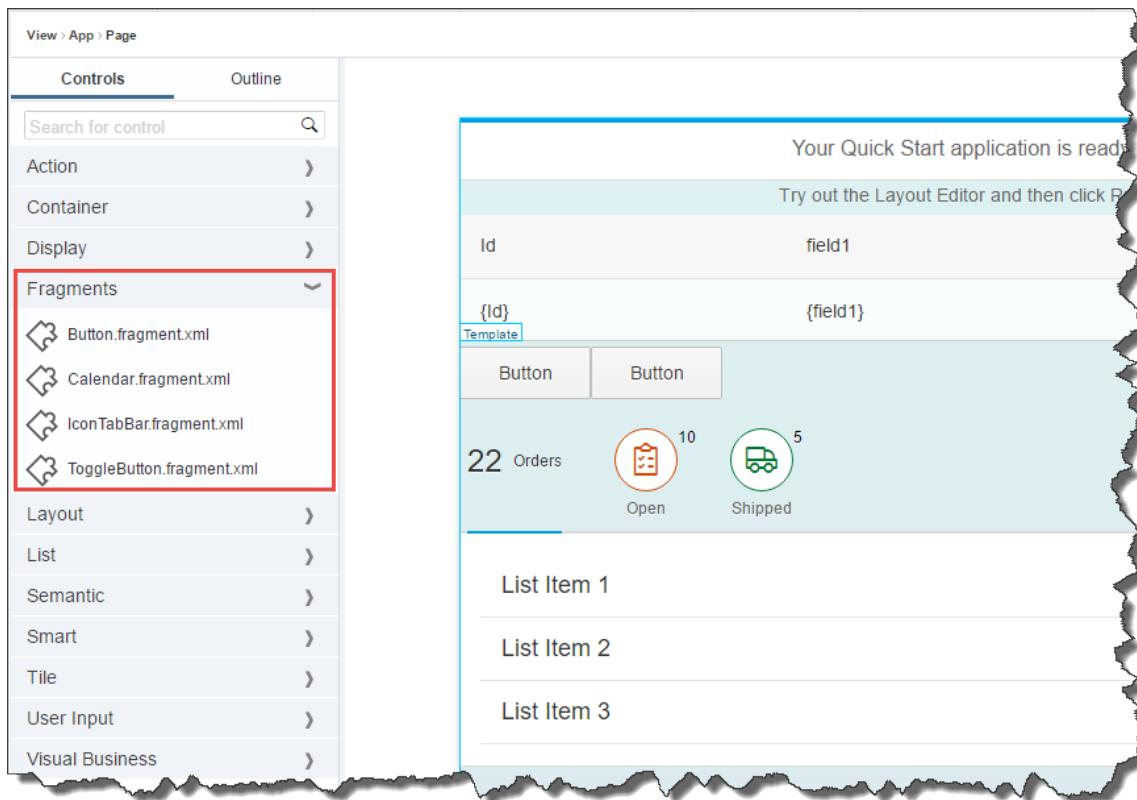
In the code editor, you can now select *Find* or *Find and Replace* in the context menu to search within the code. These options were previously available only from the *Search* menu.

For more information, see [Find and Replace in an Open File \[page 171\]](#).

Layout Editor (Changed)

- **Add fragments from the *Controls* tab**

A new section has been added to the *Controls* tab which contains the fragments used in the project. Only fragments for supported controls are displayed in this section.



You can drag and drop these fragments into the canvas for quick reuse.

- **Mozilla Firefox support**

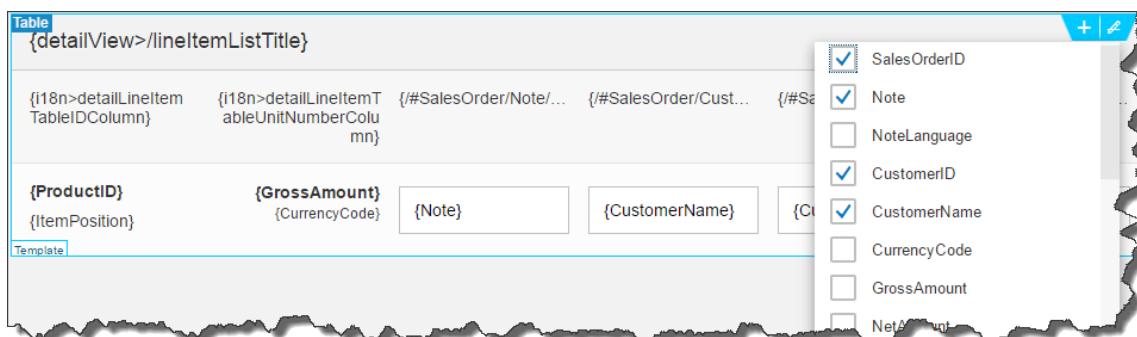
You can now work in the layout editor using the Mozilla Firefox browser.

- **Name your Quick Start App (New)**

When creating a Quick Start App, you are now prompted to provide a name for it.

- **Table Builder (New)**

The new table builder feature allows you to select data fields from your OData service and then automatically adds the relevant columns in the table.



- **Setting SAPUI5 Version (Moved)**

This functionality has now moved to the *Project Settings* pane. A warning message appears in the layout editor if there is a version compatibility issue.

For more information, see [Layout Editor \[page 332\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.9 September 2016

Learn about the new and changed features that are available for SAP Web IDE in September 2016.

SAP Web IDE Personal Edition (New)

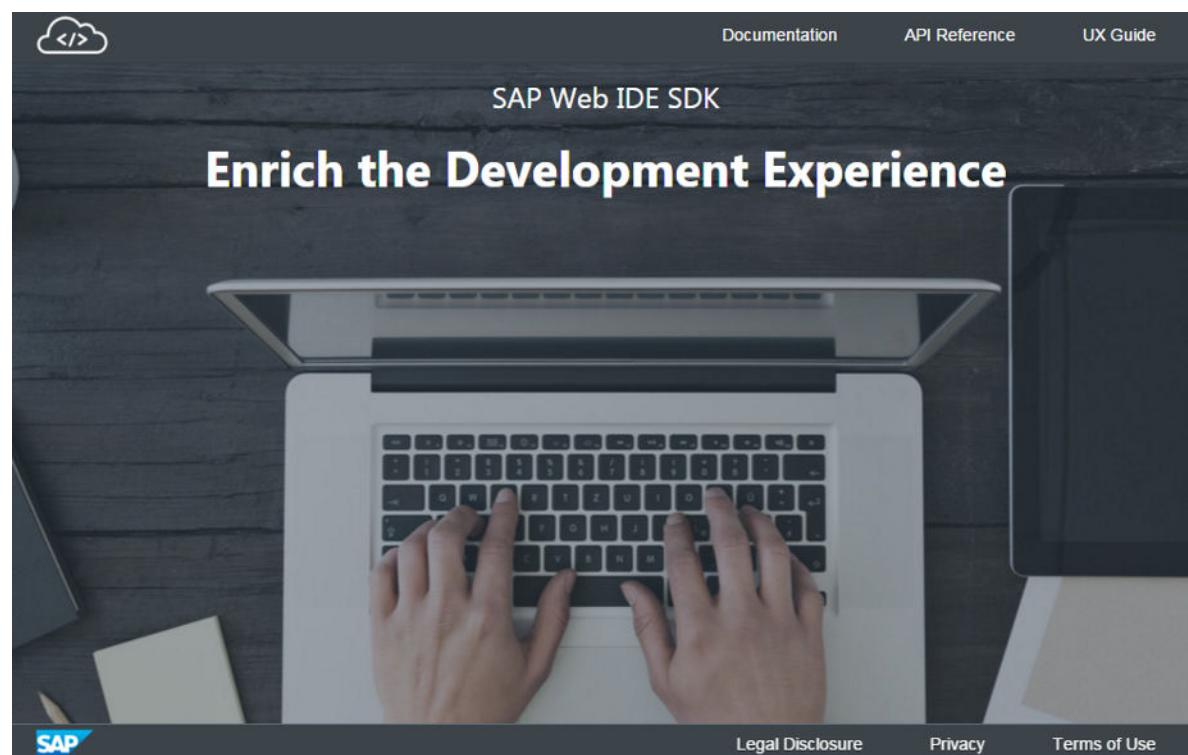
Following strong customer demand, we are happy to announce the release of SAP HANA Cloud Platform, Web IDE personal edition (SAP Web IDE personal edition).

SAP Web IDE personal edition is intended as a complementary IDE. It can be installed on a personal workstation for off-line development by a single developer.

For more information, see [SAP Web IDE Personal Edition \[page 476\]](#).

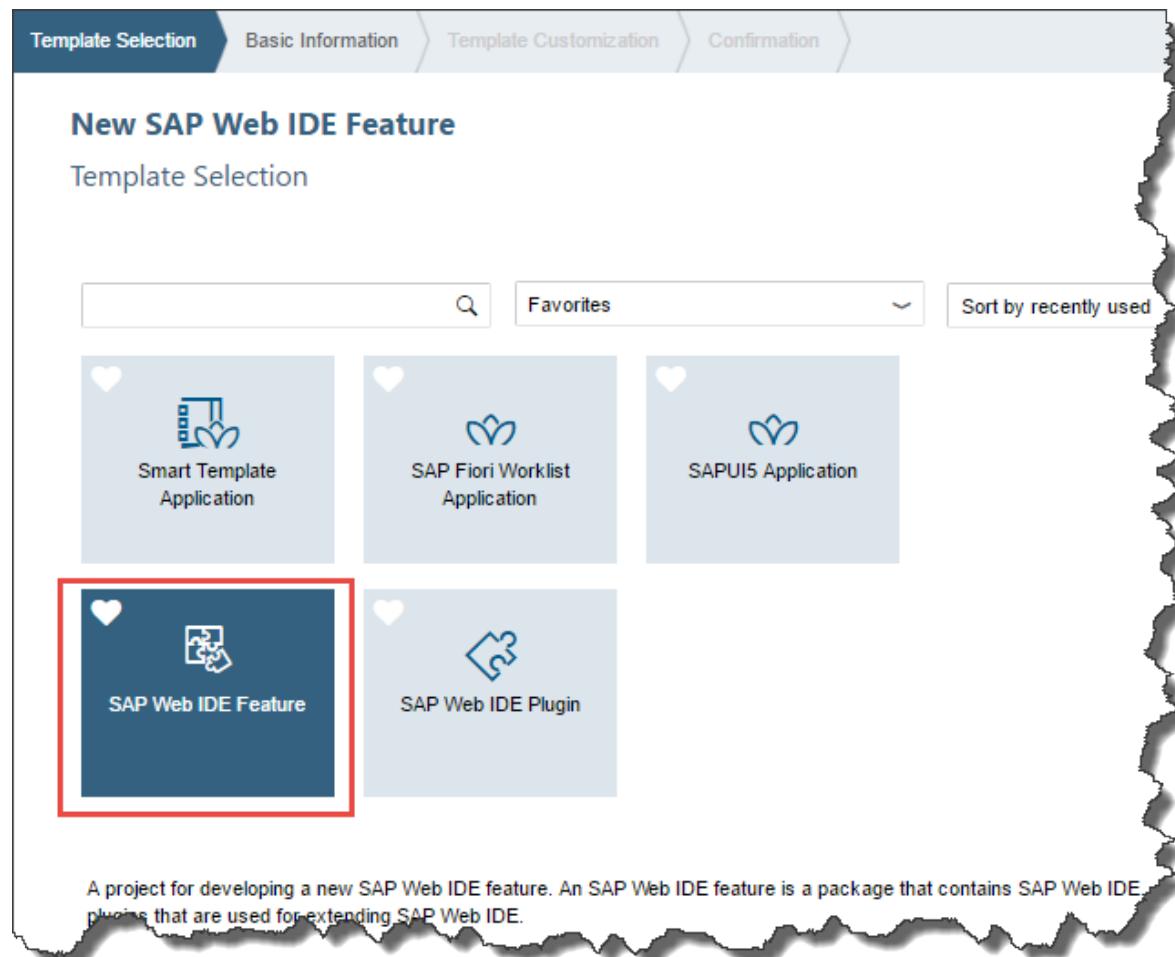
SAP Web IDE SDK App (New)

The new SDK App provides an API Reference and documentation, including tutorials. The SDK helps you develop your SAP Web IDE plugins, including new application templates and features, and integrate tools into SAP Web IDE. For more information, open the SDK app from the [Help](#) menu in SAP Web IDE.



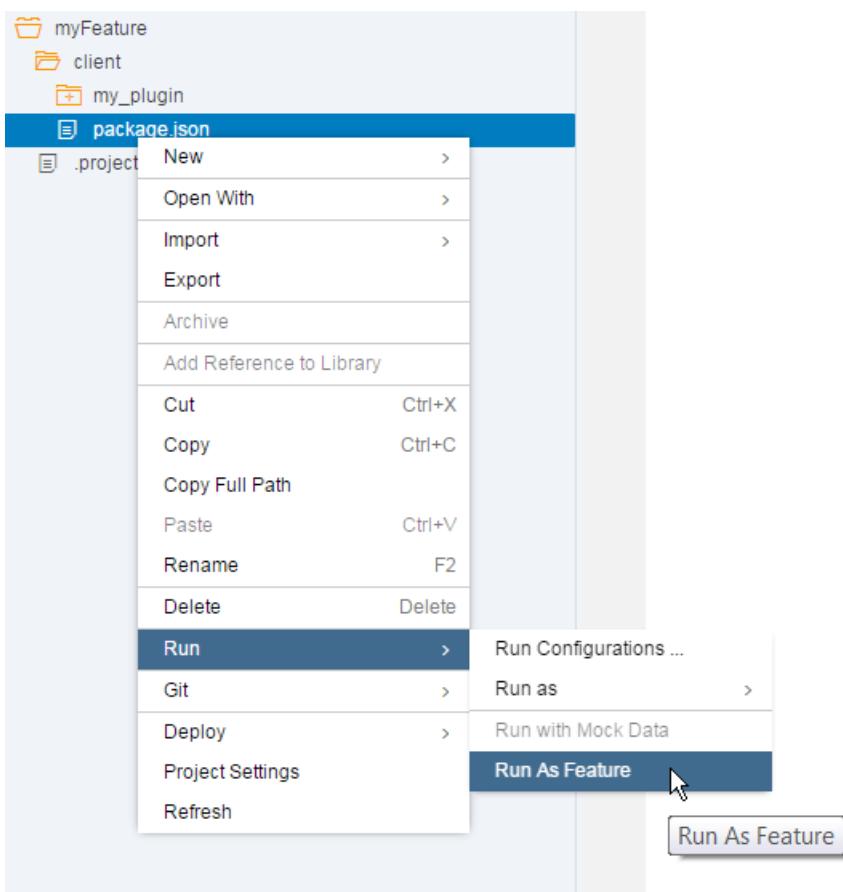
Template for SAP Web IDE Feature Creation (New)

In the [New Project](#) wizard you can now create a new feature project to bundle one or more SAP Web IDE plugins. An SAP Web IDE feature is a package that contains SAP Web IDE plugins that are used for extending SAP Web IDE.



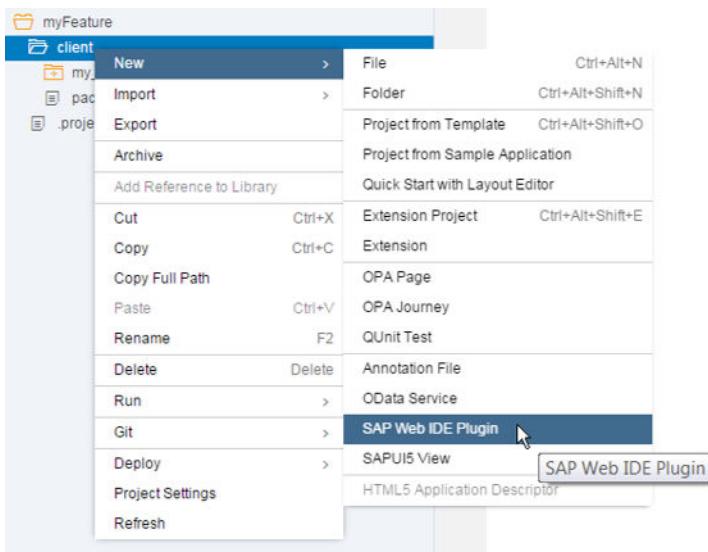
Run as Feature Capability (New)

Now you can preview your feature directly from your SAP Web IDE workspace by right-clicking the package.json file, then choosing [▶ Run ▶ Run as Feature](#). This allows you to test all the bundled plugins in the feature.



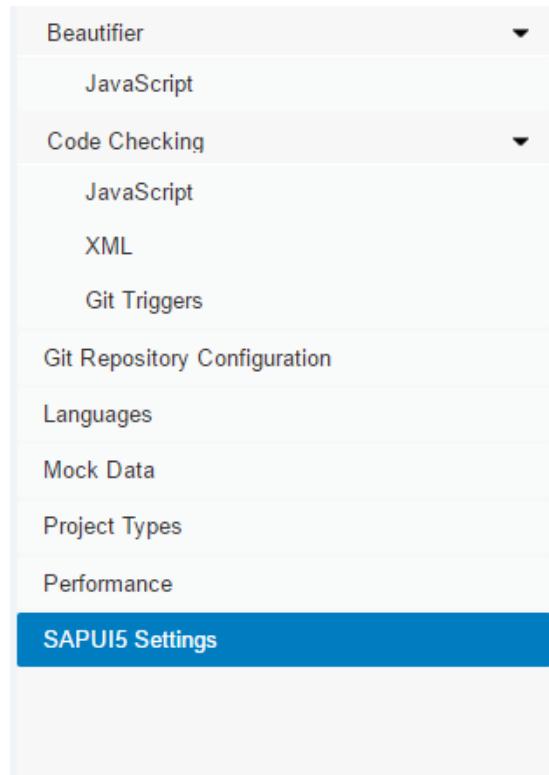
Add a New SAP Web IDE Plugin to a Feature (New)

Now you can add additional plugins to a feature using the plugin wizard by right-clicking the `client` folder of a feature and selecting SAP Web IDE Plugin. The `package.json` file gets updated accordingly.



Setting the SAPUI5 Version (New)

The [Project Settings](#) pane now lets you set the SAPUI5 version for your application – this affects design-time tools such as code completion, static code validation and layout editor controls, as well as the default version for new run configurations and the version for runtime.



SAPUI5 Settings

Select an SAPUI5 version for design-time

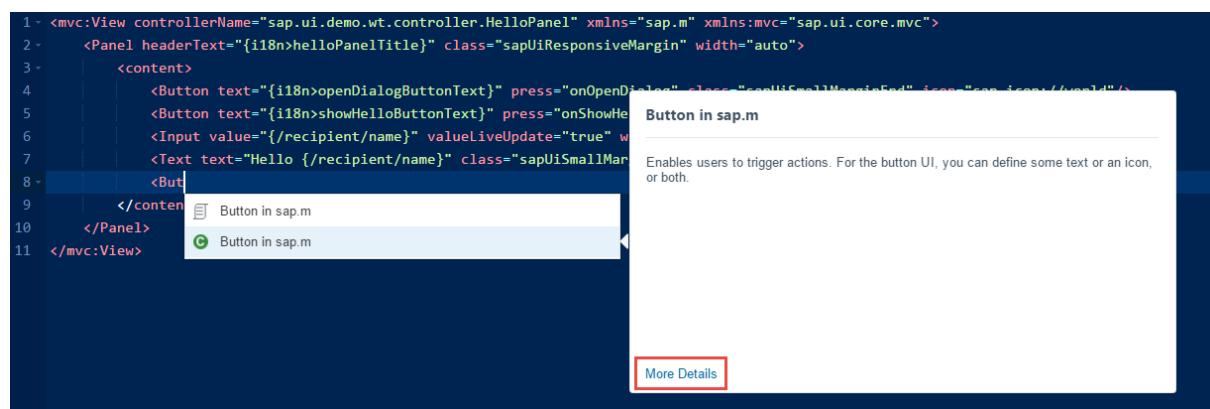
SAPUI5 Version

1.38.7

For more information, see [Set the SAPUI5 Version \[page 201\]](#).

Link to SAPUI5 Documentation in Code Completion (New)

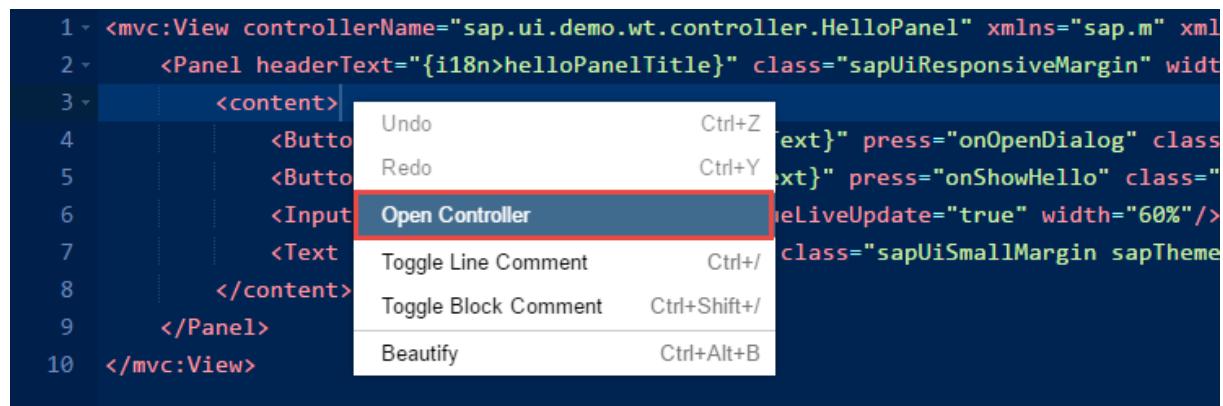
When hovering over an SAPUI5 element during code completion, a tooltip is displayed that shows some of the SAPUI5 documentation for that element. Now, the tooltip also includes a link to the full SAPUI5 documentation for that element.



For more information, see [Using Code Completion \[page 220\]](#).

Navigate from a View to its Controller (New)

In the code editor for an XML view, you can now right-click and select *Open Controller* to open the controller in a code editor. If you right-click an event handler in the view and select *Open Controller*, you are taken to the event handler function.



For more information, see [Working in the Code Editor \[page 214\]](#).

Models Tab in Application Descriptor Editor (New)

The application descriptor editor now includes a *Models* tab for setting the SAPUI5 models in your project, either OData models or other models such as XML, JSON or i18n resource models. You can create and edit the models, for example, change the data source for an OData model.

A screenshot of the SAP Web IDE Application Descriptor Editor. At the top, there is a navigation bar with tabs: Settings, Data Sources, Models (which is highlighted with a red box), Routing, and Navigation. Below the navigation bar, the word 'Models' is displayed in bold blue text. Underneath, a section titled 'Define model services for the application and configure their properties:' contains a list box with two items: 'i18a' and 'invoice'. To the right of the list box are two buttons: a plus sign (+) and a minus sign (-). The entire interface has a clean, modern design with a light gray background and blue highlights for active tabs and sections.

For more information, see [Models Tab Options \[page 208\]](#).

Refactor Function Parameters (New)

In the code editor, you can now refactor JavaScript function parameters, in addition to function names and variables. Change the parameter name by right-clicking the parameter and selecting *Refactor*. All references to the parameter are automatically updated.

For more information, see [Working in the Code Editor \[page 214\]](#).

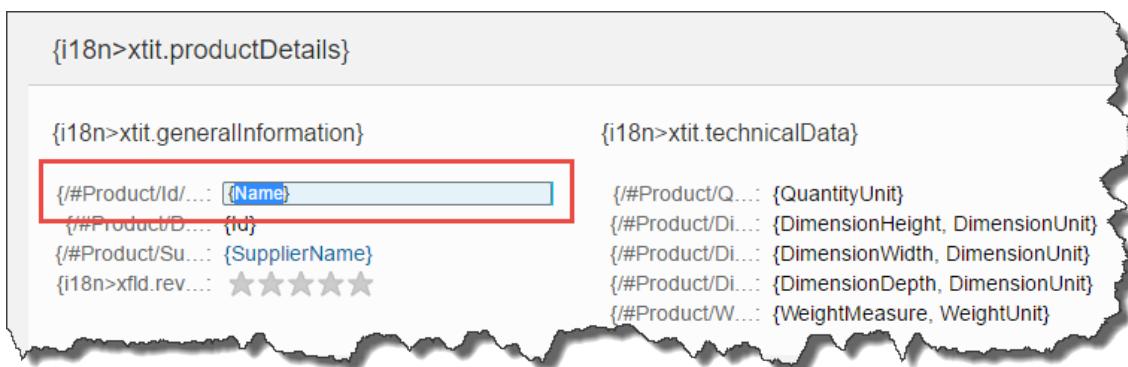
Layout Editor (Changed)

- **Layout Editor Available in Internet Explorer 11 (New)**

You can now work in the layout editor when using Internet Explorer 11.

- **Inline Editing (New)**

You can now edit the strings of a control from the canvas. This includes strings that are bound to a specific model and free-text strings.



- **Selecting (New)**

It is now easier to select the different layers within a view. Press **Ctrl** and click a control. The control's parent is selected. A higher layer is selected each time you click.

For more information, see [Layout Editor \[page 332\]](#).

Empty Plugin Template Renamed (Changed)

The *Empty Plugin* template has been renamed to *SAP Web IDE Plugin*.

API Reference Pane Removed (Removed)

The *API Reference* pane, which provided SAPUI5 information from a pullout pane from the right side, has been removed. The API reference information is now displayed in a tooltip when hovering over an SAPUI5 element.

Parent topic: What's New in Previous Releases of SAP Web IDE [page 7]

Related Information

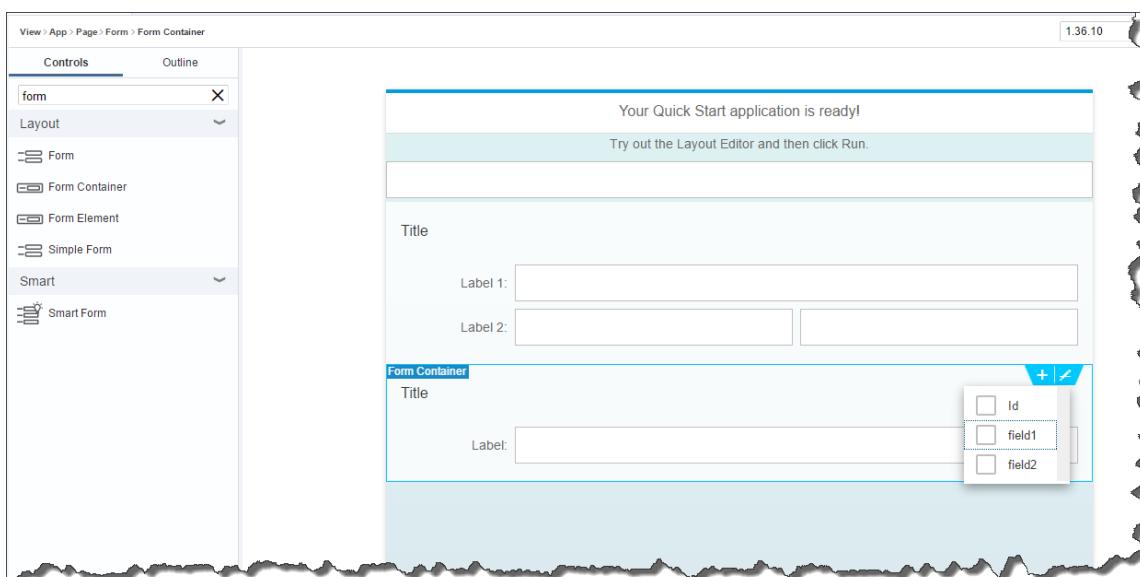
[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.10 August 2016

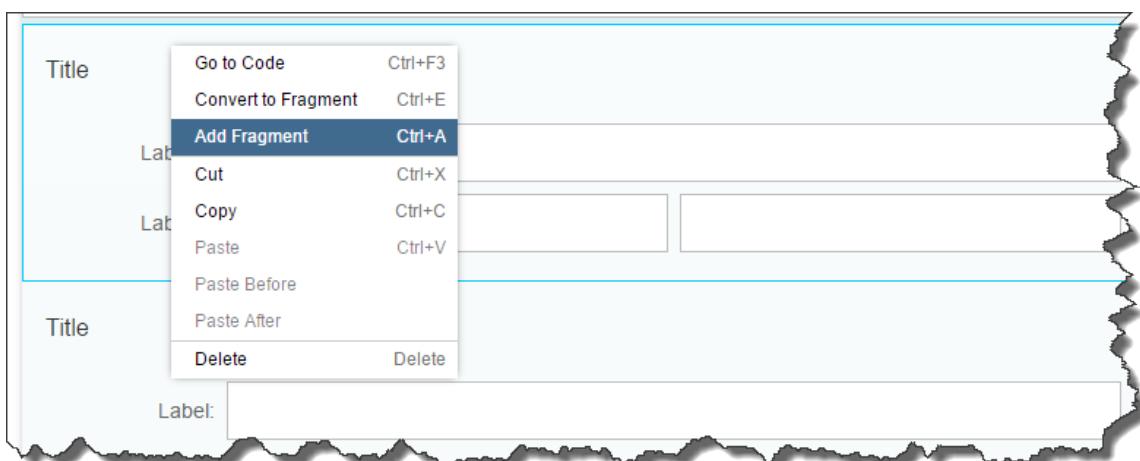
Learn about the new and changed features that are available for SAP Web IDE in August 2016.

Layout Editor (New)

- You can now add a `sap.ui.layout.form`. It includes release notes that describe what is to your view control using the layout editor. This form also contains the form builder feature. With it, you select data fields from your OData service and the controls and labels are automatically created in the form.



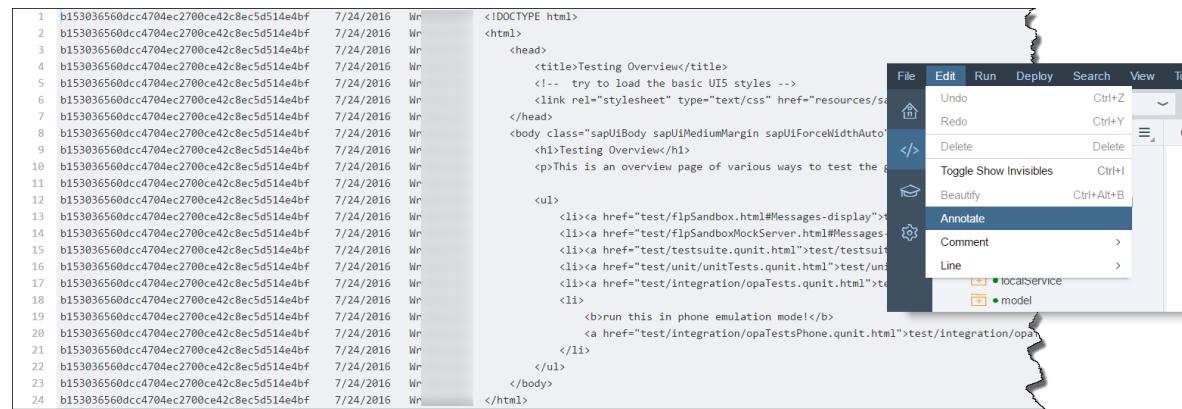
- The following features for supporting fragments have been added:
 - Adding an existing fragment from the context menu in the canvas and the Outline pane.
 - Removing a fragment from the view.
 - Moving a fragment to a new location in the view (only for fragments that contain one control/container. Fragments with multiple roots are not supported.)



For more information, see [Work with the Layout Editor \[page 338\]](#).

Git Annotations (New)

From the file context menu, you can now annotate code files (if they have been committed to Git) to see who was the last to change each line of code, the Git commit that last changed the line, and the date of that commit. The feature is similar to the `git blame` command.

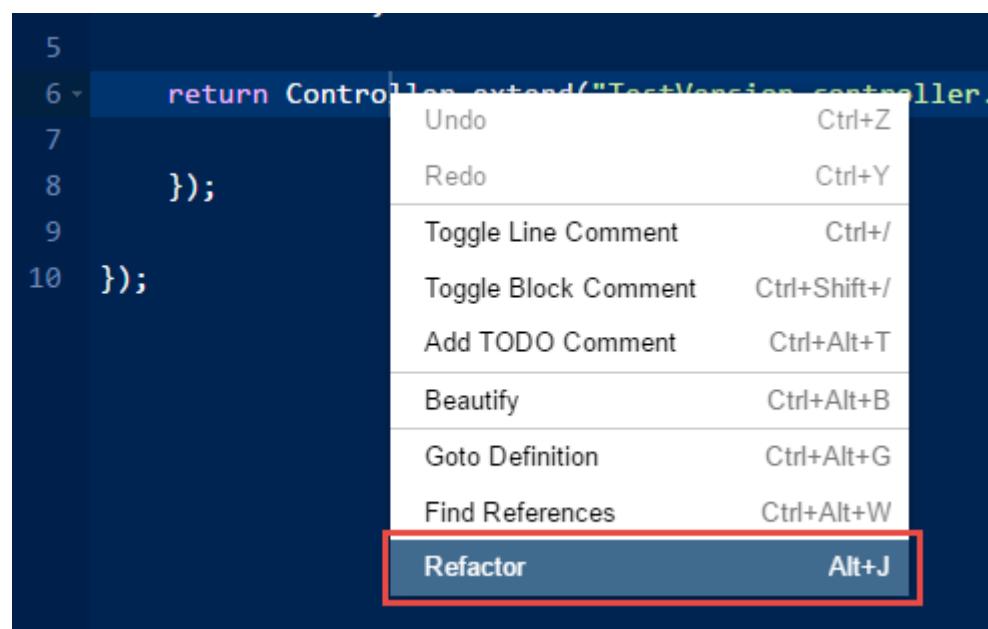


From the context menu in the annotated code file, you can open the [Git History](#) pane for the selected commit.

For more information, see [Working in the Code Editor \[page 214\]](#).

Refactor (New)

You can now change the name of a JavaScript function or variable, by right-clicking in the editor and selecting [Refactor](#). After you choose a valid new name, all references to the function or variable are automatically updated.



For more information, see [Working in the Code Editor \[page 214\]](#).

Code Completion Rich Tooltip (New)

Code completion of JavaScript and XML now includes a rich tooltip for SAPUI5 elements. The tooltip provides SAPUI5 documentation for the selected object. The tooltip replaces the [API Reference](#) pane.

For more information, see [Using Code Completion \[page 220\]](#).

Annotation Modeler (New)

The annotation modeler now supports the property `SampleSize` for the annotation `UI.DataPoint`.

For more information on the vocabulary, see [Architecture \[page 370\]](#).

Smart Business Service (New)

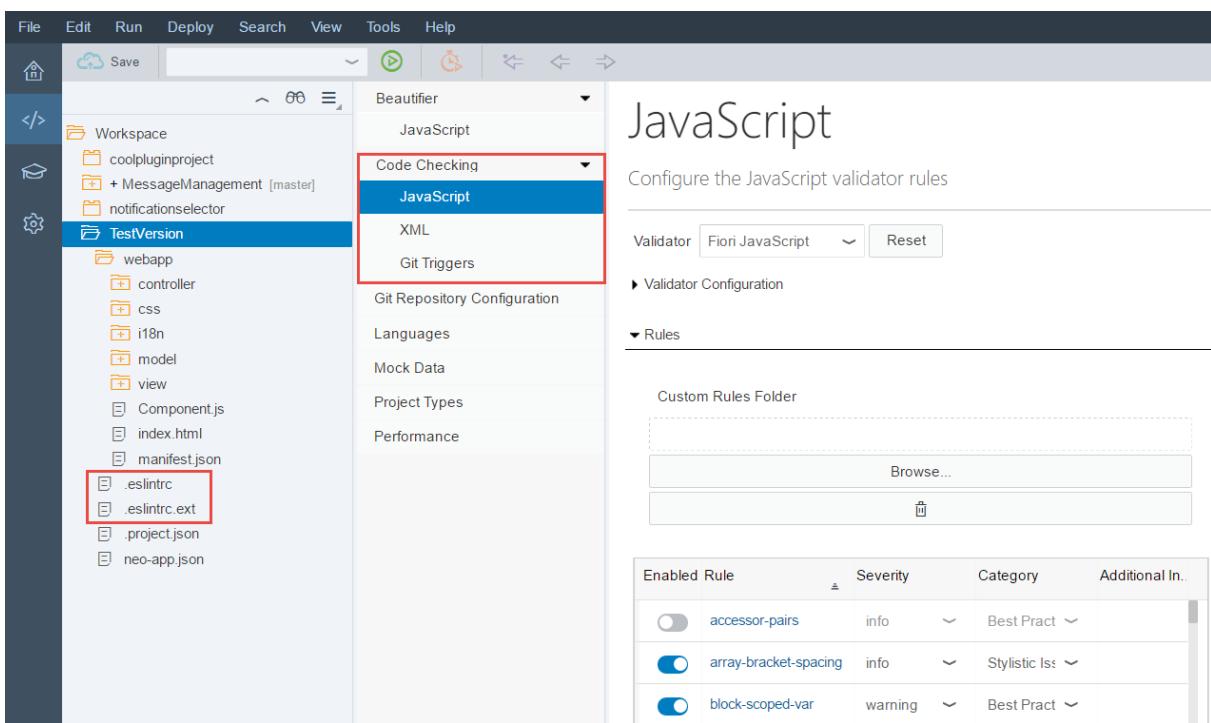
With the Smart Business service you can generate an annotation file on the fly from within the New Project wizard.

For more information, see [Use the Smart Business Service \[page 203\]](#).

Code Checking (Changed)

The rules for JavaScript code checking are now stored in dedicated files, `.eslintrc` and `.eslintrc.ext`, instead of in `.project.json`. If you have already established JavaScript rules in an `.eslintrc` file in another development environment, you can import the file to the root of your project and the settings will take effect for that SAP Web IDE project.

For existing projects with customized rules, the settings remain in `.project.json`, unless you change the rules and save them, in which case they are automatically migrated to `.eslintrc` and `.eslintrc.ext` files, which are added to the root of your project.



For more information, see [JavaScript Validation \[page 243\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)

[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

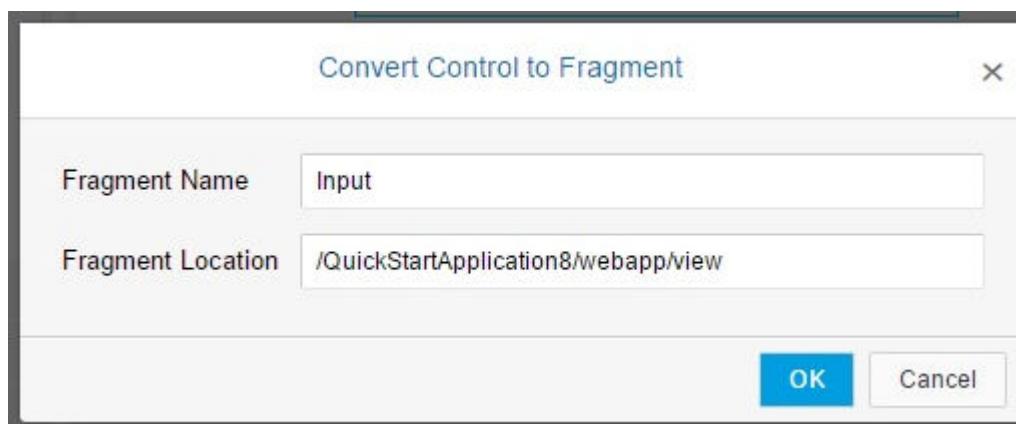
2.1.11 July 2016

Learn about the new and changed features that are available for SAP Web IDE in July 2016.

For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

Layout Editor (New)

You can extract a control from a view and wrap it in a fragment which can then be reused.



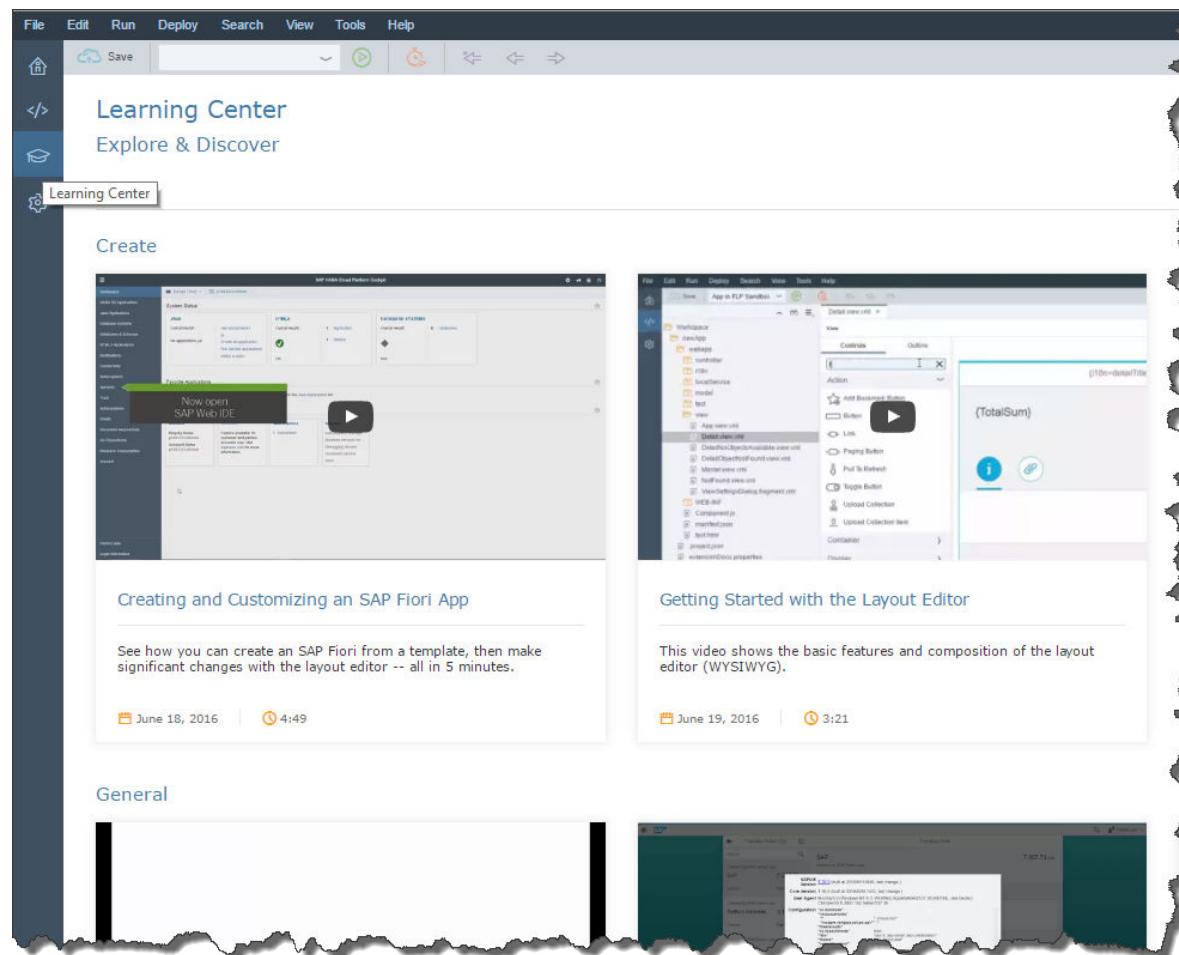
For more information, see [Work with the Layout Editor \[page 338\]](#).

Learning Center (New)

You can now learn about SAP Web IDE features and capabilities in the new Learning Center.

The Learning Center contains tutorial videos showing you how to work with SAP Web IDE.

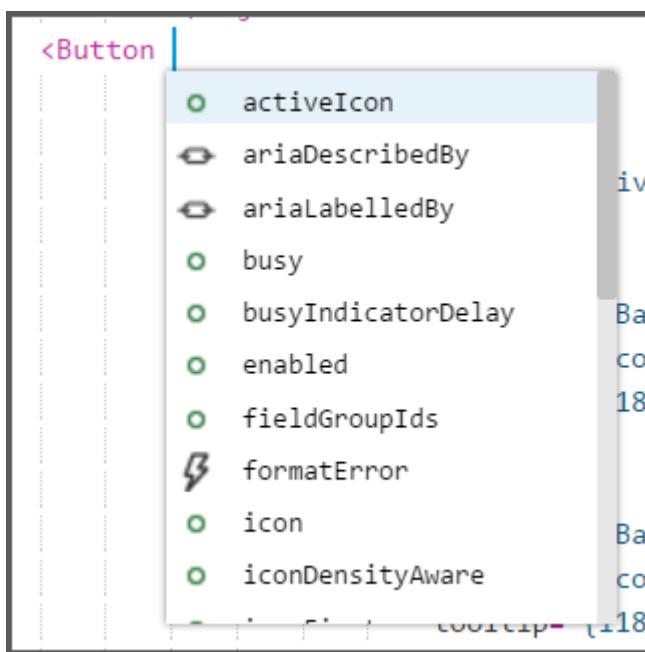
To open it, select the Learning Center perspective from the left sidebar.



Code Completion (New)

Code completion has been improved in the following ways:

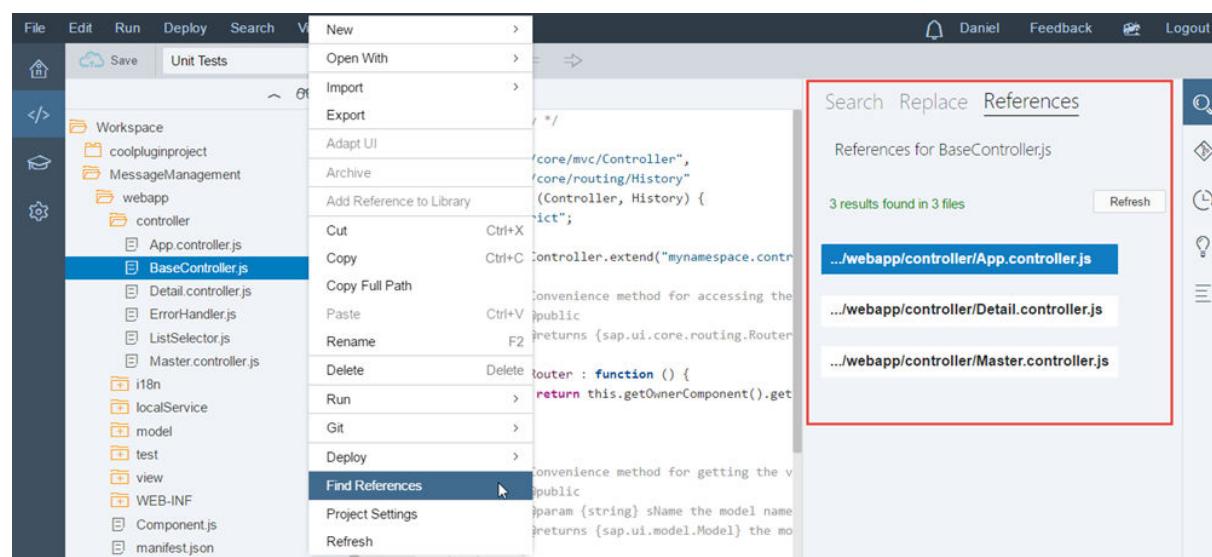
- Code completion is now available when specifying an SAPUI5 namespace in an XML file.
- Code completion within XML files now shows object associations, in addition to properties and events.



- You can now press **Right Arrow** to select the proposal from auto-hint code completion (in addition to pressing **Tab** or **Enter**).

Find References (New)

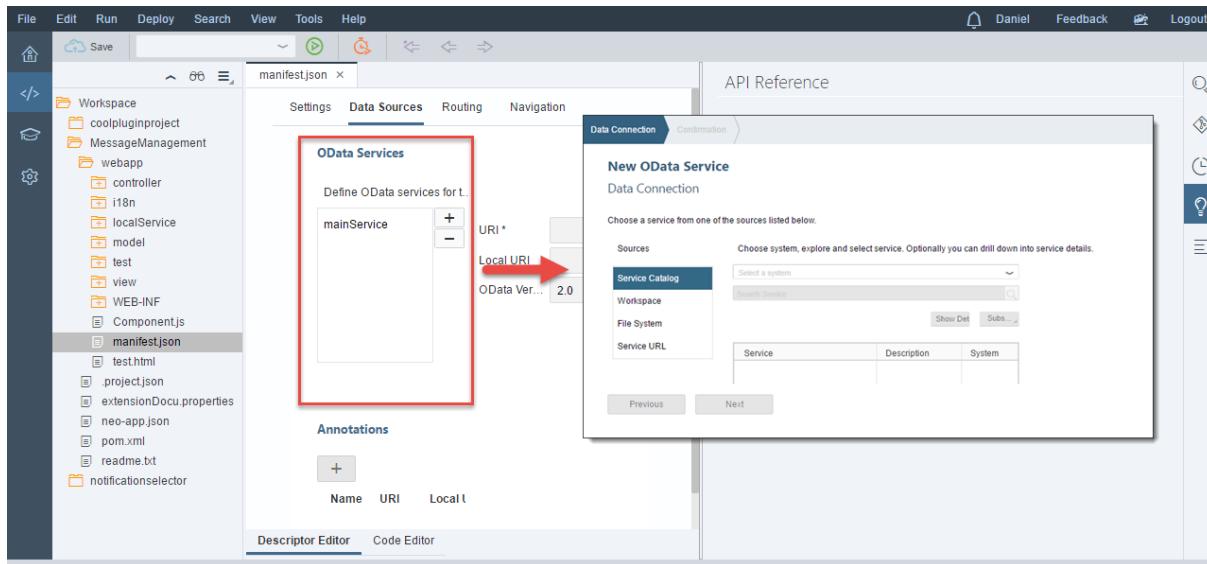
You can now right-click on a JavaScript file and select *Find References* to display a list of other JavaScript files in the project that reference the selected file.



For more information, see [Find References \[page 175\]](#).

Adding Data Sources (New)

When adding a data source in the application descriptor editor (by double-clicking the `manifest.json` file), you now get a wizard to make selecting a data source easier. The wizard is the same as the one that already appears as part of the wizard when creating a new project from a template.



SAPUI5 Version Validation (New)

In the deployment wizard, a validation is performed to ensure the same SAPUI5 version is used in the application and in the SAP system. Version incompatibility might cause problems at runtime.

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)

[April 2017 \[page 10\]](#)

[March 2017 \[page 12\]](#)

[February 2017 \[page 14\]](#)

[January 2017 \[page 18\]](#)

[December 2016 \[page 22\]](#)

[November 2016 \[page 31\]](#)

[October 2016 \[page 36\]](#)

[September 2016 \[page 40\]](#)

[August 2016 \[page 47\]](#)

[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.12 June 2016

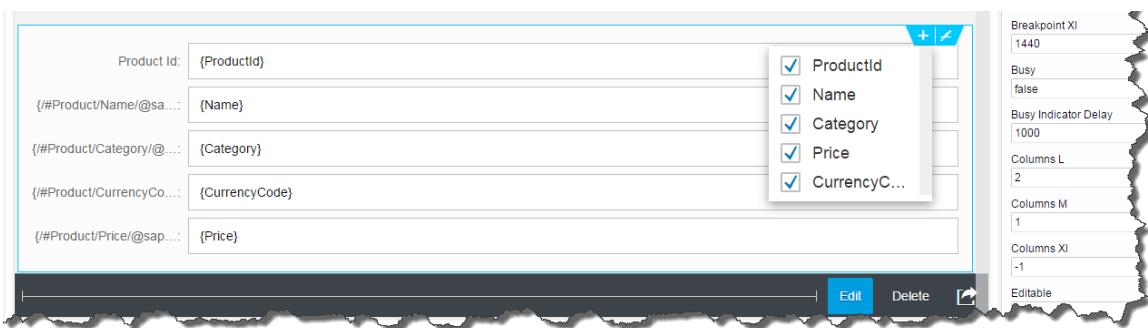
Learn about the new and changed features that are available for SAP Web IDE in June 2016.

SAP Web IDE Release Cycles (New)

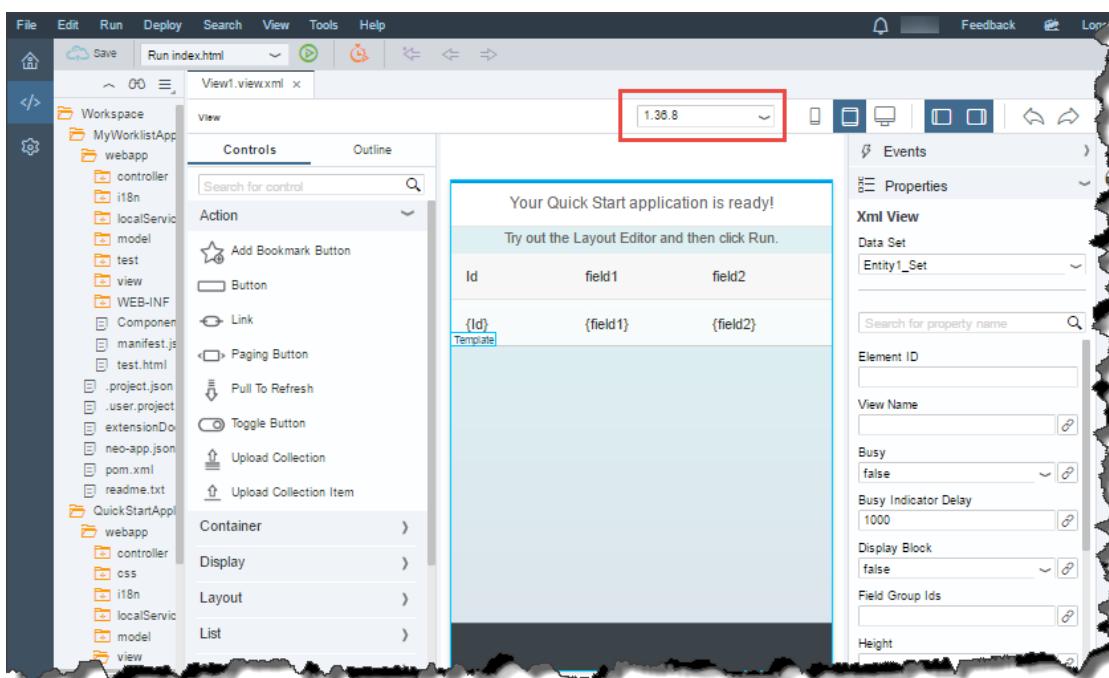
As of this release, SAP Web IDE will be updated every 2 weeks.

Layout Editor (New)

- There is a new form builder feature. With it, you select data fields from your OData service and the controls and labels are automatically created in the form.



- You can now bind a control property to a data field, an i18n model, and to a label annotation. For more information, see [Layout Editor Binding Capabilities \[page 343\]](#).
- You can now edit the *Visibility* property of a control from the *Properties* pane.
- You can now see the SAPUI5 version of the application being edited in the layout editor. You can change the version as necessary.



Finding References (New)

You can now find references to functions, variables, and object properties in JavaScript files in a project with up to 80 files. Search results are displayed in the *References* tab in the *Search* pane and are grouped by document. You can click each result to go to the code call.

Right-click the function, variable, or object property in the JavaScript file and choose *Find References* from the context menu.

models.js

```

1 sap.ui.define([
2     "sap/ui/model/json/JSONModel",
3     "sap/ui/Device"
4 ], function(JSONModel, Device) {
5     "use strict";
6
7     return {
8
9         createDeviceModel: function() {
10            var oModel =
11            oModel.setDef(
12            return oModel
13        }
14    };
15 });
16
17 });

```

Search Replace References

References for createDeviceModel in .

1 results found in 1 files Refresh

/App1/webapp/Component.js 1

24 this.setModel(models.createDeviceModel());

Undo Ctrl+Z
Redo Ctrl+Y
Toggle Line Comment Ctrl+/
Toggle Block Comment Ctrl+Shift+/
Add TODO Comment Ctrl+Alt+T
Beautify Ctrl+Alt+B
Goto Definition Ctrl+Alt+G
Find References Ctrl+Alt+W

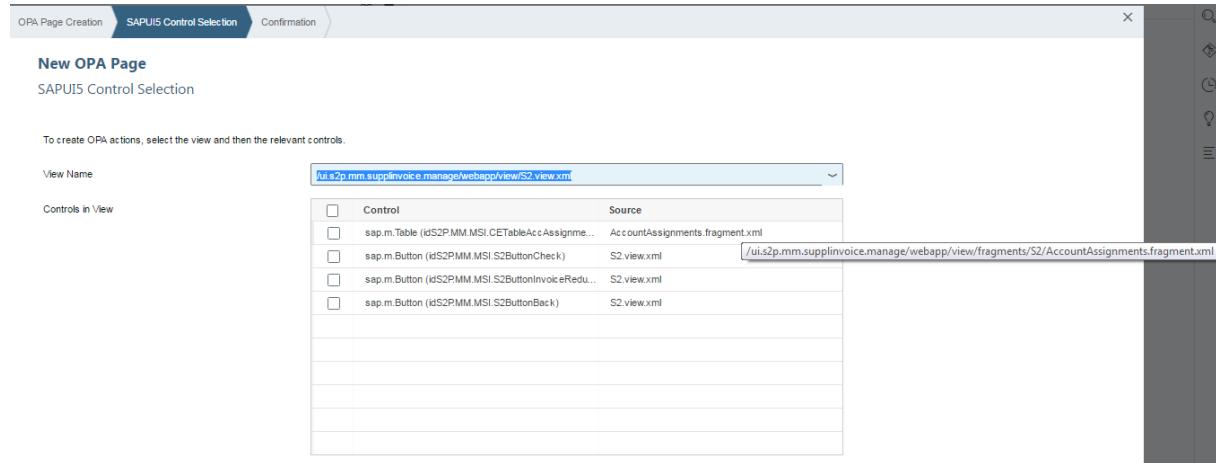
For more information, see [Find References \[page 175\]](#).

Sorting in Code Completion (New)

If you create an instance of an SAPUI5 control, the proposals provided by the code assist mechanism will show first the SAPUI5 options and then the general JavaScript methods.

Creating OPA Pages (Changed)

When you create an OPA page using the OPA Page Creation wizard, you can now choose the controls for the OPA action template that is generated. All the controls that are in the selected view or any of its fragments, and that are relevant for an OPA test are available for selection.



For more information, see [Create Tests \[page 332\]](#).

Embedded Type Code Completion (Changed)

For JavaScript AMD files, 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.

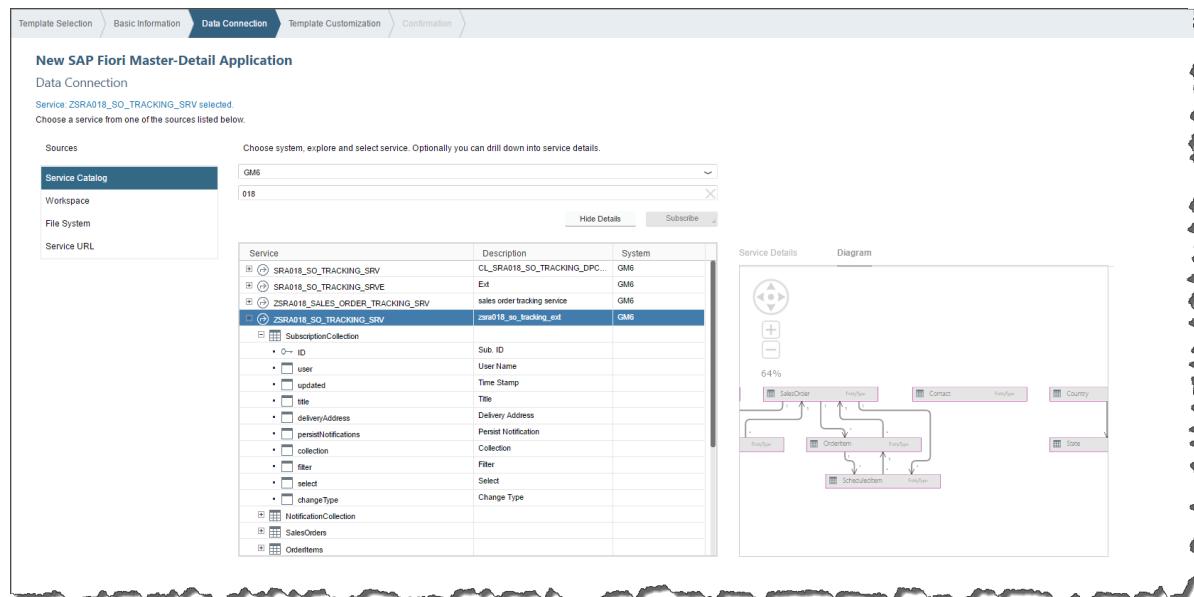
For more information, see [Using Code Completion \[page 220\]](#).

Data Connection Step in New Project Wizard (Changed)

The *New Project* wizard now contains a new control from which you can select the data source required for your project.

The new control allows you to see the selected service structure, metadata, and a visual representation of the entities.

Additionally, you can see live data.



For more information, see [Select a Data Source \[page 183\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

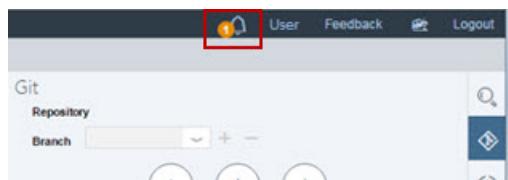
[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.13 May 2016

Learn about the new and changed features that are available for SAP Web IDE in May 2016.

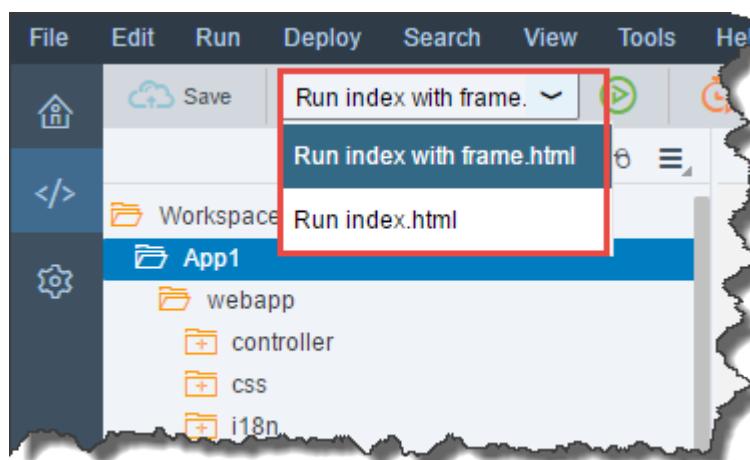
Notification Alerts (New)

SAP Web IDE has a new notification mechanism that displays short update messages, such as upcoming changes and known issues. When a notification is published, the notification displays a badge with the number of unread notification alerts.

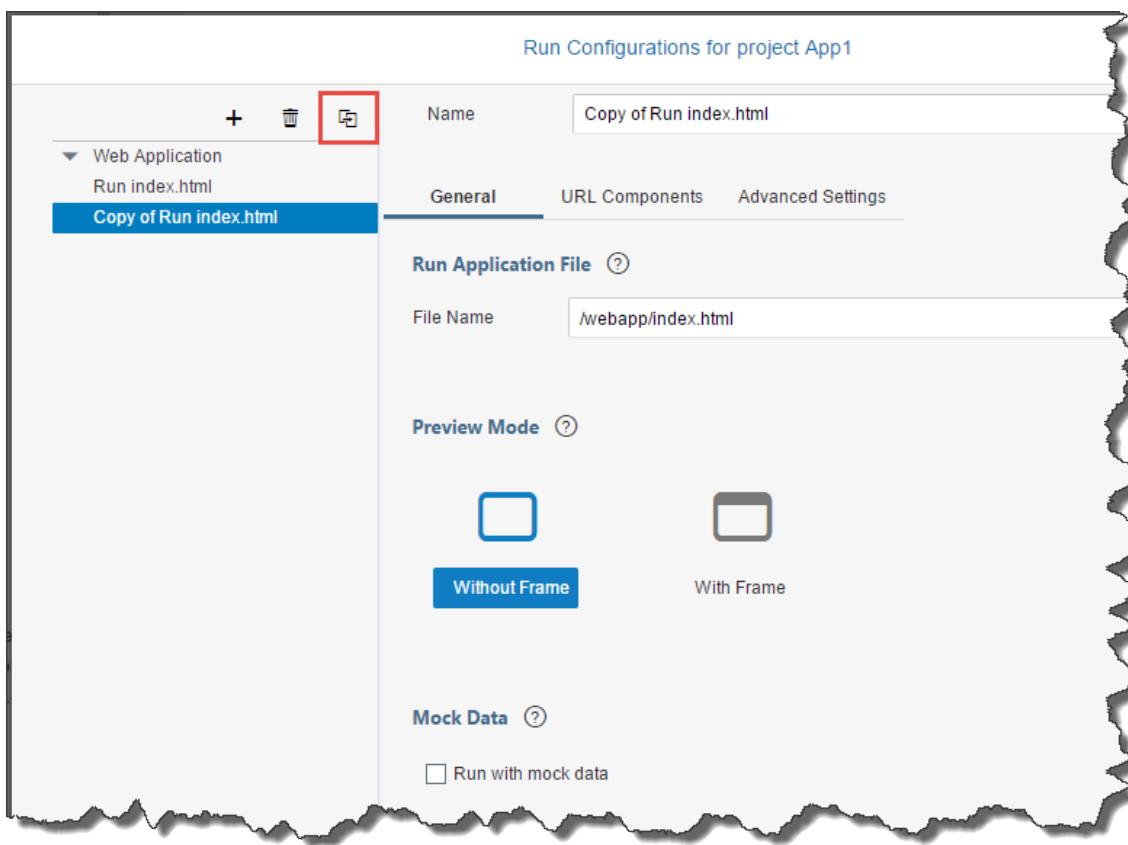


Running Applications (Changed)

- You can now choose from the toolbar which run configuration to use when running applications for testing. Select a run configuration and then choose the *Run* button. The drop-down menu includes all the run configurations that can be applied to your project, and the last used run configuration is selected by default.



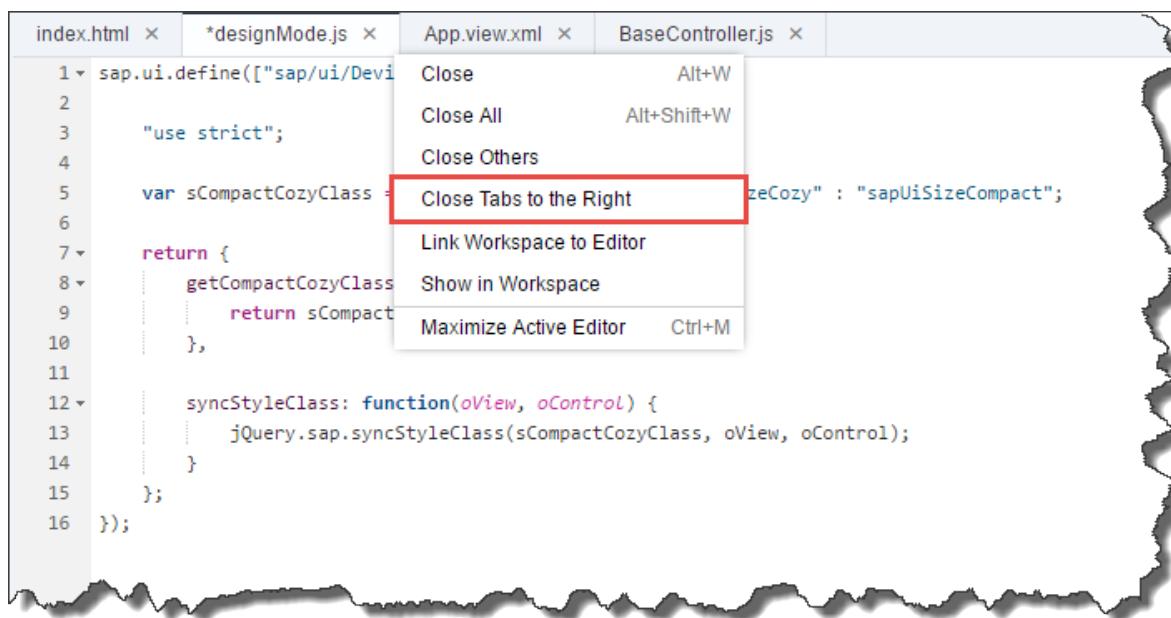
- In the *Run Configurations* page, you can create a new run configuration by duplicating an existing run configuration and then editing it. Select an existing configuration and choose the new *Duplicate* button.



For more information, see [Create Run Projects \[page 435\]](#) and [Run Applications from the Workspace \[page 433\]](#).

Code Editor Tabs (Changed)

You can now close all tabs to the right of the open tab. Use the *Close All Tabs to the Right* option in the context menu.



For more information, see [Working in the Code Editor \[page 214\]](#).

Customizing Code Editor Fonts (Changed)

You can now choose the code editor font in the [Code Editor](#) Preferences page.

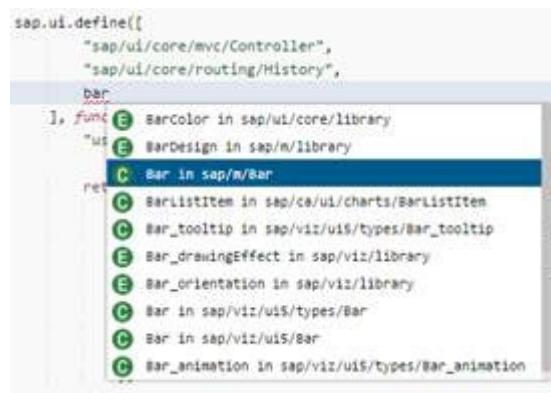
A screenshot of the SAP Web IDE Preferences page, specifically the "Code Editor" section. On the left, there's a sidebar with "Appearance" settings: "Code Editor Theme" set to "SAP Cumulus", "Font" dropdown (set to "(Default)", highlighted with a red box), "Font Size", "Code Folding", and checkboxes for "Full line selection" (unchecked), "Highlight selected word" (checked), "Show invisible characters" (unchecked), and "Use spaces for tab indentation" (unchecked). On the right, there's a "Preview" window showing a snippet of code in the selected font. The font dropdown list includes: (Default) (highlighted), Arial Monospaced for SAP, Consolas, Courier, Courier New, Lucida Console, Lucida Sans Typewriter, and MS Mincho.

For more information, see [Configure the Code Editor \[page 217\]](#).

Code Completion Icons (Changed)

The icons that are displayed for code completion suggestions are now aligned with the SAPUI5 Demo Kit.

This example shows icons for events and classes for AMD JavaScript code completion.



For more information, see [Using Code Completion \[page 220\]](#).

Annotation Modeler (Changed)

You can now view and access issues in the `annotation.xml` file that are specific to the annotation modeler in the *Problems* view.

For more information, see [Validation of the Annotation File \[page 391\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)

[April 2017 \[page 10\]](#)

[March 2017 \[page 12\]](#)

[February 2017 \[page 14\]](#)

[January 2017 \[page 18\]](#)

[December 2016 \[page 22\]](#)

[November 2016 \[page 31\]](#)

[October 2016 \[page 36\]](#)

[September 2016 \[page 40\]](#)

[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.14 April 2016

Learn about the new and changed features that are available for SAP Web IDE in April 2016.

Quick Access Feature (New)

Quickly access capabilities and trigger actions.

Press `CTRL` + `3` to open the *Quick Access* dialog box and double-click the action you want to perform.

Quick Access

The screenshot shows the SAP Web IDE's Quick Access dialog. At the top is a search bar with a magnifying glass icon. Below it is a blue header bar with the word "About". The main content area contains a list of links with their keyboard shortcuts:

- Add Reference to Library
- Advanced Repository Search Ctrl+Shift+F
- API Reference Ctrl+Shift+A
- Application from SAP HANA Cloud Platform
- Application from SAPUI5 ABAP Repository
- Application Status
- Archive
- Back Ctrl+Alt+R
- Clear Console

At the bottom of the dialog, the text "Results found: 64" is displayed.

Translation Hub - Experimental (New)

You can translate your project's `i18n.properties` file using the Translation Hub service.

Generate Translation Files

The screenshot shows the SAP Web IDE's "Generate Translation Files" dialog. It has two main sections:

- Application Domain:** A dropdown menu set to "Basis".
- Target Languages*:** A list titled "AVAILABLE LANGUAGES" with checkboxes next to language names. The checked languages are German and English. Other languages listed include Czech, Danish, Greek, and Spanish.

At the bottom right of the dialog are "Generate" and "Cancel" buttons.

For more information, see [Use the Translation Hub \[page 198\]](#).

Creating a Local Git Repository (New)

You can now create a local Git repository for your project without deploying the project to the SAP HANA Cloud Platform. This also lets you connect your local repository to any remote Git repository.

For more information, see [Initialize a Local Git Repository \[page 411\]](#).

Layout Editor (New)

The layout editor has the following new features:

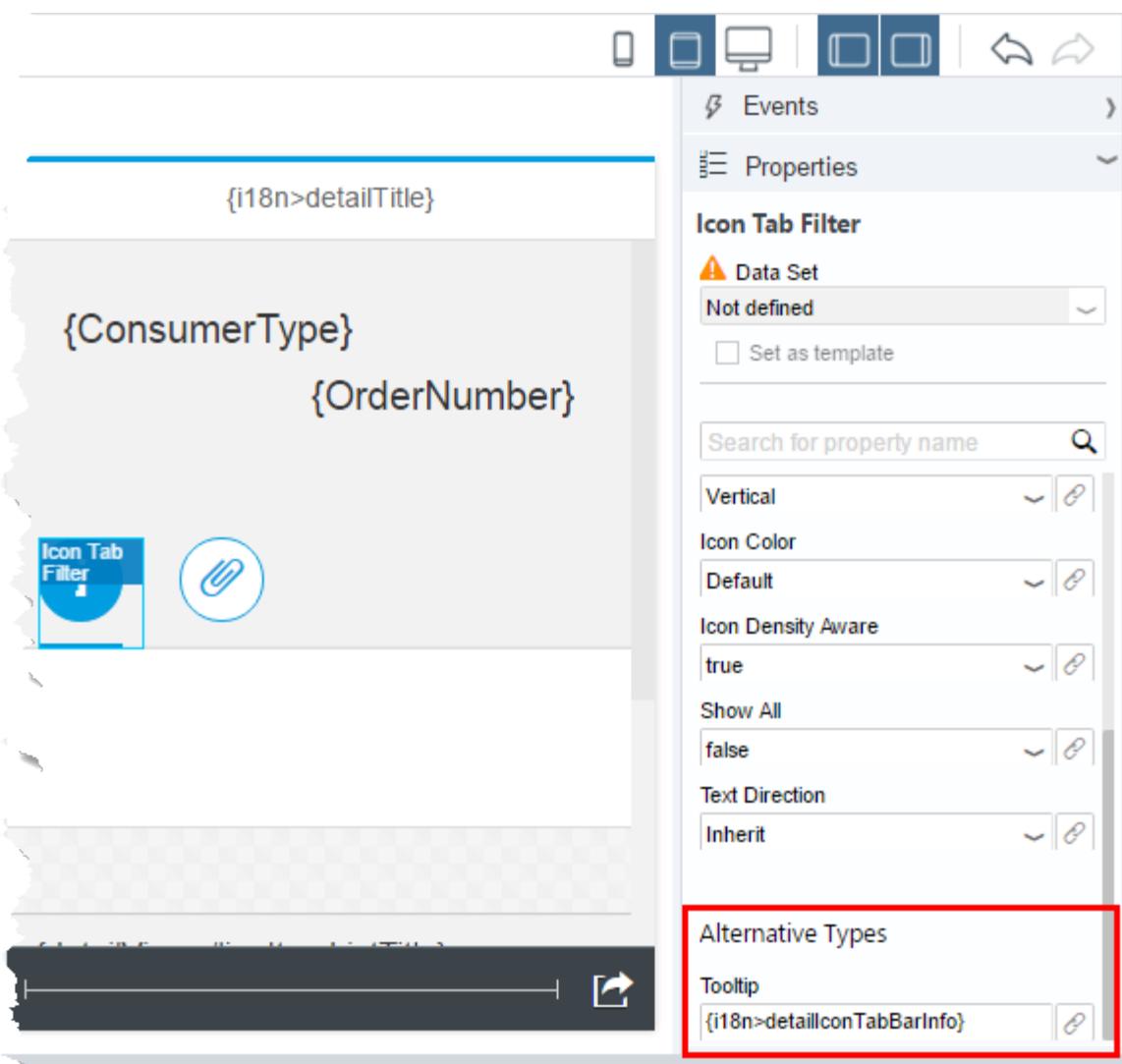
- New SAPUI5 controls have been added to the layout editor:
 - sap.ui.vbm.AnalyticMap
 - sap.ui.vbm.GeoMap

For more information, see [SAPUI5 Controls Supported in the Layout Editor \[page 349\]](#).

- **Alternative Types**

You can now define an alternative type for an aggregation that supports AltType. This allows you to enter a string to quickly define a default control.

For example, a tooltip is an aggregation that supports AltType. You can either create a new *Tooltip* control in the tooltip aggregation, or enter the tooltip string in the *Tooltip* AltType.



Deploying to SAP HANA Cloud Platform (Changed)

You can now deploy any application to SAP HANA Cloud Platform as a new application or as an update to a previously deployed application in any of your accounts.

The deployed artifact is the result of the application build that was performed behind the scenes during deployment. This represents only the productive version of the application and does not reflect the project's source files.

The application source code should be managed in Git. When you deploy the application to SAP HANA Cloud Platform, the source code is no longer automatically pushed to the SAP HANA Cloud Platform Git. You can choose to connect your project to any Git repository, including the SAP HANA Cloud Platform Git, and push your changes there later.

Deploy Application to SAP HANA Cloud Platform

Notice: This functionality was recently changed. If you want to connect your app to a Git repository, use the "Initialize Local Repository" command from the Git menu. You can then perform all Git operations on this repository from the Git pane. [More information](#).

Application Details

Deploy a new application Update an existing application

Account *	x2a4336b4
Project Name	CA_FIORI_INBOX
Application Name *	Select an application.

Version Management

Version * 1.0.0 Activate

Buttons: Deploy | Cancel

For more information, see [Deploy Applications to SAP Cloud Platform \[page 451\]](#).

Code Completion Snippets for SAPUI5 Method Properties (Changed)

The code editor now provides code completion suggestions for properties in SAPUI5 methods. For example, suggestions for `Opa5.createPageObjects` include all the properties for the methods: `baseClass`, `namespace`, `actions`, and `assertions`.



```

Opa5.createPageObjects(
{
  "our-page-object-name": {
    "baseClass": fn_baseClass, // function
    "namespace": fn_namespace, // function
    "actions": {},
    "assertions": {}
  }
}
)

```

The screenshot shows a code editor with the following code snippet:

```

Opa5.createPageObjects(
{
  "our-page-object-name": {
    "baseClass": fn_baseClass, // function
    "namespace": fn_namespace, // function
    "actions": {},
    "assertions": {}
  }
}
)

```

The word `fn_baseClass` is highlighted with a red underline, indicating it is a suggestion or part of a code completion list. The code editor interface includes a vertical toolbar on the left and a status bar at the bottom.

For more information, see [Using Code Completion \[page 220\]](#).

AMD Code Completion for Modules (Changed)

The code editor now provides code completion suggestions for SAPUI5 module dependencies in Asynchronous Module Definition (AMD). Proposed suggestions are for the element type of module according to the element name entered.

The example shows how the selected suggestion is added in the code with quotes.



The screenshot shows a code editor with the following code:

```
1 - sap.ui.define([
2     "sap/m/RatingIndicator",
3     "sap/m/Link",
4     "sap/m/Label",
5     "sap/ui/core/Control",
6     but
7     ], function() {  
8         "use strict";  
9         "sap/m/Button" in sap.ui.commons/library  
10        return "sap/m/Button"  
11    })
```

A code completion dropdown is open at the end of the line, showing suggestions for the module dependency. The suggestion "Button in sap/m/Button" is highlighted in blue. The dropdown also lists other suggestions like "ButtonStyle in sap/ui/commons/library", "Button in sap/ui/commons/Button", and "ButtonType in sap/m/library".



The completed code in the editor is:

```
sap.ui.define([
    "sap/m/RatingIndicator",
    "sap/m/Link",
    "sap/m/Label",
    "sap/ui/core/Control",
    "sap/m/Button"
], function(RatingIndicator, Link,
    "use strict";
```

For more information, see [Using Code Completion \[page 220\]](#).

Annotation Modeler (Changed)

The annotation modeler has been enhanced with UI-based features:

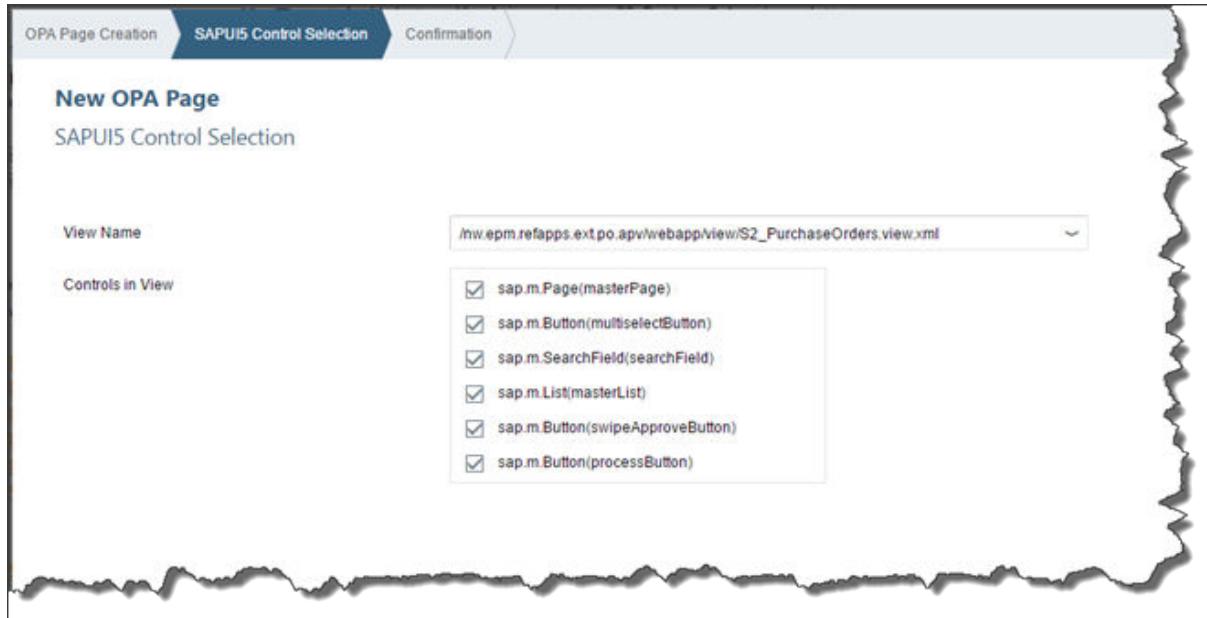
- In the *Annotation Structure* pane, the *Info* and *Description* columns have been replaced by the *Key Information* column, which displays the most relevant information at a glance. In addition, tooltips have been implemented to make the *Key Information* column easier to use.
- In the *Properties* pane, the *Apply* button has been removed. Values entered for the properties of annotation terms are validated and applied immediately.
- In the *Properties* pane, the order in which the properties are displayed has been changed. If available, the ID or qualifier is shown first followed by the label and all mandatory fields. Fields that are not mandatory are shown after the mandatory ones.

The target handling for facet properties has been enhanced to allow the selection of a hierarchical path to the target so that an annotation from another entity set can be set as a target annotation.

For more information, see [Annotation Modeler \[page 369\]](#).

OPA Page Creation Wizard (Changed)

The *OPA Page Creation* wizard includes a new step to choose the SAPUI5 controls that you want to test.



For more information, see [Create Tests \[page 332\]](#).

Problems View (Changed)

You can now view information about XML semantic validation issues in the Problems View.

Problems for Project MyFirstApp			
All (14)	Errors (7)	Warnings (6)	Infos (1)
Severity	Description	File	Location
⚠	Fiori XML Analysis: An ID must be set for this control.	Masterview.xml(19, 12)	/MyFirstApp/webapp/view
⚠	Fiori XML Analysis: An ID must be set for this control.	Masterview.xml(38, 28)	/MyFirstApp/webapp/view
⚠	Fiori XML Analysis: An ID must be set for this control.	Masterview.xml(52, 12)	/MyFirstApp/webapp/view
⚠	The control «sap.m.DateTimeInput» is deprecated since version 1.32.8.	Masterview.xml(5, 4)	/MyFirstApp/webapp/view
⚠	The event select for «sap.m.List» is deprecated since version 1.16.	Masterview.xml(27, 17)	/MyFirstApp/webapp/view
⚠	ESLint: (no-unused-vars) s is defined but never used	App.js(8, 4)	/MyFirstApp/webapp/test/integration/pages
			Category
			Fiori Architectural Guidelines Error
			Fiori Architectural Guidelines Error
			Fiori Architectural Guidelines Error
			Semantic Error
			Semantic Error
			Variable

For more information, see [Using the Problems View \[page 323\]](#).

Deprecation (Changed)

XML semantic validation now includes deprecation warnings for properties, events, and aggregations, as well as deprecated controls. These warnings are displayed in the code editor and in the Problems View. The code editor information provides alternative suggestions.

The screenshot shows the SAP Web IDE interface with several tabs open: ProductImage.fragment.xml, Master.js, NotFound.js, and Detail.view.xml. The Detail.view.xml tab is active, displaying the following XML code:

```
1 <mvc:View controllerName="n.controller.Detail" xmlns="sap.m" xmlns:mvc="sap.ui.  
2   <Table id="MyTable"  
3     select="Onselect"></Table>  
4 </mvc:View>
```

A semantic warning is displayed in a callout box over the 'select' attribute of the Table element:

warning: Semantic Error: : The event select for <sap.m.Table> is deprecated since version 1.16. Use the selectionChange event instead.

For more information, see [XML Semantic Validation \[page 297\]](#).

Outline View (Changed)

The *Outline* pane now reflects the function assignment names of anonymous functions assigned to variables.

The screenshot shows the SAP Web IDE interface with the Outline pane open. The left side shows a portion of a JavaScript file with code like:

```
75 *  
76 * Returns the <code>sap.ui.core.Icon</code> control used to display the expand/collapse icon.  
77 * @returns {sap.ui.core.Icon}  
78 * @private  
79 */  
80 NavigationListItem.prototype._getExpandIconControl = function () {  
81   var expandIconControl = this.getAggregation('_expandIconControl');  
82  
83   return expandIconControl;  
84 };  
85  
86 */
```

The right side shows the Outline pane with a tree view of function definitions. One item is highlighted in blue:

- tex: function()QueryLibraryItem,Icon,NavigationList,Renderer,IconPool
- NavigationListItem
- object
- tex: NavigationListItem.prototype._getExpansionControl
- tex: NavigationListItem.prototype._getExpansionIconControl(expand)
- tex: NavigationListItem.prototype.getTooltip(exp)
- tex: NavigationListItem.prototype.getNavigationList
- tex: NavigationListItem.prototype.createPopups(list)
- tex: NavigationListItem.prototype.onPopupsSelect(event)
- tex: NavigationListItem.prototype._selectItem(event)

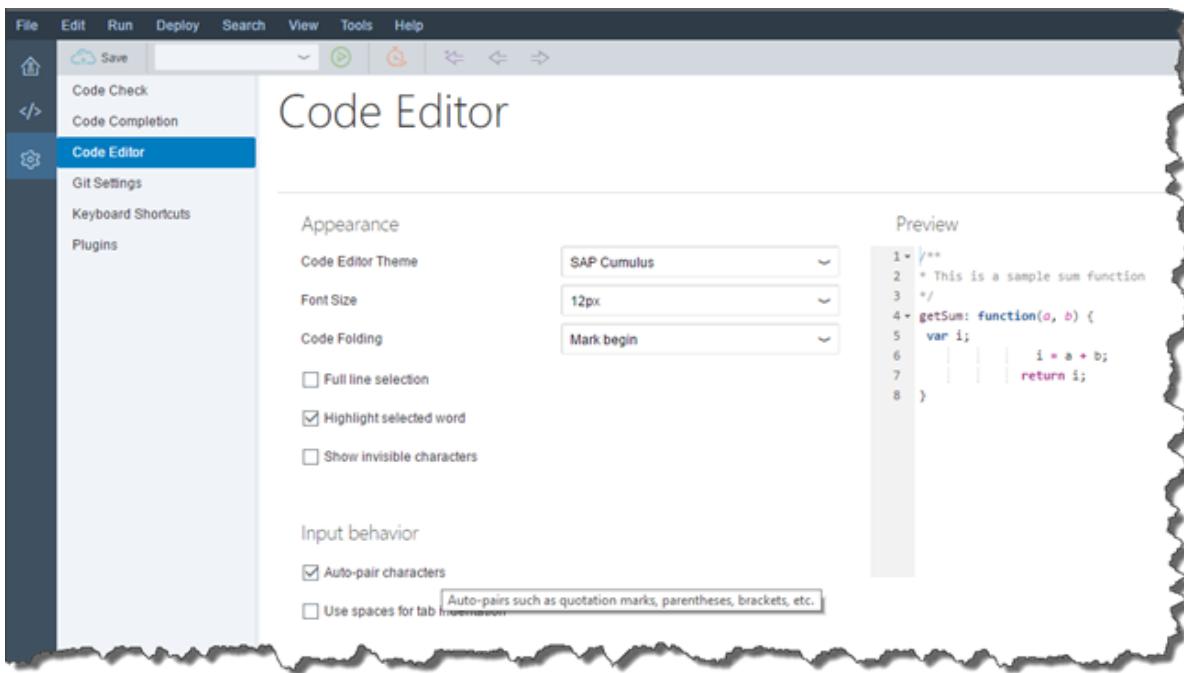
For more information, see [Using the Outline Pane for JavaScript Files \[page 329\]](#).

Code Editor Settings (Changed)

You can now customize additional settings for the appearance and input behavior of the code editor, including:

- Auto-pairing of characters such as quotation marks, brackets, parentheses, and so on.
- Highlighting of all occurrences of a selected word.
- New themes for the code editor using a dark background.

You can see your settings in the preview as you make changes.



For more information, see [Configure the Code Editor \[page 217\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)

SAP Web IDE 1.17 [page 99]
SAP Web IDE 1.16 [page 103]
SAP Web IDE 1.15 [page 106]
SAP Web IDE 1.14 [page 109]
SAP Web IDE 1.13 [page 111]
SAP Web IDE 1.12 [page 114]
SAP Web IDE 1.11 [page 118]
SAP Web IDE 1.10 [page 121]
SAP Web IDE 1.9 [page 124]
SAP Web IDE 1.8 [page 127]
SAP Web IDE 1.7 [page 131]
SAP Web IDE 1.6 [page 136]
SAP Web IDE 1.5 [page 138]
SAP Web IDE 1.4 [page 140]
SAP Web IDE 1.3 (Trial Only) [page 145]
SAP Web IDE 1.2 [page 146]

2.1.15 March 2016

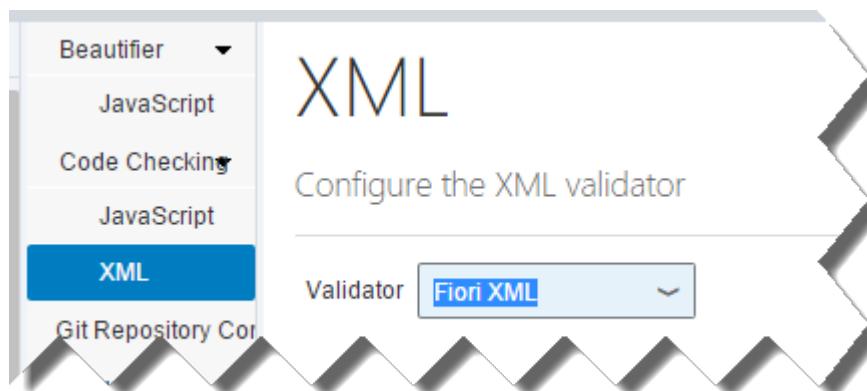
This section gives you an overview of the new features and functions of SAP Web IDE. It includes release notes that describe what is new, enhanced, changed, or deleted in March 2016.

For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

SAP Fiori Validators (New)

SAP Web IDE now includes SAP Fiori validators for XML and JavaScript. These SAP Fiori validators contain the same rules that are used to build SAP Fiori projects. Use these validators to prevent issues that could break the build of your SAP Fiori project.

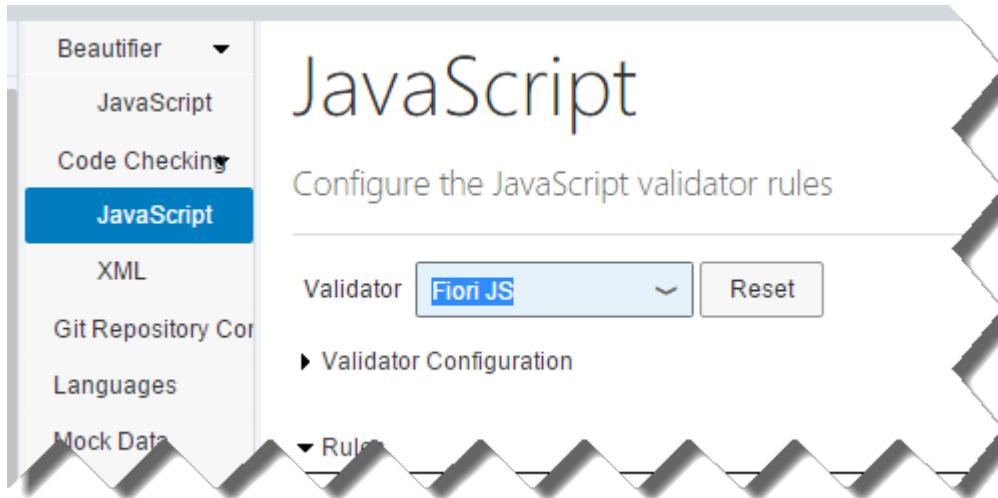
- To use the SAP Fiori XML validator, choose [Project Settings](#) [Code Checking](#) [XML](#).



- [XML Validation \[page 297\]](#)

For more information, see [XML Validation \[page 297\]](#).

- To use the SAP Fiori JavaScript validator, choose [Project Settings](#) [Code Checking](#) [JavaScript](#).



- [Customize JavaScript Validator Configuration \[page 245\]](#)

For more information, see [Customize JavaScript Validator Configuration \[page 245\]](#).

XML Semantic Validation (New)

SAP Web IDE now includes semantic validation for XML files.

For more information, see [XML Semantic Validation \[page 297\]](#).

Code Checking Triggers on Push to Source Control Repository (New)

You can now configure per project whether to run code checking when pushing code changes to the source control repository in a specific project.

Choose [Project Settings](#) [Code Checking](#) [Triggers](#).

Code Checking Triggers

Push to Source Control Repository

Define the push process flow according to the severity of problems found.

Severity	
<input checked="" type="checkbox"/> Notify before push	Error
<input type="checkbox"/> Block push	Error

When notification is configured, project developers who push code changes receive a notification about problems before the push starts. They can view the problems and fix them before pushing, or if the push process is not blocked, they can choose to continue with the push.

i Note

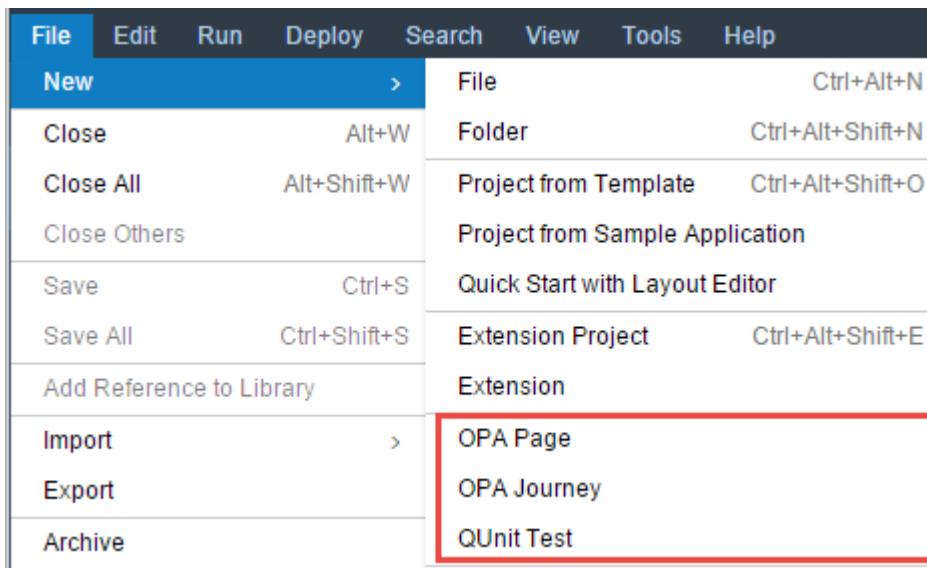
Notification is not automatically configured for existing projects.

For more information, see [Customize Code Checking Triggers \[page 197\]](#).

Wizard to Create OPA Pages, OPA Journeys, and QUnit Tests (New)

You can now create OPA pages, OPA journeys, and QUnit tests in a project using a wizard.

Choose  *File*  and the required option.



For more information, see [Create Tests \[page 332\]](#).

Application Testing with Performance Measurements (New)

You can run an application and view performance measurements as you activate user controls. You can configure thresholds for alerts, which will be displayed when measurements exceed those thresholds.

SAPUI5 Application Performance Results									
Status	Start Time	Initiator	Event	E2E Duration [ms]	Client Duration [ms]	Round Trips	Round Trips Dura...	Backend Requests	Backend Duration...
✓	5:08:00 PM	__popover6	click	0	0	0	0	0	0
✓	5:08:00 PM	userPreferencesButt	click	105	105	0	0	0	0
✓	5:07:56 PM	actionsBln	click	54	54	0	0	0	0
✓	5:07:49 PM	__tile43	click	834	723	1	110	0	0
⚠	5:07:37 PM	sf	click	578	275	2	173	0	0
✓	5:05:32 PM	__xmlview2--pullToR	click	0	0	0	0	0	0

For more information, see [Running Applications with Performance Measurements \[page 446\]](#).

Keyboard Shortcut Customization (New)

You can now customize keyboard shortcuts in SAP Web IDE.

For more information, see [Customize Keyboard Shortcuts \[page 169\]](#).

Template Development (Changed)

When you develop a template, you can define the run configurations that will be generated when a project is created with that template.

Compare Editor (Changed)

You can now compare two commit versions of the same file in the source control repository. Choose the two versions to compare in the *Git History* pane.

The screenshot shows the SAP Web IDE's Git History pane. At the top, there are search and filter fields: 'History of Branch' set to '750549/47' and 'Folder Name' set to '/com.sap.watt'. Below these are navigation buttons for navigating through the history. The main area displays a commit log with several entries:

- [Fix] GitLog Next Step with the git correct api
- Merge "remove jslint leftovers and replace with esl"
- Merge "Added a Tag text to i18n"** (highlighted in blue)
- remove jslint leftovers and replace with eslint config
- Added a Tag text to i18n** (highlighted in blue)
- reuselib - some modification
- Merge "Replace JSLint with ESLint - static code ch"
- Merge "[FIX] - fix some strings"
- Merge "[Fix] Git Log - Insert the up down search b
- [FIX] - fix some strings
- Replace JSLint with ESLint - static code checks for
- [Fix] Git Log - Insert the up down search button int

On the left side of the commit log, there is a vertical timeline with colored nodes (blue, orange, red) connected by lines, representing the commit graph. At the bottom of the pane, there is a 'Compare' button and a table titled 'Common files' containing two rows:

	Common files
<input type="checkbox"/>	i18n.properties
<input checked="" type="checkbox"/>	GitLog.view.js

A vertical toolbar on the right side of the pane contains icons for search, refresh, and other operations.

For more information, see [Git Commands from the Git History Pane \[page 427\]](#).

Annotation Modeler (Changed)

The annotation modeler has been enhanced for better user guidance. In the Properties pane, the user input is validated against the field type.

For more information, see [Annotation Modeler \[page 369\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)
- [SAP Web IDE 1.17 \[page 99\]](#)
- [SAP Web IDE 1.16 \[page 103\]](#)
- [SAP Web IDE 1.15 \[page 106\]](#)
- [SAP Web IDE 1.14 \[page 109\]](#)
- [SAP Web IDE 1.13 \[page 111\]](#)
- [SAP Web IDE 1.12 \[page 114\]](#)
- [SAP Web IDE 1.11 \[page 118\]](#)
- [SAP Web IDE 1.10 \[page 121\]](#)
- [SAP Web IDE 1.9 \[page 124\]](#)
- [SAP Web IDE 1.8 \[page 127\]](#)

- [SAP Web IDE 1.7 \[page 131\]](#)
- [SAP Web IDE 1.6 \[page 136\]](#)
- [SAP Web IDE 1.5 \[page 138\]](#)
- [SAP Web IDE 1.4 \[page 140\]](#)
- [SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
- [SAP Web IDE 1.2 \[page 146\]](#)

2.1.16 February 2016

This section gives you an overview of the new features and functions of SAP Web IDE. It includes release notes that describe what is new, enhanced, changed, or deleted.

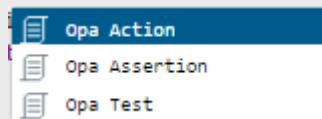
For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

Code Completion for OPA and QUnit Tests (New)

- In an OPA page, SAP Web IDE can now provide snippets for OPA actions and OPA assertions. You can also use code completion to add snippets for OPA tests in an OPA journey.

Example

Enter `opa` **Ctrl** + **Space**. The following suggestions are displayed.

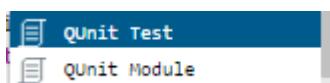


On choosing *Opa Assertion*, the following snippet is inserted:

```
AssertionName : function() {
    return this.waitFor({
        id: "ControlId",
        viewname: "viewName",
        success: function() {
            Opa5.assert.ok(false, "Implement me");
        },
        errorMessage: "Was not able to find the control with the id ControlId"
    });
}
```

You can then add the required definitions in the placeholders to create the OPA assertion.

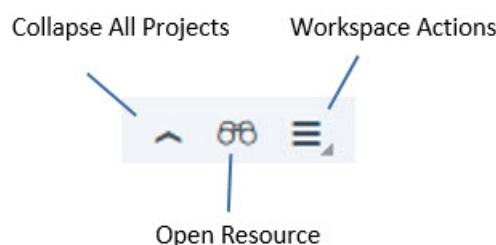
- In QUnit tests, you can enter `qunit` **Ctrl** + **Space** to use code completion. Choose from the following options to add snippets for QUnit modules or QUnit tests:



For more information, see [Code Completion for OPA and QUnit Tests \[page 223\]](#).

Workspace Toolbar (New)

A new toolbar has been added above the workspace:

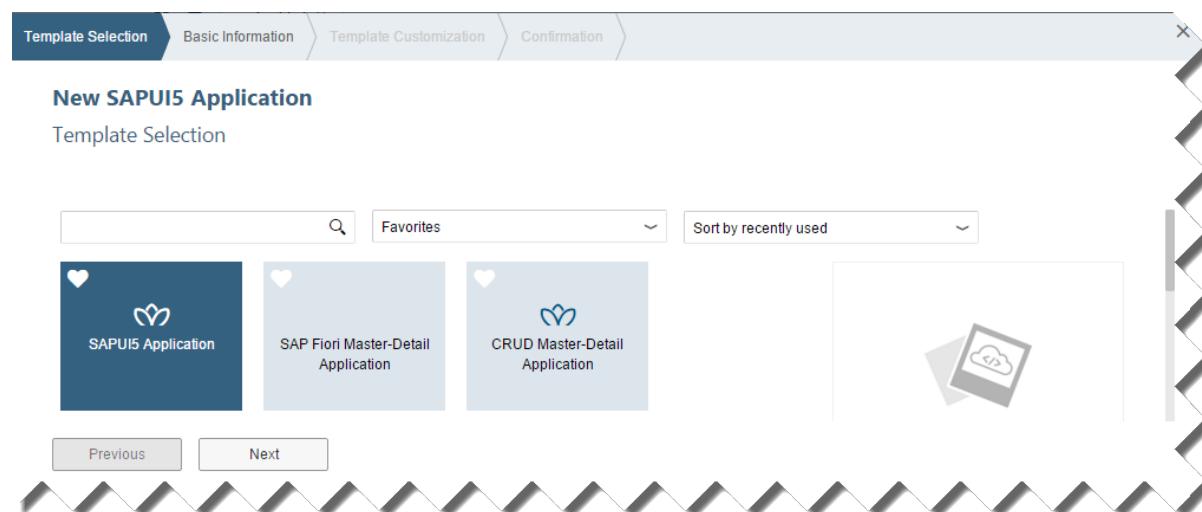


The *Workspace Actions* include:

- [Link Workspace to Editor](#)
- [Show Hidden Files](#)

Sort by Recently Used (New)

You can now sort the elements in the wizards (for example templates or components) according to the most recently used. The options displayed in the context menu will be sorted in this order as well.



Code Completion (Changed)

The code editor now provides code completion suggestions for SAPUI5 static and instance methods when you use dependency declarations with Asynchronous Module Definition (AMD).

Example of code completion for static methods:

```
sap.ui.define([
    "sap/ui/test/Opa5",

], function(Opa5) {
    "use strict";
Opa5..
fx createPageObjects(mPageObjects) : Object
fx emptyQueue() : Object
fx extendConfig() : undefined
fx getContext() : Object
```

Example of code completion for instance methods:

```
sap.ui.define([
    "sap/ui/test/Opa5",

], function(Opa5) {
    "use strict";
    var Opa5Instance = new Opa5();
    Opa5Instance..
fx iStartMyUIComponent(oOptions) : Object
fx iTeardownMyUIComponent() : Object
fx waitFor(oOptions) : Object
(x) assert : {ok,strictEqual}
(x) createPageObjects : {}
(x) getWindow : {}
(x) matchers : {PropertyStrictEquals}
```

For more information, see [Using Code Completion \[page 220\]](#).

Layout Editor (New, Changed)

Supported SAPUI5 Semantic Buttons (New)

i Note

Semantic buttons can only be added to these semantic pages:

- Detail Page
- Fullscreen Page
- Master Page

The following SAPUI5 semantic buttons are now supported in the layout editor:

- sap.m.semantic.AddAction
- sap.m.semantic.CancelAction
- sap.m.semantic.DiscussInJamAction
- sap.m.semantic.EditAction
- sap.m.semantic.FavoriteAction
- sap.m.semantic.FilterAction
- sap.m.semantic.FlagAction
- sap.m.semantic.ForwardAction
- sap.m.semantic.GroupAction
- sap.m.semantic.MainAction
- sap.m.semantic.MessagesIndicatorAction
- sap.m.semantic.MultiSelectAction
- sap.m.semantic.NegativeAction
- sap.m.semantic.OpenInAction
- sap.m.semantic.PositiveAction
- sap.m.semantic.PrintAction
- sap.m.semantic.SaveAction
- sap.m.semantic.SendEmailAction
- sap.m.semantic.SendMessageAction
- sap.m.semantic.ShareInJamAction
- sap.m.semantic.SortAction

For more information, see [SAPUI5 Controls Supported in the Layout Editor \[page 349\]](#).

Dialog Box for Adding SAPUI5 Semantic Buttons (New)

When you drag the *App* control to the canvas from the *Controls* tab, new dialog box opens, letting you select one or more semantic buttons to be added.

Annotation Modeler (Changed)

The annotation modeler has been enhanced with UI-based features:

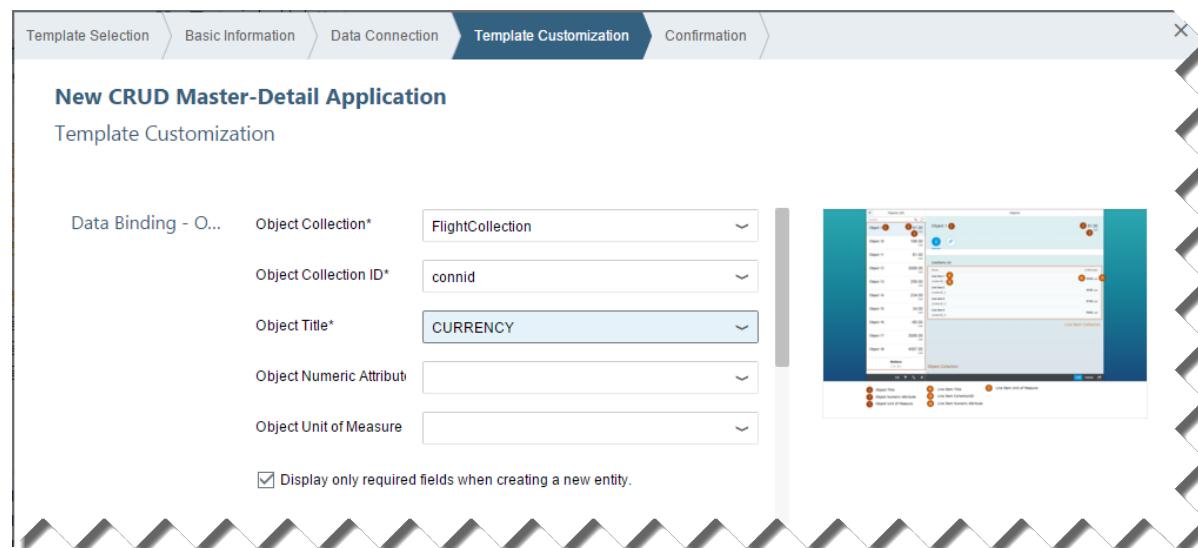
- In the Annotation Structure pane, newly created annotation terms are directly highlighted after creation, and their properties are immediately shown in the Properties pane.

- Editable annotation terms can be distinguished from non-editable terms at a glance using different font colors.
- Checks for qualifiers have been implemented to ensure that annotation terms are uniquely identified if used multiple times in the annotation file.

For more information, see [Annotation Modeler \[page 369\]](#).

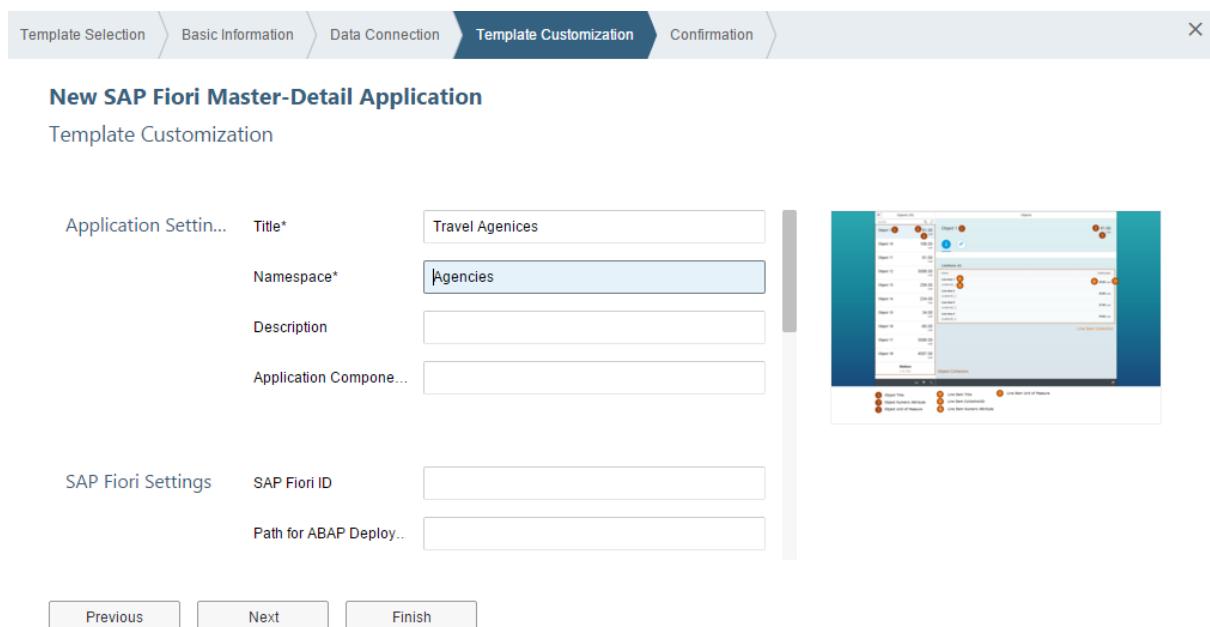
CRUD Template (Changed)

When creating a project using the CRUD template, in the *Template Costumization* step you can choose to only have the required fields visible when creating a new element.



New Project from Template Wizard (Changed)

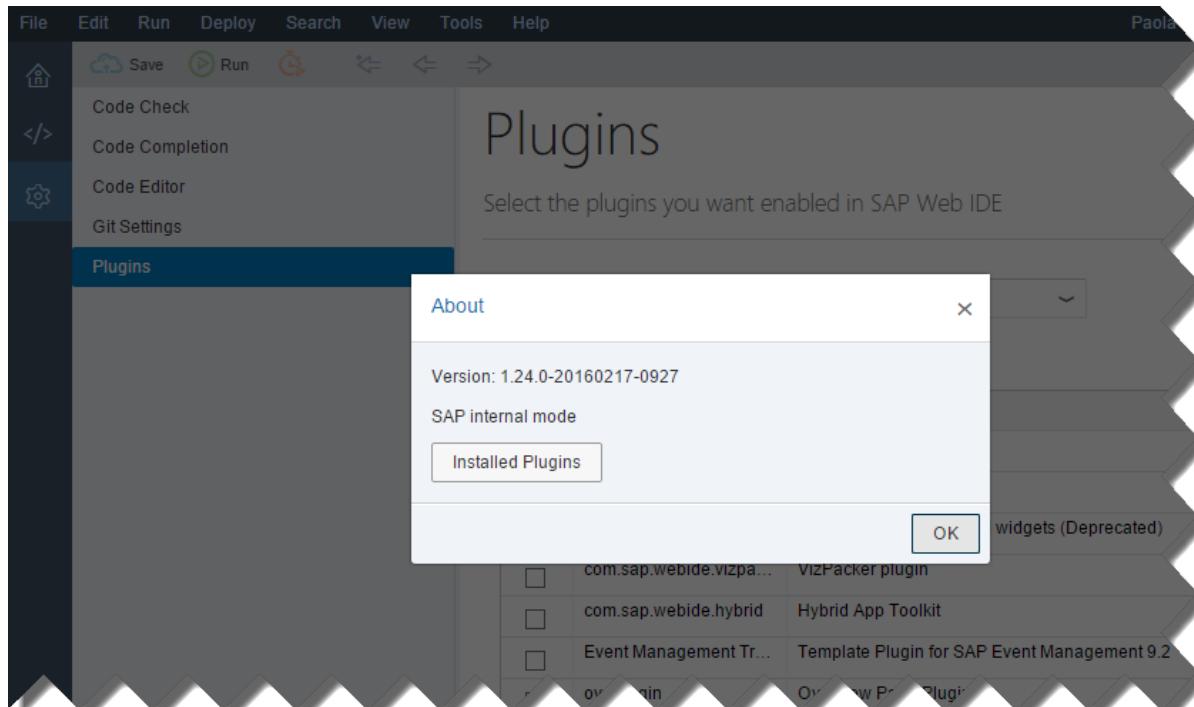
You can now trigger the project generation from any step when all other steps after it are optional.



List of Installed Plugins (Changed)

The list of installed plugins (previously called *Available Plugins*) has moved from the Preferences page to the *Help* menu.

To open the list, go to *Help* *About* and click *Installed Plugins*.



Compare Editor (Changed)

- The read-only file is now indicated by background shading.
- As you navigate through the changes in the file, the color of the changed code deepens to indicate your cursor position.

For more information, see [Compare Code Versions \[page 417\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)

[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.17 January 2016

Learn about the new and changed features that became available on January 2016.

For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

SAP Web IDE Versioning

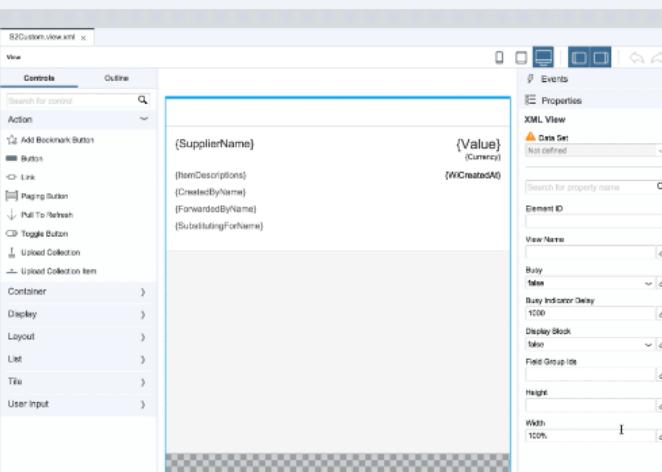
SAP Web IDE is moving to a new versioning concept. The versions will no longer be formatted as in previous versions (1.17, 1.18, 1.19). As of this release, we will use the date to identify the version. You can see the current version under [Help > About](#). The version appears in yyymmdd format (for example, 160128).

Tips and Tricks (New)

A dialog opens when you start up SAP Web IDE providing a description and a visual reference of how to use selected features.

You can open the Tips and Tricks dialog from the [Help](#) menu as well.

Tips and Tricks



Open Layout Editor from the Extensibility Pane

You can open an extended view in the layout editor directly from the extensibility pane. In the Outline, right-click the extended view, then in the context menu select "Open Layout Editor". You can also open it from the Open button at the bottom of the pane.

Don't show on startup

[Previous](#) [Next](#) [Close](#)

CRUD Template (Changed)

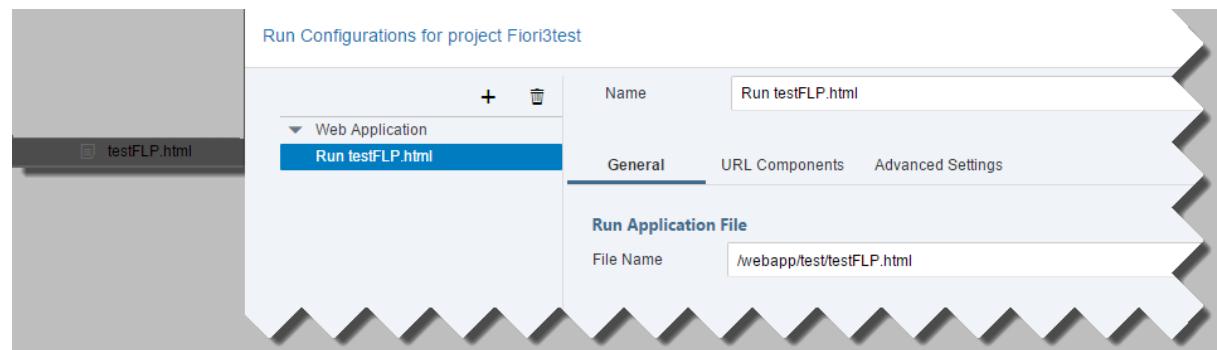
The *CRUD Master-Detail Application* template now includes Update capabilities.

Run Configurations (Changed)

The following changes have been made to *Run Configurations*:

- When you create a new run configuration, the default name is Run <run_application_filename>. If the run application file is not known, the default name is Configuration.
- The name of the field for the path of the file that runs the application is now *Run Application File*.
 - If you choose *Run Configurations* from the context menu of the file that runs the application, this file name appears in the *Run Application File* field.
 - If you choose *Run Configurations* for a project with only one file that can run the application, this file name appears in the *Run Application File* field.
 - In all other cases, you select the file from the list of possible options.

In the following example, *Run Configurations* was chosen from the context menu for the file testFLP.html in the project *Fiori3test*.



- [Create Run Projects \[page 435\]](#)
- [Create Run Projects \[page 435\]](#)
- [Create Run Projects \[page 435\]](#)

For more information, see [Create Run Projects \[page 435\]](#).

Code Editor (Changed)

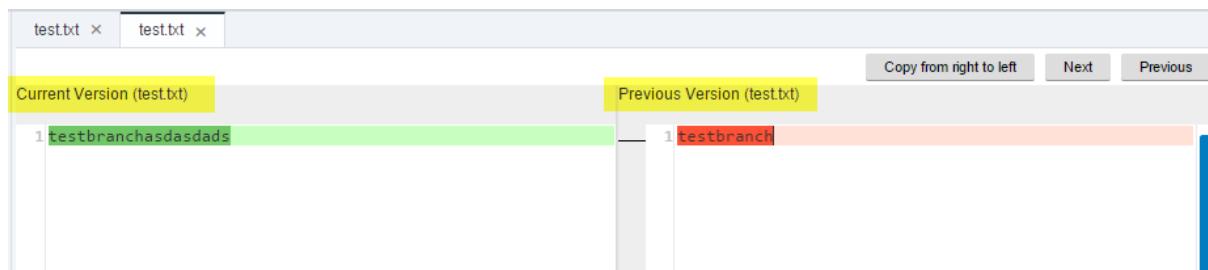
The following improvements have been made to the code editor:

- You can move between code editor tabs:
 - $\text{Alt} + \text{U}$ - Move one tab to the right.
 - $\text{Alt} + \text{Q}$ - Move one tab to the left.
- Code editor tabs load much more quickly after SAP Web IDE is refreshed.

For more information, see [Working in the Code Editor \[page 214\]](#).

Compare Editor (Changed)

The panes in the Compare editor that you access from the Git pane, now have titles. The title of each pane indicates whether it contains the latest editable version or the previous read-only version.



For more information, see [Compare Code Versions \[page 417\]](#).

Extensibility Pane (Changed)

You can now open extended extension points in the layout editor from the extensibility pane.

For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).

Deployment to SAP HANA Cloud Platform (Changed)

You can now deploy to SAP HANA Cloud Platform without automatically creating a Git repository.

This enables you to deploy applications that are connected to non-SAP HANA Cloud Platform Git repositories.

For more information, see [Deploy Applications to SAP Cloud Platform \[page 451\]](#).

Layout Editor (New, Changed)

Supported SAPUI5 Control (New)

The `sap.m.App` control is now supported in the layout editor.

For more information about SAPUI5 controls, see [UI development toolkit for HTML5 - Demo Kit](#).

Create a Function on Events Pane (New)

You can now create a new function for a controller from the `Events` pane.

For more information, see [Create a New Function \[page 340\]](#).

Improved Usability (Changed)

Usability has been improved for the following:

- On the *Outline* tab, when you are adding a control using the *Add Control* dialog box, the following are now provided for each control in the list:
 - Icon
 - Technical name
 - TooltipFor more information, see [Add Controls from the Outline Tab \[page 339\]](#).
- On the *Controls* tab, when you are dragging and dropping a control onto the canvas, there are now tooltips displaying the names of the exact drop targets.
For more information, see [Add Controls from the Controls Tab \[page 341\]](#).
- In the *Data Binding* dialog box, you can now add data fields to the string in the *Expression* box by double-clicking them in the *Data Fields* list.
For more information, see [Bind Data to a Simple Control \[page 344\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)
- [SAP Web IDE 1.17 \[page 99\]](#)
- [SAP Web IDE 1.16 \[page 103\]](#)
- [SAP Web IDE 1.15 \[page 106\]](#)
- [SAP Web IDE 1.14 \[page 109\]](#)
- [SAP Web IDE 1.13 \[page 111\]](#)
- [SAP Web IDE 1.12 \[page 114\]](#)
- [SAP Web IDE 1.11 \[page 118\]](#)

[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.18 SAP Web IDE 1.19

Learn about the new and changed features that are available for SAP Web IDE 1.19.

Problems View (New)

You can now view information about problems in the projects in your workspace in a dedicated *Problems* view. When you select one or more projects, SAP Web IDE automatically analyzes the selected projects for problems. You can also choose to analyze all the projects in your workspace.

SAP Web IDE analyzes and displays information for the following file types:

- JavaScript files - Syntax errors and ESLint validation errors
- XML files - Syntax errors
- JSON files - Syntax errors

To open the *Problems* view, select **View > Problems**.



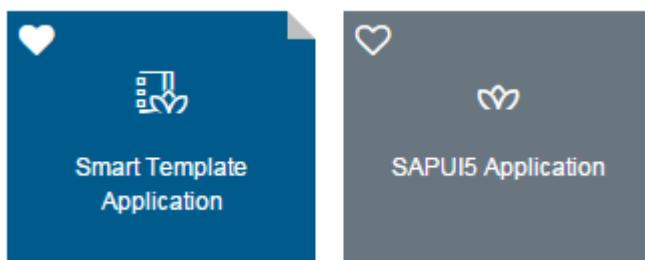
Problems for Project nw.epm.refapps.purchaseorders.manage			
Severity	Description	File	Location
⚠	ESLint: (no-warning-comments) Unexpected todo comment.	MockRequests.js(109, 3)	/nw.epm.refapps.purchaseor...
ℹ	ESLint: (no-param-reassign) Assignment to function parameter 'sTargetId'.	MockRequests.js(114, 4)	/nw.epm.refapps.purchaseor...
⚠	ESLint: (no-warning-comments) Unexpected todo comment.	MockRequests.js(237, 3)	/nw.epm.refapps.purchaseor...
ℹ	ESLint: (no-param-reassign) Assignment to function parameter 'sHash'.	Common.js(12, 3)	/nw.epm.refapps.purchaseor...
ℹ	ESLint: (no-param-reassign) Assignment to function parameter 'sHash'.	Common.js(15, 3)	/nw.epm.refapps.purchaseor...
ℹ	ESLint: (no-param-reassign) Assignment to function parameter 'oOptions'.	Common.js(25, 3)	/nw.epm.refapps.purchaseor...
ℹ	ESLint: (no-param-reassign) Assignment to function parameter 'sUrlParamet	Common.js(8, 2)	/nw.epm.refapps.purchaseor...

- [Using the Problems View \[page 323\]](#)

For more information, see [Using the Problems View \[page 323\]](#).

Favorites Template Category (New)

In the *New Project from Template* wizard, a *Favorites* category has been added to the list of categories. You can click on the heart on the relevant tile to add it to this category.



CRUD Template (New)

The template library now includes the *CRUD Master-Detail Application* template that displays data from an OData service using the master-detail pattern and includes Create, Read, and Delete capabilities. This template belongs to the *SAP Fiori Application* category.

Hide/Show Hidden Files (New)

From the *View* menu, you can hide or show hidden files that are generated by SAP Web IDE, for example, the `.user.project.json` file.

The `.user.project.json` file contains the project settings for a specific user.

Validation of Application Descriptor Files (New)

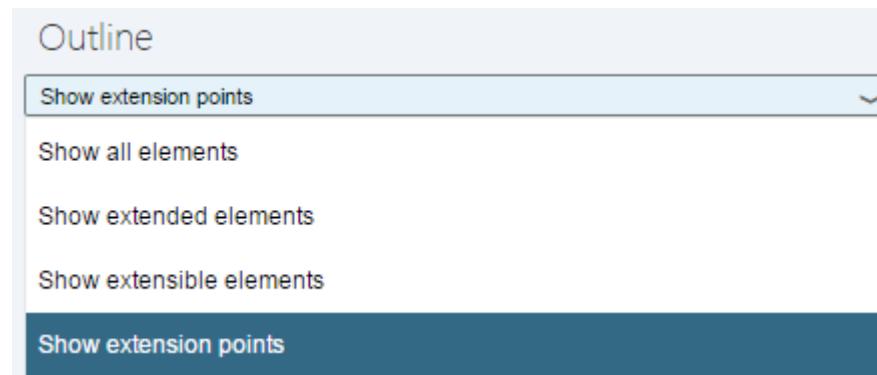
When you make changes to your project's `neo-app.json` file in SAP Web IDE, schema validation is performed automatically and any errors are displayed as annotations.

For more information, see [Checking Code \[page 239\]](#).

Extensibility Pane (New)

From the extensibility pane, you can now choose to show all extension points available in the application

(identifiable by the  icon).

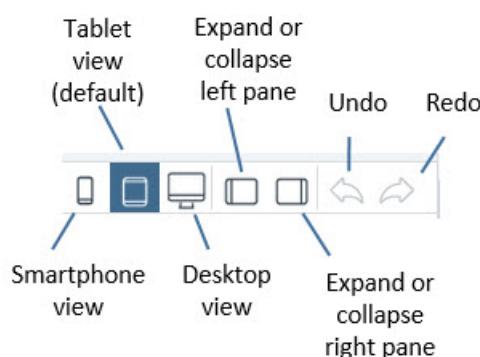


For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).

Layout Editor (Changed)

Toolbar

A new toolbar has been added to the layout editor, as shown below:



You can use the buttons on the toolbar to change the device format view, expand or collapse the left or right pane, and undo or redo actions.

For more information, see [Layout Editor \[page 332\]](#).

Supported SAPUI5 Semantic Pages

The following SAPUI5 semantic pages are now supported for specific operations (see below for a list of the supported operations):

- Detail Page
- Fullscreen Page
- Master Page
- Semantic Page
- Share Menu Page

The following operations are supported for the abovementioned semantic pages:

- Modify content.
- Add controls to the content aggregations from the *Controls* and the *Outline* tabs.
- Select controls to content aggregations from the canvas as well as from the *Outline* tab.
- Edit properties of semantic pages and their content.

Run Configurations (Changed)

The user interface to manage run configurations has been updated. The run configuration settings are now organized in tabs, which are displayed according to the type of run configuration. The *Run Configurations* window is accessible from the *Run* menu, as previously, but is no longer accessible from *Project Settings*.

→ Tip

Mouse over the image for more information about the new *Run Configurations* user interface.

Run Configurations for project nw.epm.refapps.purchaseorders.manage

Name **New Web Application 2**

General URL Components Advanced Settings

Runnable File

File Name /webapp/test.html

Preview Mode

Without Frame With Frame

Mock Data

Run with mock data

- [Create Run Projects \[page 435\]](#)

The only change in functionality is that all defined URL parameters are enabled and added to the URL at runtime.

For more information, see [Create Run Projects \[page 435\]](#).

Registration to SAP Fiori Launchpad (Changed)

New functionality has been added to the *Assignment* step in the *Register to SAP Fiori launchpad* wizard.

You can now select the SAP Fiori launchpad site to which you want to register your deployed application.

For more information, see [Register Applications to SAP Fiori Launchpad \[page 454\]](#).

Application Descriptor Editor (Changed)

Two new tabs have been added to the application descriptor editor:

- *Routing* tab for configuration of application routing.
- *Data Sources* tab for configuration of the OData services that are the data sources for the application.

For more information, see [Modifying the Application Descriptor Configuration \[page 205\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)

[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.19 SAP Web IDE 1.18

Learn about the new and changed features that are available for SAP Web IDE 1.18.

For information about new and changed features in previous releases, see [What's New in Previous Releases of SAP Web IDE \[page 7\]](#).

Featured Templates Category (New)

When creating a new project using the wizard, there is a new *Featured* template category that displays the most commonly-used templates. These templates all follow the recommended SAP guidelines.

For more information, see [Create Projects from Templates \[page 181\]](#).

Deployment to Different Accounts (New)

For the first deployment, you can deploy an application to an account on SAP HANA Cloud Platform that is different to the one where you are running SAP Web IDE. You must be a member of the account to which you wish to deploy.

For more information, see [Deploy Applications to SAP Cloud Platform \[page 451\]](#).

Extending Applications Containing Multiple Component.js Files (New)

You can now create an extension project for project that contains multiple Component.js files. You can then create extensions to enhance the project.

URL Parameters in the Extensibility Pane (New)

In the Run Configurations page, you can now add URL parameters that are added to the application's URL when running the extensibility pane.

For more information, see [Create Run Projects \[page 435\]](#).

SAP Enterprise Portal Plugin (New)

The SAP Enterprise Portal plugin enables you to deploy any application or plugin from SAP Web IDE to the SAP Enterprise Portal.

For more information, see [Plugins and Add-Ons \[page 472\]](#).

Cross-File Navigation (Changed)

You can now navigate to variable definitions in other files within the same project when you use dependency declarations with Asynchronous Module Definition (AMD).

For more information, see [Working in the Code Editor \[page 214\]](#).

Mock Data Editor (Changed)

The mock data editor now supports the new SAPUI5 project template structure. In these projects, metadata is stored under the `webapp/localService` folder, and mock data `json` files are stored in the `mockdata` subfolder under `webapp/localService`.

For more information, see [Edit Mock Data \[page 444\]](#).

SAP Fiori Overview Page Plugin (Changed)

The SAP Fiori Overview Page plugin is now available for productive use.

For more information, see [Plugins and Add-Ons \[page 472\]](#).

Annotation Modeler (Changed)

The usability for adding annotations have been improved and new user interface features for concatenating values and internationalized labels have been added.

For more information, see [Annotation Modeler \[page 369\]](#).

Layout Editor (New, Changed)

Supported SAPUI5 Controls (New)

There are 68 additional SAPUI5 controls that are now supported in the layout editor.

Add SAPUI5 Controls from the Outline Tab (New)

You can now add any supported SAPUI5 control from the new *Outline* tab.

For more information, see [Add Controls from the Outline Tab \[page 339\]](#) and [Delete Controls from the Outline Tab \[page 339\]](#).

Cut, Copy, and Paste (New)

You can now use cut, copy, and paste functionality when adding controls from the *Outline* tab to the canvas. Either right-click and use the options in the context menu or use the *Edit* menu.

In addition to the *Cut*, *Copy*, and *Paste* options, you can now choose *Paste Before* or *Paste After* from the context menu.

Events Area in Properties Pane (New)

In the properties pane to the right of the canvas, a new *Events* area allows you to select an existing event handler from the controller for an event of the selected control.

For more information, see [Layout Editor \[page 332\]](#).

Layout (Changed)

The *Controls* tab now replaces the palette, and the *Outline* tab now replaces the *Outline* pane. These new tabs are located to the left of the canvas.

For more information, see [Layout Editor \[page 332\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)

[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.20 SAP Web IDE 1.17

Learn about the new and changed features that are available for SAP Web IDE 1.17.

Resetting the Default JavaScript Validation Rules (New)

You can now reset the validation configuration and rules in the default JavaScript ESLint validator to the default configuration.

For more information, see [Customize JavaScript Validator Configuration \[page 245\]](#).

SAP Translation HUB Service (New, Experimental)

The SAP Translation HUB service is now integrated into SAP Web IDE. You can enter a string in the code editor and use auto-complete (`CTRL` + `SPACE`) to automatically externalize the term to the predefined properties file (`i18n`).

i Note

The SAP Translation HUB service and all features that come with it are experimental and are not supported for productive use.

SAP Fiori Smart Template Application Template (New)

The *SAP Fiori Smart Template Application* template creates SAP Fiori applications that are based on OData services and annotations that require no JavaScript UI coding.

An app that is based on smart templates uses predefined template views and controllers that are provided centrally, so no application-specific view instances are required.

The SAPUI5 runtime interprets metadata and annotations of the underlying OData service and uses the corresponding views for the SAP Fiori app at start-up.

For more information, see [Smart Templates: Create Apps Without Front-End Code](#).

Application Status (New)

You can now see whether your application is deployed to SAPUI5 ABAP Repository. If the application is already deployed to SAPUI5 ABAP Repository, its system, name, package, and URL are displayed. If it is not deployed, you can choose to deploy it.

For more information, see [Check the Application Status \[page 456\]](#).

Annotation Modeler (New)

The Annotation Modeler provides an intuitive user interface for annotating OData entities. Depending on the selection, the Annotation Modeler displays different panes that allow you to add annotation terms to the local annotation file, edit them, or override annotation terms coming from other sources into the local annotation file.

For more information, see [Annotation Modeler \[page 369\]](#).

SAPUI5 Versions in Unit Test Run Configurations (New)

You can now choose a specific SAPUI5 version in run configurations for unit tests.

For more information, see [Create Run Projects \[page 435\]](#).

Cross-File Code Completion (Changed)

SAP Web IDE can now display code completion suggestions from other files within the same project when you use dependency declarations with Asynchronous Module Definition (AMD).

For more information, see [Using Code Completion \[page 220\]](#).

Layout Editor (New, Changed)

Additional Supported Controls (New)

The following SAPUI5 controls have been added to the palette in the layout editor.

SAPUI5 Controls

Control Display Name	Palette Area	SAPUI5 Control Name
Add Bookmark Button	Action	sapushell.ui.footerbar.AddBookmarkButton
Paging Button	Action	sap.m.PagingButton
Toggle Button	Action	sap.m.ToggleButton
Carousel	Container	sap.m.Carousel
Flex Box	Container	sap.m.FlexBox
Icon Tab Header	Container	sap.m.IconTabHeader
Icon Tab Separator	Container	sap.m.IconTabSeparator
Message Strip	Container	sap.m.MessageStrip
Nav Container	Container	sap.m.NavContainer
Overflow Toolbar	Container	sap.m.OverflowToolbar
Toolbar Separator	Container	sap.m.ToolbarSeparator
Busy Indicator	Display	sap.m.BusyIndicator
Invisible Text	Display	sap.ui.core.InvisibleText
Item	List	sap.ui.core.Item
Select List	List	sap.m.SelectList
Feed Input	User Input	sap.m.FeedInput
Radio Button Group	User Input	sap.m.RadioButtonGroup

For more information, see [UI development toolkit for HTML5 - Demo Kit](#).

Layout Editor Tutorial (Changed)

The tutorial about building a new application with the layout editor has been completely updated.

For more information, see [Try It: Build an Application with the Layout Editor \[page 362\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.21 SAP Web IDE 1.16

Learn about the new and changed features that are available for SAP Web IDE 1.16.

Application Descriptor Editor (New)

You can now modify your project application descriptor file (`manifest.json`) in a multi-tab editor that contains two subtabs: *Code Editor* and *Descriptor Editor*.

Schema validation is automatically performed on `manifest.json` files that are open in the *Code Editor* tab.

The *Descriptor Editor* subtab includes two AppDescriptor configuration subtabs: *Settings* and *Navigation*. The configuration fields in the Descriptor editor assist you by providing available choices, placeholder suggestions, and input validation.

To use the Application Descriptor editor, open your project `manifest.json` file in your workspace, and choose the *Descriptor Editor* subtab at the bottom of the pane. The changes that you make in the *Descriptor Editor* subtab are automatically updated in the *Code Editor* subtab and vice versa.

For more information, see [Modifying the Application Descriptor Configuration \[page 205\]](#).

Application Destination Mapping (New)

You can now test your application with a destination system that is different from the one defined in the application `neo-app.json` file. In the new *Application Destination Mapping* section in the *Run Configuration* settings, you can map destinations that are defined in your project to any system that is included in your SAP HANA Cloud Platform account.

For more information, see [Create Run Projects \[page 435\]](#).

Logout Page (New)

After logging out, a new page is displayed indicating that you are logged out and including a link that allows you to log back in.

Overview Page Plugin (New, Experimental)

The Overview Page plugin for SAP Web IDE allows developers to create their own overview page applications. Overview pages provide users with an interactive overview of a specific area, enabling them to interact with business data and perform key actions.

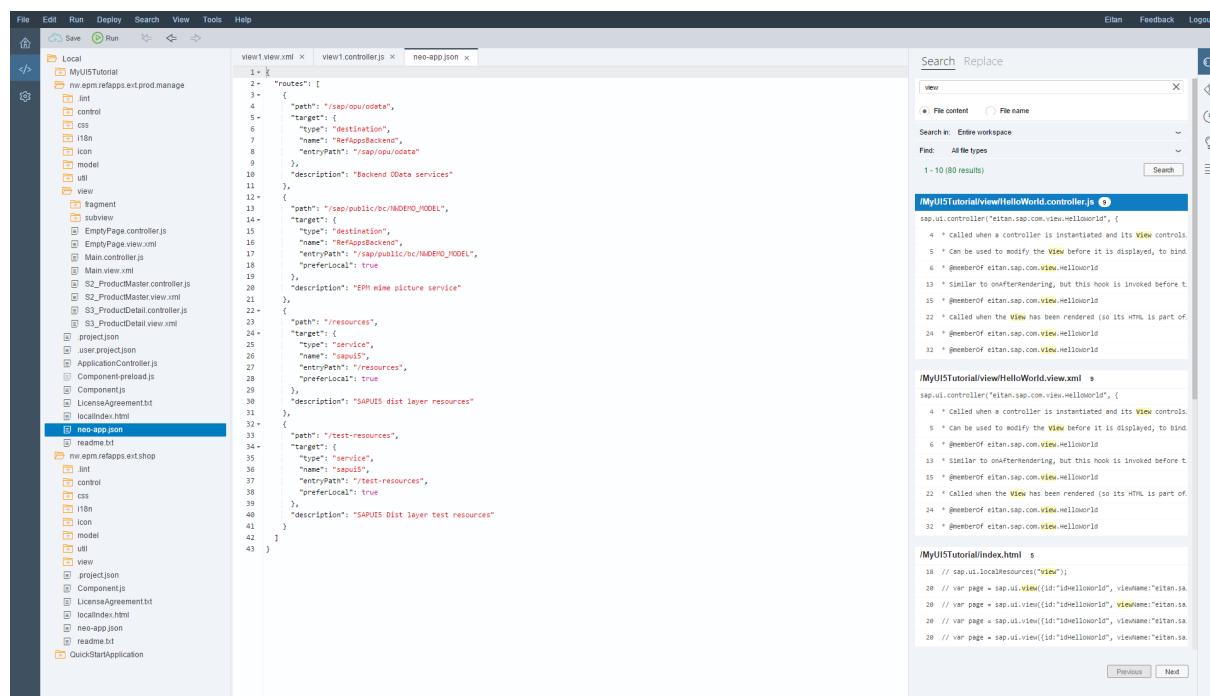
i Note

The Overview Page plugin and all features that come with it are experimental and not supported for productive use.

For more information, see [Plugins and Add-Ons \[page 472\]](#).

Improved UI (Changed)

As part of an enhancement to the SAP Web IDE appearance, you can now enjoy a new UI design.



Run Configuration Without a Frame (Changed)

When you run an application, it now opens with no frame, by default. You can change this setting in the [Frame](#) field in the [Run Configuration](#) settings.

For more information, see [Create Run Projects \[page 435\]](#).

Create Multiple Local Branches from Remote Branch (Changed)

You can now create more than one local branch from a remote branch.

For more information, see [Use Multiple Local Branches \[page 423\]](#).

Build Creates Resources File (Changed)

When you build an application, a `resources.json` file is created listing all the resources in the project.

For more information, see [Application Build \[page 456\]](#).

Project Shows Current Checked-Out Branch (Changed)

In the workspace, each project that is under source control indicates the branch that is currently checked out.

For more information, see [Use Multiple Local Branches \[page 423\]](#).

Hybrid App Toolkit Add-on Enhancements (Changed)

The Hybrid App Toolkit add-on has new and improved features.

For more information, see *What's New in HAT in SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments*.

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)

[SAP Web IDE 1.19 \[page 90\]](#)

[SAP Web IDE 1.18 \[page 96\]](#)

[SAP Web IDE 1.17 \[page 99\]](#)

[SAP Web IDE 1.15 \[page 106\]](#)

[SAP Web IDE 1.14 \[page 109\]](#)

[SAP Web IDE 1.13 \[page 111\]](#)

[SAP Web IDE 1.12 \[page 114\]](#)

[SAP Web IDE 1.11 \[page 118\]](#)

[SAP Web IDE 1.10 \[page 121\]](#)

[SAP Web IDE 1.9 \[page 124\]](#)

[SAP Web IDE 1.8 \[page 127\]](#)

[SAP Web IDE 1.7 \[page 131\]](#)

[SAP Web IDE 1.6 \[page 136\]](#)

[SAP Web IDE 1.5 \[page 138\]](#)

[SAP Web IDE 1.4 \[page 140\]](#)

[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)

[SAP Web IDE 1.2 \[page 146\]](#)

2.1.22 SAP Web IDE 1.15

Learn about the new and changed features that are available for SAP Web IDE 1.15.

Cross-file Navigation for Function Definitions and Code Completion (New)

- You can now use the context menu [Goto Definition](#) option to navigate to a function definition in another file in the same project.
- When using code completion, (`Ctrl` + `Space`), suggestions from other files within the same project are now displayed.

i Note

These features are available only when using SAPUI5 module patterns; target files contain the `jQuery.sap.declare` statement and the source file contains the `jQuery.sap.require`.

For more information, see [Working in the Code Editor \[page 214\]](#) and [Using Code Completion \[page 220\]](#).

Embedded Type Code Completion (New)

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.

The following example is for adding a button to the UI.

↳ Sample Code

```
/* @type sap.m.Button */
var x;
x.
```

```
formatConcatenateNameId : function(sName, sID) {
    /** @type sap.m.Button */
    var x = sap.ui.getCore().byId("btnBack");
    x.s
    M setActiveIcon(sActiveIcon) : sap.m.Button
    M setEnabled(bEnabled) : sap.m.Button
    M setIcon(sIcon) : sap.m.Button
    M setIconDensityAware(bIconDensityAware) : sap.m.Button
    M setIconFirst(bIconFirst) : sap.m.Button
    M setText(sText) : sap.m.Button
    M setTextColor(sTextDirection) : sap.m.Button
};
```

setIconDensityAware(bIconDensityAware) : sap.m.Button

Sets a new value for property `iconDensityAware`. By default, this is set to true but then one or more requests are sent trying to get the density perfect version of image if this version of image doesn't exist on the server. If only one version of image is provided, set this value to false to avoid the attempt of fetching density perfect image. When called with a value of `null` or `undefined`, the default value of the property will be restored. Default value is `true`.

For more information, see [Using Code Completion \[page 220\]](#).

German Keyboard Shortcuts for Toggle Comment Actions (New)

You can use the following keyboard shortcuts for German language keyboards to perform [Toggle comment](#) actions.

	Microsoft Windows	Mac OS
Toggle line comment	[Alt] + [7]	[Alt] + [7]
Toggle block comment	[Ctrl] + [Shift] + [7]	[Command] + [Shift] + [7]

For more information, see [Keyboard Shortcuts \[page 166\]](#) and [Working in the Code Editor \[page 214\]](#).

Layout Editor: Quick Start Application (New)

On the Welcome page, there is a new tile, [Quick Start with Layout Editor](#), to quickly create a new application, enabling you to start coding immediately. With just one click, you can create an application that you can run using the provided mock data. A data set for data binding and a metadata file are included.

For more information, see [Create a Quick Start Application with the Layout Editor \[page 348\]](#).

Run Menu Options (Changed)

The options in the *Run* menu in the main toolbar and context menu have been reorganized and enhanced:

- The first section in the *Run* menu options displays up to five run configurations that you can use for the selected project. The run configurations are listed according to recent usage; the last run configuration used is first in the list.
- The *Run Configurations* option is now named *New Configuration*. Choose this option to define a new run configuration for your project.
- The *Run As* submenu and the other options remain unchanged.

For more information, see [Running Applications in Development Mode \[page 432\]](#).

Hybrid App Toolkit Add-on Enhancements (Changed)

The Hybrid App Toolkit add-on has new and improved features.

For more information, see [What's New in HAT in SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)

SAP Web IDE 1.17 [page 99]
SAP Web IDE 1.16 [page 103]
SAP Web IDE 1.14 [page 109]
SAP Web IDE 1.13 [page 111]
SAP Web IDE 1.12 [page 114]
SAP Web IDE 1.11 [page 118]
SAP Web IDE 1.10 [page 121]
SAP Web IDE 1.9 [page 124]
SAP Web IDE 1.8 [page 127]
SAP Web IDE 1.7 [page 131]
SAP Web IDE 1.6 [page 136]
SAP Web IDE 1.5 [page 138]
SAP Web IDE 1.4 [page 140]
SAP Web IDE 1.3 (Trial Only) [page 145]
SAP Web IDE 1.2 [page 146]

2.1.23 SAP Web IDE 1.14

Learn about the new and changed features that are available for SAP Web IDE 1.14.

Layout Editor (New, Changed)

Additional Supported Controls (New)

The following SAPUI5 controls have been added to the palette in the layout editor.

SAPUI5 Controls

Control Display Name	Palette Area	SAPUI5 Control Name
Upload Collection	Action	sap.m.UploadCollection
Upload Collection Item	Action	sap.m.UploadCollectionItem

For more information, see <https://sapui5.hana.ondemand.com/sdk/#content/Controls/index.html>.

Cut, Copy, and Paste Controls and Containers

In the layout editor, on the canvas, you can now cut, copy, and paste controls and containers, including all their properties. In addition, you can now paste these items into a different view.

SAPUI5 Versions

The layout editor now supports different SAPUI5 versions.

For more information, see [Create Run Projects \[page 435\]](#).

JSDoc Comment Generation (New)

You can now generate a JSDoc comment snippet for a function in your code. The JSDoc comment snippet provides a template for documenting the function, which allows you to enter information about function arguments and parameters in the corresponding placeholders.

For more information, see [Generate JSDoc Comment Snippets \[page 218\]](#).

URL Fragment Identifier in Run Configurations (New)

You can now define a URL fragment identifier (also known as a hash fragment) in run configurations for web applications and SAP Fiori components. When you run the application, the fragment identifier is appended to the application URL after other URL parameters and is preceded by a hash (#) delimiter.

The *Run Configurations* pane contains a new section, *URL Components*, in which you define the fragment identifier, and other URL application parameters as required. This section replaces the *Application URL Parameters* section.

For more information, see [Create Run Projects \[page 435\]](#).

Application Build (New)

A build of the application is now performed when you deploy your application. The build includes minification of the application's JavaScript file and the collection of files into a distributable runtime package.

For more information, see [Application Build \[page 456\]](#).

Hybrid App Toolkit Add-on Enhancements (Changed)

The Hybrid App Toolkit add-on has new and improved features.

For more information, see [What's New in HAT in SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)

[April 2017 \[page 10\]](#)

[March 2017 \[page 12\]](#)

February 2017 [page 14]
January 2017 [page 18]
December 2016 [page 22]
November 2016 [page 31]
October 2016 [page 36]
September 2016 [page 40]
August 2016 [page 47]
July 2016 [page 51]
June 2016 [page 55]
May 2016 [page 60]
April 2016 [page 64]
March 2016 [page 73]
February 2016 [page 79]
January 2016 [page 86]
SAP Web IDE 1.19 [page 90]
SAP Web IDE 1.18 [page 96]
SAP Web IDE 1.17 [page 99]
SAP Web IDE 1.16 [page 103]
SAP Web IDE 1.15 [page 106]
SAP Web IDE 1.13 [page 111]
SAP Web IDE 1.12 [page 114]
SAP Web IDE 1.11 [page 118]
SAP Web IDE 1.10 [page 121]
SAP Web IDE 1.9 [page 124]
SAP Web IDE 1.8 [page 127]
SAP Web IDE 1.7 [page 131]
SAP Web IDE 1.6 [page 136]
SAP Web IDE 1.5 [page 138]
SAP Web IDE 1.4 [page 140]
SAP Web IDE 1.3 (Trial Only) [page 145]
SAP Web IDE 1.2 [page 146]

2.1.24 SAP Web IDE 1.13

Learn about the new and changed features that are available for SAP Web IDE 1.13

Deploy Menu (New)

A *Deploy* menu was added to the menu bar with the following options:

- Deploy to the SAPUI5 ABAP Repository

- Deploy to SAP HANA Cloud Platform
- Register to SAP Fiori Launchpad
- Application Status

API Management Integration (New)

You can now use the API Management system as the data source for your applications.

For more information, see [Create Projects from Templates \[page 181\]](#).

SAP Event Management Plugin (New)

The SAP Event Management plugin for SAP Web IDE enables developers to generate their own transactional SAP Fiori apps for SAP Event Management.

For more information, see [Plugins and Add-Ons \[page 472\]](#).

Layout Editor (New, Changed)

View the Canvas in Different Sizes (New)

On the canvas, you can preview the UI of your application as it will appear on different-sized device screens. You can access these views by clicking the new icons that appear at the bottom of the canvas:

- Cell phone  view
- Tablet  view
- Desktop  view

For more information, see [Layout Editor \[page 332\]](#).

Undo/Redo Supported in Canvas (New)

While working on the canvas, you can undo (`Ctrl` + `Z`) and redo (`Ctrl` + `Y`) any changes you have made.

Additional Supported Controls (New)

The following SAPUI5 controls have been added to the palette in the layout editor.

SAPUI5 Controls

Control Display Name	Palette Area	SAPUI5 Control Name
Calendar	User Input	sap.ui.unified.Calendar

Control Display Name	Palette Area	SAPUI5 Control Name
<i>Calendar Legend</i>	<i>Display</i>	<code>sap.ui.unified.CalendarLegend</code>
<i>Table</i>	<i>List</i>	<code>sap.m.Table</code>
<i>Column</i>	<i>List</i>	<code>sap.m.Column</code>
<i>Column List Item</i>	<i>List</i>	<code>sap.m.ColumnListItem</code>
<i>Grid</i>	<i>Layout</i>	<code>sap.ui.layout.Grid</code>
<i>Tile Container</i>	<i>Tile</i>	<code>sap.m.TileContainer</code>
<i>Standard Tile</i>	<i>Tile</i>	<code>sap.m.StandardTile</code>
<i>Pull To Refresh</i>	<i>Action</i>	<code>sap.m.PullToRefresh</code>

For more information, see <https://sapui5.hana.ondemand.com/sdk/#content/Controls/index.html>.

Open Code Tab from Canvas (Changed)

In the layout editor, from the context menu of any control in the canvas, you can go directly to a tab that displays the XML code for the control.

Open Fragment File from Canvas (Changed)

In the layout editor, from the context menu of any fragment in the canvas, you can go directly to the related fragment file.

For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).

Open Applications with No Index.html File (Changed)

In the layout editor, you can open applications that have no `index.html` file.

Hybrid App Toolkit Add-on Enhancements (Changed)

The Hybrid App Toolkit add-on has new and improved features.

For more information, see [What's New in HAT in SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)

[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.25 SAP Web IDE 1.12

Learn about the new and changed features that are available for SAP Web IDE 1.12.

Layout Editor Available for Productive Use (New)

The layout editor is no longer experimental. All its features are now available for productive use.

Layout Editor Enhancements (New)

- An SAPUI5 library containing SAP icons has been added for controls that display icons. In the *Select Icon* dialog box, when you make a selection from a list of SAPUI5 icons, the *Icon* field is now populated with the property name and the icon appears on the control.
- The properties most frequently used by each control appear at the top of the properties pane.
- Navigational bread crumbs appear at the bottom of the canvas.
- A *Page* control has been added to the palette.
- A message area has been added to the top of the canvas providing useful tips while you work in the layout editor.
- In the *Data Binding* dialog box, you can now enter keywords to filter the data fields on the left side of the window. For more information, see [Binding Data \[page 343\]](#).
- The default form factor for the canvas is now tablet size.
- A confirmation message is displayed before deleting a container.

For more information, see [Layout Editor \[page 332\]](#).

Access to the Layout Editor from the Extensibility Pane (New)

If you replace a view, you can open the layout editor from the extensibility pane to edit it.

For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).

Indication for Deprecated and Experimental APIs (New)

When using code completion, deprecated APIs are indicated with strikethrough text. The *API Reference* pane also includes information about deprecation and experimental status in its context-sensitive help.

For more information, see [Using Code Completion \[page 220\]](#).

Git Stash (New)

The Git Stash command has been added.

For more information, see [Stash Changes \[page 419\]](#).

Library Versions in Run Configuration (New)

Within a run configuration, you can now select the version for each library that is referenced by the application.

For more information, see [Create Run Projects \[page 435\]](#).

Hide Workspace (New)

You can now hide the workspace to make more room for editing your files.

For more information, see [Resize Panes \[page 178\]](#).

Code Completion (Changed)

The following enhancements have been made to code completion:

- Overall performance of code completion has improved.
- Code completion now supports multiple namespaces in XML.
- Code completion uses the latest version of SAPUI5 that was released to customers, unless an earlier SAPUI5 version is specified in the project `neo-app.json` file.

For more information, see [Using Code Completion \[page 220\]](#).

Import Stopped Applications from SAP HANA Cloud Platform (Changed)

You can import an existing application that is in STOPPED status from SAP HANA Cloud Platform into the SAP Web IDE workspace.

For more information, see [Import Applications from SAP Cloud Platform \[page 191\]](#).

Hybrid App Toolkit Add-on Enhancements (Changed)

The Hybrid App Toolkit add-on has new and improved features.

For more information, see *What's New in HAT in SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments*.

Git Tags in Git History (Changed)

In the Git History pane, a tag icon appears next to any commit that has been tagged. Hovering over the icon displays the tags.

For more information, see [Git History \[page 426\]](#).

Filter by Branch in Git History (Changed)

You can now filter the Git History pane by selected branches. Previously, you could only select one specific branch or all branches.

For more information, see [Git History \[page 426\]](#).

Search (Changed)

The workspace search has been enhanced as follows:

- The search is now done in realtime. As soon as a file is changed and saved, the new file contents are searchable.
- The search now matches strings within a word and not just at the start of a word.

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)

[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.26 SAP Web IDE 1.11

Learn about the new and changed features that are available for SAP Web IDE 1.11.

Feedback Form (New)

You can provide feedback for SAP Web IDE. Open the feedback form from the [Help](#) menu or click [Feedback](#) on the menu bar, near the SAP logo.

SAP Hana Cloud UI Theme Designer Access\ (New)

You can access the SAP HANA Cloud UI Theme Designer from the [Tools](#) menu or from the [Helpful Links](#) section on the Welcome page.

Hybrid App Toolkit Add-on Enhancements (Changed)

The Hybrid App Toolkit add-on has new and improved features.

For more information, see [What's New in HAT in SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments](#).

Mock Data Generation (Changed)

The mock data generation wizard now generates mock data files according to the entity sets, not according to the entity types as previously.

For more information, see [Edit Mock Data \[page 444\]](#).

Run Configurations (Changed)

You can now test Fiori components and web applications with any of the available versions of SAPUI5 that are deployed on SAP HANA Cloud Platform.

For more information, see [Create Run Projects \[page 435\]](#).

XML Formatting (Changed)

The XML beautifier now performs line wrapping in addition to indentation.

For more information, see [Working in the Code Editor \[page 214\]](#).

Code Completion (Changed)

Code completion now supports SAPUI5 1.26 and 1.28 libraries.

For more information, see [Supported SAPUI5 Libraries \[page 238\]](#).

JavaScript Code Checking (Changed)

You can now customize the JavaScript validator configuration for global and environment variables in addition to the validator rules.

For more information, see [Customize JavaScript Validator Configuration \[page 245\]](#).

Layout Editor (Changed, Experimental)

In previous versions of the layout editor, data binding capabilities were available only for views that contain custom data. Now, you can use data binding for all XML views.

i Note

The layout editor and all features that come with it are experimental and are therefore available only in the trial landscape.

For more information, see [Layout Editor \[page 332\]](#).

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.27 SAP Web IDE 1.10

Learn about the new and changed features that are available for SAP Web IDE 1.10.

UI Controller Hooks (New)

You can extend the functionality of a specific behavior in your extension projects' controllers using UI controller hooks. The hooks are visualized in the extensibility pane.

For more information, see [Extend Controllers \[page 464\]](#).

Application Status Page (New)

On the Application Status page you can see the application's deployment status together with its version information, URL, and status of registration to SAP Fiori launchpad.

For more information, see [Check the Application Status \[page 456\]](#).

Code Check Levels (New)

You can set the level of code checking to display all messages or only errors, warnings, and so on.

For more information, see [Configure Code Checking \[page 240\]](#).

XML Beautifier (New)

You can beautify XML code. The XML beautifier formats code with indentation of one tab space and is not configurable.

For more information, see [Working in the Code Editor \[page 214\]](#).

CSS Beautifier (New)

You can beautify CSS files. The CSS beautifier formats indentation with one tab space and is not configurable.

For more information, see [Working in the Code Editor \[page 214\]](#).

Tab Navigation (New)

You can navigate between tabs in the code editor using the navigation buttons in the toolbar or keyboard shortcuts. You can navigate backward and forward, and to the last edit location.

For more information, see [Work with Files and Folders \[page 176\]](#).

Hybrid App Toolkit add-on Enhancements (New)

The Hybrid App Toolkit add-on has new and improved features.

For more information, see *What's New in HAT 1.3* in [SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments](#).

Application Deployment to SAP HANA Cloud Platform (Changed)

You can deploy a new version of an application that was already deployed to SAP HANA Cloud Platform.

For more information, see [Deploy Applications to SAP Cloud Platform \[page 451\]](#).

Run Configurations (Changed)

The *Run* button in the open configuration has been renamed to *Run Now* and moved to below the configuration name. Its functionality has not changed.

If a required field is incomplete or incorrect, a message appears at the top of the *Run Configurations* pane, 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.

For more information, see [Create Run Projects \[page 435\]](#).

Toggle Comment Shortcuts (Changed)

The Toggle Comment keyboard shortcuts for Microsoft Windows are now:

- Toggle line comment: `Ctrl` + `/`
- Toggle block comment: `Ctrl` + `Shift` + `/`

For more information, see [Keyboard Shortcuts \[page 166\]](#).

Import and Export (Changed)

You can select the folder to which to import files or create a new folder from within the import wizard.

You can export a single file (in addition to exporting a project or folder).

For more information, see:

- [Work with Files and Folders \[page 176\]](#)
- [Import Projects from an Archive \[page 190\]](#)

Layout Editor (Changed, Experimental)

The following improvements have been added to the layout editor:

- You can open and edit a fragment file.
- You can see fragments as read-only components in the canvas.
- You can define data binding for control properties more easily.
- You can use comments, fragments, and extension points in the XML file of the view.

i Note

The layout editor and all features that come with it are experimental and are therefore available only in the trial landscape.

For more information, see [Layout Editor \[page 332\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)

[April 2017 \[page 10\]](#)

[March 2017 \[page 12\]](#)

[February 2017 \[page 14\]](#)

[January 2017 \[page 18\]](#)

[December 2016 \[page 22\]](#)

[November 2016 \[page 31\]](#)

[October 2016 \[page 36\]](#)

[September 2016 \[page 40\]](#)

[August 2016 \[page 47\]](#)

[July 2016 \[page 51\]](#)

[June 2016 \[page 55\]](#)

[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.28 SAP Web IDE 1.9

Learn about the new and changed features that are available for SAP Web IDE 1.9.

Welcome Page (New)

When accessing SAP Web IDE for the first time, or whenever a new version is released, a welcome page with useful links and common actions is displayed.

Performing Actions on Multiple Items (New)

You can select multiple files or folders in the workspace and perform an action on all of the selected items simultaneously; for example, copy, cut, or delete. Actions that are not applicable to the multi-selection are disabled in the menu.

For more information, see [Work with Files and Folders \[page 176\]](#).

Validating Syntax for XML Files (New)

Syntax validation for XML files is now supported.

For more information, see [Configure Code Checking \[page 240\]](#).

Beautifying JSON Files (New)

You can now beautify your code from the context menu of JSON files.

For more information, see [Working in the Code Editor \[page 214\]](#).

Validating JavaScript (New)

- You can customize code checking rules in the JavaScript validator tool or edit the default rules.
- You can create your own code checking rules and browse for and choose the folder containing them for a selected project.
- Fatal syntax errors are reflected in the header of the file tab, which displays an error icon and tooltip when you save your code.
- The JSHint code checking tool is deprecated.

For more information, see [Customize Code Checking Rules \[page 196\]](#).

Code Completion for Property Files (New)

There is now code completion support for easily inserting code snippets into your `i18n` and `messagebundle` property files. As you type, the code snippets are highlighted, making it easier to navigate between them.

For more information, see [Code Completion for Property Files \[page 223\]](#).

Context Menu Support for Code Editors (New)

- You can now use the Goto definition from the context menu of your JavaScript files.
- You can also use the gutter context menu (or a keyboard shortcut) to expand and collapse the entire hierarchy of your JS, XML, JSON, CSS, and HTML files.

For more information, see [Working in the Code Editor \[page 214\]](#).

Configuring Unit Tests (New)

You can run your application as a unit test to determine if it is working properly. You can configure as many run configurations as you need to test your project on different HTML files or non-HTML files.

For more information, see:

- [Run Applications from the Workspace \[page 433\]](#)
- [Create Run Projects \[page 435\]](#)

Importing Projects (Changed)

When importing from SAP HANA Cloud Platform or the ABAP repository, a new folder is created, regardless of what was selected in the workspace.

You can no longer import from SAP HANA Cloud Platform or from the ABAP repository using the project's context menu. This option can be found under .

For more information, see [Importing Projects \[page 190\]](#).

Registering to SAP Fiori Launchpad (Changed)

Support is now enabled for multiple SAP Fiori Launchpad subscriptions.

For more information, see [Register Applications to SAP Fiori Launchpad \[page 454\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)

[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.29 SAP Web IDE 1.8

Learn about the new and changed features that are available for SAP Web IDE 1.8.

SAP Fiori Full Screen Template (New)

The new SAP Fiori Full Screen template is now available.

Defining Languages (New)

You can now 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.

For more information, see:

- [Create Run Projects \[page 435\]](#)
- [Define Application Languages \[page 197\]](#)

Setting Project Type (New)

Each new project is now configured with a base project type, which you cannot change. You can select additional project types to add type-specific behaviors to the project.

For more information, see [Set Project Types \[page 200\]](#).

Defining Run Configurations (New)

In addition to the default run configurations for available project types, you can create custom run configurations for your project. The run configuration defines how your project is executed and includes settings for preview mode, whether to use mock data, application resource mapping, and application URL parameters.

For more information, see [Create Run Projects \[page 435\]](#).

Code Snippets in Extension Point Fragments (New)

Code snippets are provided in the generated fragment with proposed implementation, once an extension point implementation is added.

Fetching Mock Data Resources from Other Projects (New)

When configuring mock data usage settings for your project, you can enter a path to the service metadata xml file residing either in your current project or in a different project in your workspace.

For more information, see [Configure Mock Data Usage \[page 199\]](#).

Accessing SAP Hana Cloud Platform Cockpit from the Menu Bar (New)

You can now access SAP Hana Cloud Platform cockpit from the [Tools](#) menu.

Registering to SAP Fiori Launchpad (Changed)

The following features have been added to the [Register to Fiori Launchpad](#) wizard.

- Dynamic tile support, including live tile preview.
- Option to add intent to the application.

For more information, see [Register Applications to SAP Fiori Launchpad \[page 454\]](#).

Code Completion (Changed)

JavaScript code completion and API reference view now support multiple SAPUI5 library versions based on the project configuration.

Running Applications (Changed)

There are new options in the *Run* menu that allow you to run applications from the workspace. You can choose the run configuration that you want to use from the *Run As* submenu or from the *Recently Used* submenu.

When you click the *Run* button, the application runs using the last-used run configuration that matches the project type.

For more information, see [Run Applications from the Workspace \[page 433\]](#).

Location of the Plugin Management Pages (Changed)

The optional and available plugin management pages are now accessible from the SAP Web IDE Preferences pane.

Layout Editor (Changed, Experimental)

The following improvements are available in the layout editor:

- SAPUI5 controls *Date Picker* and *Panel* are now supported.
- You can now open views without the *Page* control in the layout editor.
- In the canvas, the graphical display of the XML view uses the full browser window. This means that you only have to scroll if the browser window is too small.

i Note

The layout editor and all features that come with it are experimental and are therefore available only in the trial landscape.

Hybrid Application Toolkit Plugin (Changed)

Hybrid Application Toolkit, version 1.2.0 of the plugin for Web IDE 1.8, contains the following new features:

- Hybrid Application Toolkit uses a UI-based installation assistant to simplify the installation and setup of the development environment. This assistant replaces the command line script used in previous releases.
- You can preview hybrid apps in the web browser:

- Calls to native APIs for the Cordova plugins do not break at runtime. Instead, they fail gracefully.
- Calls to native APIs for the Cordova Camera plugin and the Cordova Device plugin are fully supported by making use of your machine's local resources.
- You can use a generated QR code to load the application URL to preview hybrid apps on the Companion App. This feature allows you to use the Companion App without having the device physically connected to the development system.

For more information, see [Plugins and Add-Ons \[page 472\]](#).

Fact Sheet Editor Plugin (Changed)

The fact sheet editor plugin is enhanced as follows:

- You can now create modification free fact sheets that allow users to save changes made in the plugin as an .xsl file. When this file is uploaded to the backend system, it is applied to the original annotation file. As a result, the fact sheet contains both SAP changes coming from the annotation file and user-specific changes provided by the .xsl file, if these changes are compatible.
- The *Modification Free Fact Sheet* project contains the log of user changes. This log is maintained using the domain-specific language (DSL) in the .xsl file.
- Internationalization of new and changed labels is now supported by means of Online Text Repository (OTR) records. When changing a label for a field, you can assign an existing OTR alias/label record or add a new OTR alias/label to be associated with this field. New OTR aliases must be added manually to the back-end system.

For more information, see https://help.hana.ondemand.com/webide_factsheet/frameset.htm.

Testing HTML5 Applications Together with Reuse Applications (Changed)

You can test HTML5 applications that reuse applications together with these applications as a preview from the workspace. The reuse applications and your projects no longer need to have the same name.

For more information, see [Running Applications in Development Mode \[page 432\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)

[April 2017 \[page 10\]](#)

[March 2017 \[page 12\]](#)

[February 2017 \[page 14\]](#)

[January 2017 \[page 18\]](#)

[December 2016 \[page 22\]](#)

November 2016 [page 31]
October 2016 [page 36]
September 2016 [page 40]
August 2016 [page 47]
July 2016 [page 51]
June 2016 [page 55]
May 2016 [page 60]
April 2016 [page 64]
March 2016 [page 73]
February 2016 [page 79]
January 2016 [page 86]
SAP Web IDE 1.19 [page 90]
SAP Web IDE 1.18 [page 96]
SAP Web IDE 1.17 [page 99]
SAP Web IDE 1.16 [page 103]
SAP Web IDE 1.15 [page 106]
SAP Web IDE 1.14 [page 109]
SAP Web IDE 1.13 [page 111]
SAP Web IDE 1.12 [page 114]
SAP Web IDE 1.11 [page 118]
SAP Web IDE 1.10 [page 121]
SAP Web IDE 1.9 [page 124]
SAP Web IDE 1.7 [page 131]
SAP Web IDE 1.6 [page 136]
SAP Web IDE 1.5 [page 138]
SAP Web IDE 1.4 [page 140]
SAP Web IDE 1.3 (Trial Only) [page 145]
SAP Web IDE 1.2 [page 146]

2.1.30 SAP Web IDE 1.7

Learn about the new and changed features that are available for SAP Web IDE 1.7.

SAP Web IDE Optional Plugins and Add-Ons (New)

SAP Web IDE, hybrid app toolkit add-on has been added to the list of optional plugins and add-ons that you can enable.

For more information, see [Plugins and Add-Ons \[page 472\]](#).

Registering Applications to SAP Fiori Launchpad (New)

You can register your SAP HANA Cloud Platform deployed application to SAP Fiori launchpad directly from SAP Web IDE.

For more information, see [Register Applications to SAP Fiori Launchpad \[page 454\]](#).

Updates to Templates (Changed)

- The Fiori Starter Application and the SAPUI5 Master-Detail Application templates have merged. The new template is now called Fiori Master-Detail.

Adding Custom Content to the Finish Step (New)

As a developer, you can now add content to the content area in the *Confirm* step of the *New Project* wizard.

Extensibility Pane UX Enhancements (Changed)

The following UX enhancements were made to the extensibility pane:

- Changes to the outline tree style
- Grouping of elements by views, fragments, and controllers
- Showing extension points in the application when switching to Extensibility Mode

For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).

Extensibility Fragment Support (New)

You can extend existing SAP Fiori applications by hiding UI controls and using extension points from fragments.

For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).

Updating Existing Applications in the SAPUI5 ABAP Repository (Changed)

When updating an existing application that resides in the SAPUI5 ABAP repository, the application is not deleted and created. The deployment mechanism updates, creates, and deletes only the relevant files.

For more information, see [Update Existing Applications Residing in the SAPUI5 ABAP Repository \[page 450\]](#).

Source Control UX Enhancements (Updated)

The UX for working with Git was updated. The following were changed:

- Git History pane
- Search capabilities
- Deletion of multiple branches

For more information, see [Using Source Control \(Git\) \[page 402\]](#).

Added Git Functionality (New)

The following functionality was added:

- Commit and Push - you can now perform the commit and push actions by choosing only one button. For more information, see [Push Changes \[page 422\]](#).
- Push tag - all tags created within an open repository will be pushed. For more information, see [Push Changes \[page 422\]](#).

Adding Custom Mock Requests (New)

You can add a file containing custom mock requests that will be used when running applications with mock data.

For more information, see [Add Custom Mock Requests \[page 445\]](#).

Code Editor Custom Menus (New)

You can use context menus when working with code and code comments in the code editor.

For more information, see [Working in the Code Editor \[page 214\]](#).

Font Size in the Code Editor (New)

You can choose the font size for text in the code editor.

For more information, see [Configure the Code Editor \[page 217\]](#).

Using Outlines for JavaScript Files (New)

You can view an outline of JavaScript files in the Outline pane. This feature helps you to understand the main entities in the JavaScript file and the flow of the application.

For more information, see [Using the Outline Pane for JavaScript Files \[page 329\]](#).

Testing HTML5 Applications Together with Reuse Applications (New)

You can test HTML5 applications that reuse applications together with these applications as a preview from the workspace. For this to work, the reuse applications have to be contained in your workspace and the projects must have the same name as the reuse applications.

For more information, see [Running Applications for In-Development Testing \[page 432\]](#).

OData Model Editor (Changed)

Importing Files into OData Model Editor

The `Import OData Model` feature allows you to import `XML` and `EDMX` files that were defined in other editors and extend them as required. While importing the files into the OData Model editor, ensure that the naming conventions for the imported file are met. When importing the files, OData Model editor validates the file and displays the validation errors.

Exporting files from OData Model Editor

The `Export OData Model as XML` feature allows you to export individual files from the OData Model editor as `XML` files.

Selecting Theme for Code Editor

The OData Model editor allows you to choose a theme for your code editor. SAP Web IDE provides default themes for the code editor that you can use to customize the appearance of your files.

Beautifying OData Model Files

The `Beautify` option formats an `XML` string/file with your desired indentation level. The formatting rules are not configurable, but it uses a per-element indentation pattern, giving the best readability.

Validation in OData Model Editor

To enhance the user's experience, the model and schema validations are combined, and validations happen on the fly.

Output of Validations

All the `EDMX/XML` files that are created are subjected to a round of validation. In event of an error during the validation, the particular line number is colored red. Also, the error message is displayed in the console below the code editor.

For more information about new and changed OData Model Editor features, see [Developing an OData Model](#).

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.31 SAP Web IDE 1.6

Learn about the new and changed features that are available for SAP Web IDE 1.6.

SAP Web IDE Optional Plugins and Add-Ons (New)

- The Visualization Extension plugin (Vizpacker) is used in SAP Web IDE to create chart extension packages that can be used within SAP Lumira and other products.
- SAP Web IDE, hybrid app toolkit add-on, enables developers to create hybrid apps (also known as Kapsel apps) using Apache Cordova and the SAP Mobile Platform SDK.
- The SAP HANA Cloud Portal plugin is no longer in experimental mode and can be used in both the trial and the productive landscapes.

For more information, see [Plugins and Add-Ons \[page 472\]](#).

Creating a New Project from a Sample Application (New)

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

For more information, see [Create Projects from Sample Applications \[page 188\]](#).

Running Applications with Predefined URL Parameters (New)

You can preview and test your application using predefined URL parameters.

Support of Fact Sheets Based on Annotation Files of Version 4.0 (New)

You can create new fact sheets and edit existing fact sheets that are based on the annotation files of version 4.0 in the fact sheet editor plugin.

For more information, see [Using the Fact Sheet Editor](#).

Editing Field Navigation in Fact Sheets Based on Annotation Files of Version 4.0 (New)

You can add, remove, or change navigation of the fields in the fact sheets that are based on the annotation files of version 4.0 in the fact sheet editor plugin.

For more information, see [Using the Fact Sheet Editor](#).

Comparing Code in Git (Changed)

The compare editor has been improved with the following new features:

- *Next* and *Previous* buttons have been added to help you navigate between the highlighted differences in your modified code and the original code. This makes it easier to compare and view the recent changes that you made and to resolve conflicts that may have occurred following a rebase or merge operation.
- A *Copy from right to left* button allows you to move code from the original version to your modified version.

For more information, see [Compare Code Versions \[page 417\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)

[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.32 SAP Web IDE 1.5

Learn about the new and changed features that are available for SAP Web IDE 1.5.

SAP HANA Cloud Portal Plugin (Experimental, New)

This plugin enables you to create different types of OpenSocial widgets for use in Cloud Portal sites.

For more information, see [Plugins and Add-Ons \[page 472\]](#).

Adding and Activating a Version when Deploying to SAP HANA Cloud Platform (New)

You can add and activate a version when deploying to SAP HANA Cloud Platform. The application is started automatically.

For more information, see [Deploy Applications to SAP Cloud Platform \[page 451\]](#).

Automatically Subscribing to SAP Web IDE (New)

When setting up SAP Web IDE, you no longer have to subscribe to it in the cloud. Instead, you use the existing [webide](#) subscription in the SAP HANA Cloud Platform cockpit.

For more information, see [Opening SAP Web IDE \[page 156\]](#).

Searching for and Opening Documents (New)

You can search for documents and open them in the workspace by using the [Open Resource](#) option (keyboard shortcut [`CTRL` + `SHIFT` + `R`](#)).

For more information, see [Working with Files and Folders \[page 176\]](#).

Git History Pane (Changed)

From the Git History pane, you can now take the code from a specific commit and create a new branch based on it.

For more information, see [Git History \[page 426\]](#).

Running Applications with Mock Data (Changed)

An option to run applications with mock data using a URL parameter was added.

For more information, see [Run Applications with Mock Data \[page 443\]](#).

Reloading without Personalization (Changed)

By adding the parameters `settings=ignore` or `settings=delete` to the URL, you can reload SAP Web IDE without restoring the state of the workspace and the editors (skipping or deleting the persistence information).

For more information, see [Optional - SAP Web IDE User Settings \[page 163\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)

SAP Web IDE 1.19 [page 90]
SAP Web IDE 1.18 [page 96]
SAP Web IDE 1.17 [page 99]
SAP Web IDE 1.16 [page 103]
SAP Web IDE 1.15 [page 106]
SAP Web IDE 1.14 [page 109]
SAP Web IDE 1.13 [page 111]
SAP Web IDE 1.12 [page 114]
SAP Web IDE 1.11 [page 118]
SAP Web IDE 1.10 [page 121]
SAP Web IDE 1.9 [page 124]
SAP Web IDE 1.8 [page 127]
SAP Web IDE 1.7 [page 131]
SAP Web IDE 1.6 [page 136]
SAP Web IDE 1.4 [page 140]
SAP Web IDE 1.3 (Trial Only) [page 145]
SAP Web IDE 1.2 [page 146]

2.1.33 SAP Web IDE 1.4

Learn about the new and changed features that are available for SAP Web IDE 1.4.

Extensibility Pane Enhancements (New, Changed)

These features have been added to this version of the extensibility pane:

- Enhanced capabilities of the extensibility pane. For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).
- From the extensibility pane you can view the original application's views and controllers in read-only mode. For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).

Extending Applications (New)

You can now extend an existing SAP Fiori application that resides on SAP HANA Cloud Platform without importing it to SAP Web IDE. For more information, see [Extend Applications that Reside on SAP Cloud Platform \[page 459\]](#).

You can import the original application to your workspace while creating the extension project. For more information, see [Extend Applications that Reside in the SAPUI5 ABAP Repository \[page 458\]](#).

Template Enhancements (New, Changed)

The following enhancements have been made to templates:

- New Master-Detail SAPUI5 Mobile Starter-Kit template to develop a master-detail application that is connected to an OData service
- Improved the SAPUI5 application project. One template now allows you to select various view types
- Added new extension points to the Fiori Starter Application template

Extending Applications that Reside in SAP HANA Cloud Platform (New)

You can extend existing SAP Fiori applications residing on SAP HANA Cloud Platform without importing them to SAP Web IDE. For more information, see [Extend Applications that Reside on SAP Cloud Platform \[page 459\]](#).

Editing Mock Data (New)

You can model the service data that you want to use as mock data in your application. For more information, see [Edit Mock Data \[page 444\]](#).

Automatically Saving Changes (New)

You can automatically save the changes in all currently open documents at preset intervals. For more information, see [Setting User Preferences \[page 163\]](#).

Changing Fact Sheet Header (New)

You can change the fields in the header section of the fact sheets.

For more information, see [Using the Fact Sheet Editor](#).

Editing Fact Sheet Key Values (New)

You can do the following:

- Change the number of fractional digits to display for the key values of the decimal type
- Add, remove, and reorder key values

For more information, see [Using the Fact Sheet Editor](#).

Editing Fact Sheet Labels (New)

You can edit labels for facet titles and fields on the overview page.

For more information, see [Using the Fact Sheet Editor](#).

Error Logging and Troubleshooting in Fact Sheet Editor (Changed)

Error logging and troubleshooting were enhanced in the fact sheet editor plugin.

For more information, see [Using the Fact Sheet Editor](#).

Logging Out and Persistence for Personalization (New)

To end your session and log out, you can now choose *Logout* in the top-right corner of SAP Web IDE.

If you start SAP Web IDE after you have logged out, the system restores the latest state of the workspace and all the editors that were open.

Previewing Images (New)

You can now display a preview of a binary image file in the editor, in the following ways:

- Double-click the image file in the workspace
- Select the image file in the workspace and choose  

Locally Installing SAP Web IDE (New)

SAP Web IDE is now also available as an evaluation package as an SCN download. The local installation version has restricted functionality (for example, there is no cloud-specific functionality) and cannot be used productively.

OData Model Editor (New)

Validating Models

You can now validate a model using **model validation**. Model validation is the logical/semantic validation of the user-defined model. This feature ensures that the OData service that is developed using the OData Model editor interacts well with any OData client.

Opening XML Files in the OData Model Editor

You can now open XML files in the OData Model editor and use the code assist, autocomplete, and validation features.

For more information, see [Developing an OData Model](#).

Product Name (Changed)

The product name was changed to SAP Web IDE. The old name (SAP River RDE) is no longer used.

As part of this name change, the following property names were adjusted:

Old Name	New Name
RDEUsage	WebIDEUsage
RDEEnabled	WebIDEEEnabled
RDESSystem	WebIDESystem
RDEPermission	WebIDEPPermission

For more information on how to use these properties, see [Connect to ABAP Systems \[page 158\]](#).

Code Editor (Changed)

These new location features were added to this version of the code editor:

- Goto feature - allows location of identified function definitions
- Highlight all instances feature - allows finding all occurrences of a selected variable/function in an open file

For more information, see [Locate Objects in Code \[page 219\]](#).

Keyboard Shortcuts (Changed)

Several new keyboard shortcuts were added. For more information, see [Keyboard Shortcuts \[page 166\]](#).

Layout Editor (Changed, Experimental)

The functionality of the layout editor and its controls and features have been significantly stabilized.

i Note

The layout editor and all features that come with it are experimental and are available only in the trial landscape.

For more information, see [Layout Editor \[page 332\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)
[April 2017 \[page 10\]](#)
[March 2017 \[page 12\]](#)
[February 2017 \[page 14\]](#)
[January 2017 \[page 18\]](#)
[December 2016 \[page 22\]](#)
[November 2016 \[page 31\]](#)
[October 2016 \[page 36\]](#)
[September 2016 \[page 40\]](#)
[August 2016 \[page 47\]](#)
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.34 SAP Web IDE 1.3 (Trial Only)

Learn about the new and changed features that are available for SAP Web IDE 1.3 for trial use on hanatrial.ondemand.com.

Layout Editor (New, Experimental)

- 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.
For more information, see [Layout Editor \[page 332\]](#).
- Add or remove controls to your XML view or change its appearance by using drag and drop functionality.
For more information, see [Add Controls from the Controls Tab \[page 341\]](#).
- Bind properties to OData metadata.

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

- [May 2017 \[page 9\]](#)
- [April 2017 \[page 10\]](#)
- [March 2017 \[page 12\]](#)
- [February 2017 \[page 14\]](#)
- [January 2017 \[page 18\]](#)
- [December 2016 \[page 22\]](#)
- [November 2016 \[page 31\]](#)
- [October 2016 \[page 36\]](#)
- [September 2016 \[page 40\]](#)
- [August 2016 \[page 47\]](#)
- [July 2016 \[page 51\]](#)
- [June 2016 \[page 55\]](#)
- [May 2016 \[page 60\]](#)
- [April 2016 \[page 64\]](#)
- [March 2016 \[page 73\]](#)
- [February 2016 \[page 79\]](#)
- [January 2016 \[page 86\]](#)
- [SAP Web IDE 1.19 \[page 90\]](#)
- [SAP Web IDE 1.18 \[page 96\]](#)
- [SAP Web IDE 1.17 \[page 99\]](#)
- [SAP Web IDE 1.16 \[page 103\]](#)
- [SAP Web IDE 1.15 \[page 106\]](#)
- [SAP Web IDE 1.14 \[page 109\]](#)
- [SAP Web IDE 1.13 \[page 111\]](#)
- [SAP Web IDE 1.12 \[page 114\]](#)
- [SAP Web IDE 1.11 \[page 118\]](#)

[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.2 \[page 146\]](#)

2.1.35 SAP Web IDE 1.2

Learn about the new and changed features that are available for SAP Web IDE 1.2.

For an overview of new and changed features on patch level, see <http://scn.sap.com/docs/DOC-56628>.

SAP Web IDE for Productive Use

SAP Web IDE is now for the first time available for productive use on [hana.ondemand.com/cockpit](#) (Europe), [us1.hana.ondemand.com](#) (United States) and [ap1.hana.ondemand.com](#) (Asia-Pacific).

The following list gives an overview of the major features that SAP Web IDE 1.2 provides:

- Cloud-based Web IDE to support the end-to-end life cycle for SAPUI5-based applications
- Template-based project and resource creation
For more information, see [Create Projects from Templates \[page 181\]](#).
- Feature-rich source code editors with code completion for JavaScript and XML
- Mock data support for back-end independent application tests
For more information, see [Run Applications with Mock Data \[page 443\]](#).
- Instantly preview application for any device screen size - desktop, tablet, or smart phone
- Cloud-based Git as code repository to enable team collaboration
For more information, see [Using Source Control \(Git\) \[page 402\]](#).
- Support for app deployment to ABAP, SAP HANA Cloud Platform, and SAP Mobile Platform (via Kapsel)
- Tool support to extend SAP delivered applications without modifications
For more information, see [Extending SAPUI5 Applications \[page 457\]](#).
- Extensible and modular tool platform architecture with plugins for various scenarios

SAP Web IDE for Trial Use

Compared to version 1.0, SAP Web IDE 1.2 includes the following new or changed features for trial use on [hanatrial.ondemand.com](#):

Code Editor Customizations (Changed)

Core functionality of the code editor has been enhanced with new customization capabilities:

- Validate and resolve issues with code by performing static code checks and correcting issues inline.
- Improve code readability by applying formatting changes.

For more information, see [Customizing SAP Web IDE \[page 193\]](#).

Project Creation and Support (Changed)

Project creation and maintenance has been enhanced:

- User settings and project settings have been centralized and extended.
- Search and replace has been improved. You can now perform serial and global search and replace actions across files in a project, as well as within single files. For more information, see [Searching and Replacing in SAP Web IDE \[page 170\]](#).

Working with SAP Web IDE Git (Changed)

The following Git features can now be accessed from the SAP Web IDE Git pane:

- Setting Git credentials
- Comparing files before committing changes
- Rebase Interactive pane
- Git history
- Cherry-picking commits
- Tagging commits
- Reverting commits

For more information, see [Using Source Control \(Git\) \[page 402\]](#).

Extensibility Features (Changed)

The extensibility functionality has been enhanced with new capabilities:

- Extend controller functionality – you can now copy an original controller (the code is formatted and commented). For more information, see [Extend Controllers \[page 464\]](#).
- Extensibility pane enhancements:
 - You can now open the extensibility pane using mock data.
 - You can now remove extensions.
 - When the extensibility mode is selected, the application is locked for navigation and you can right-click on elements to select extensibility options.
 - Outline tree enhancements. You can now see which element was extended and by which other element.

For more information, see [Extend UI Elements Using the Extensibility Pane \[page 461\]](#).

Importing Applications from the SAPUI5 ABAP Repository (New)

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

For more information, see [Import Applications from the SAPUI5 ABAP Repository \[page 192\]](#).

Deploying Applications to the SAPUI5 ABAP Repository (New)

You can deploy an existing application from the SAP Web IDE workspace to the SAPUI5 ABAP repository.

For more information, see [Deploy Applications to the SAPUI5 ABAP Repository \[page 447\]](#).

Plugin Development Story (Changed)

- You can use a template to create a plugin project.
- You can use a template to create new templates.

- You can use the auto-complete option in template files (*.tpl).
- You can debug and test plugin projects within SAP Web IDE.

Creating Fact Sheets (New)

You can create fact sheets from scratch using the [New Fact Sheet Application](#) project.

For more information, see [Using the Fact Sheet Editor](#).

Editing Fact Sheets (Changed)

The fact sheet editor plugin has been enhanced to perform the following actions when editing fact sheets:

- Add new fields provided by OData services or remove the existing fields from the facets
- Add and reorder other facets
- Hide, unhide, or remove other facets
- Add, remove, or reorder fields on the detail pages for other facets

For more information, see [Using the Fact Sheet Editor](#).

Connecting Remote Systems (Changed)

You can now establish a connection to an ABAP back-end system by creating one destination for multi-usage.

For more information, see [Connect to ABAP Systems \[page 158\]](#).

SAP Jam Integration (New, Experimental)

Collaborate with multiple developers by connecting to SAP Jam and applying inline comments to highlighted code. You can also collaborate with customers by collecting feedback against supported simulators to preview the application design and modify the application based on that feedback.

Custom Code Checking (New)

The JavaScript editor in SAP Web IDE supports JavaScript linting for all opened *.js, *.xsjs, and *.xsjslib files. Once code checking is triggered, the detected inline code issues are displayed as annotations within the editor, according to the customer configuration that you make. You can add custom rules if needed.

For more information, see [Checking Code with Customized JavaScript Linting \[page 239\]](#).

Parent topic: [What's New in Previous Releases of SAP Web IDE \[page 7\]](#)

Related Information

[May 2017 \[page 9\]](#)

[April 2017 \[page 10\]](#)

[March 2017 \[page 12\]](#)

[February 2017 \[page 14\]](#)

[January 2017 \[page 18\]](#)

[December 2016 \[page 22\]](#)

[November 2016 \[page 31\]](#)

[October 2016 \[page 36\]](#)

[September 2016 \[page 40\]](#)

[August 2016 \[page 47\]](#)

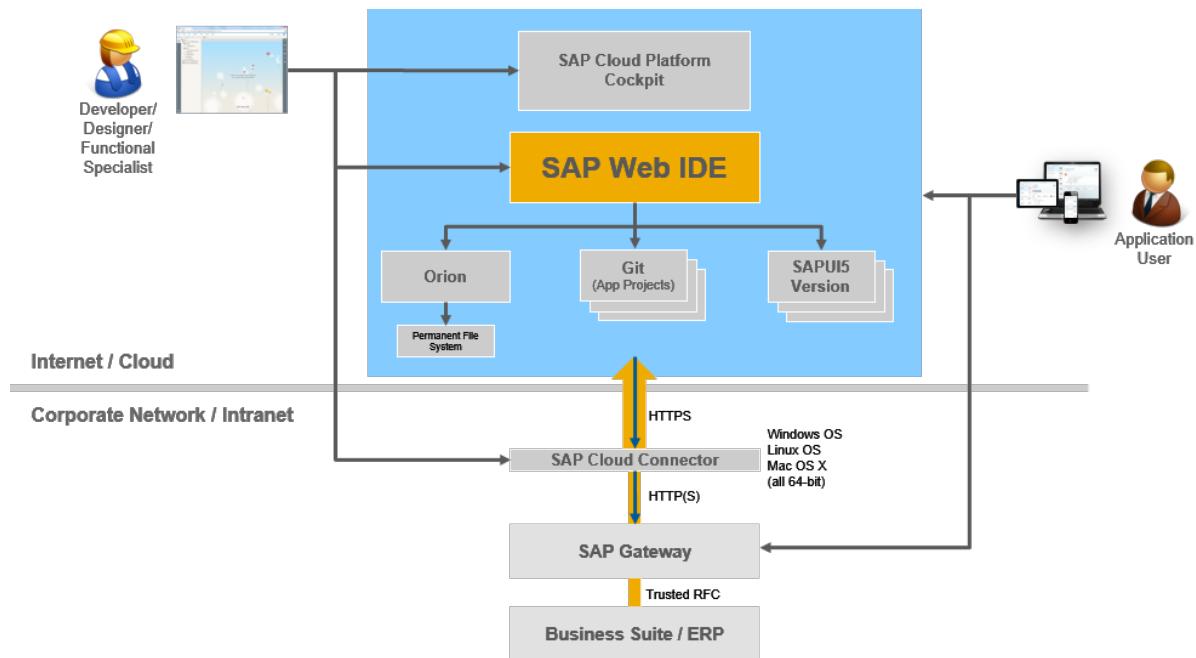
[July 2016 \[page 51\]](#)
[June 2016 \[page 55\]](#)
[May 2016 \[page 60\]](#)
[April 2016 \[page 64\]](#)
[March 2016 \[page 73\]](#)
[February 2016 \[page 79\]](#)
[January 2016 \[page 86\]](#)
[SAP Web IDE 1.19 \[page 90\]](#)
[SAP Web IDE 1.18 \[page 96\]](#)
[SAP Web IDE 1.17 \[page 99\]](#)
[SAP Web IDE 1.16 \[page 103\]](#)
[SAP Web IDE 1.15 \[page 106\]](#)
[SAP Web IDE 1.14 \[page 109\]](#)
[SAP Web IDE 1.13 \[page 111\]](#)
[SAP Web IDE 1.12 \[page 114\]](#)
[SAP Web IDE 1.11 \[page 118\]](#)
[SAP Web IDE 1.10 \[page 121\]](#)
[SAP Web IDE 1.9 \[page 124\]](#)
[SAP Web IDE 1.8 \[page 127\]](#)
[SAP Web IDE 1.7 \[page 131\]](#)
[SAP Web IDE 1.6 \[page 136\]](#)
[SAP Web IDE 1.5 \[page 138\]](#)
[SAP Web IDE 1.4 \[page 140\]](#)
[SAP Web IDE 1.3 \(Trial Only\) \[page 145\]](#)

3 Overview

SAP Web IDE is a web-based tool that enables you to create and extend end user applications for browser and mobile devices. It simplifies the end-to-end application lifecycle: development, packaging, deployment, and customer extensions for SAPUI5 and SAP Fiori applications, and allows developers to collaborate with business experts and designers to fulfill end user requirements and expectations more effectively.

Architecture

The following diagram provides an architectural overview of the SAP Web IDE environment:



Component	Description
SAP Cloud Platform	SAP Cloud Platform 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.
SAP Cloud Platform cockpit	Central point for managing all activities associated with your SAP Cloud Platform account and for accessing key information about your applications. For more information, see SAP Cloud Platform cockpit .
SAP Web IDE application	Integrated development environment used to create or extend SAP UI5 or SAP Fiori applications.

Component	Description
Orion	<p>Server that provides a number of back-end services that SAP Web IDE relies on.</p> <p>For example:</p> <ul style="list-style-type: none"> • Provides private workspaces per user or per account • Enables working with projects that are stored in a workspace • Enables working with files and folders that are stored in a workspace • Enables working with local Git repositories and exchanging Git commits with a central Git server
Git	Revision control and source code management system.
SAPUI5	<p>User interface technology that is used to build and adapt client applications.</p> <p>For more information, see UI development toolkit for HTML5 - Demo Kit</p>
SAP Cloud Platform connector	<p>Allows SAP Web IDE and SAP Cloud Platform to connect to an on-premise system securely and with minimal configuration effort.</p> <p>For more information, see SAP Cloud Platform connector.</p>
SAP Gateway	Provides a simple way to connect SAP Web IDE to an external SAP system with access to OData functionality.

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.

[Features and Benefits \[page 151\]](#)

SAP Web IDE offers many key features that can be used by end users, administrators, and developers.

3.1 Features and Benefits

SAP Web IDE offers many key features that can be used by end users, administrators, and developers.

The following table provides the SAP Web IDE Feature Scope Description.

Key Feature	Use
Zero installation and maintenance	Available as a cloud service on SAP Cloud Platform, it only requires a browser.
Code Authoring Toolset	<p>SAP Web IDE supports free-style coding by providing robust editors with capabilities of code completion (XML, property files, JavaScript 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 about problems in the projects in your workspace.</p> <p>Our code editors also beautify file formatting for JavaScript, JSON, XML, and CSS files using the context menu.</p>
Testing and debug	<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 via QR-code.</p> <p>Mock data can be automatically generated or manually created to run the application independently for front-end/back-end development de-coupling, testing and demoing purposes.</p>
Mulit-device support	The responsive patterns and controls of SAPUI5 and the adaptive design of apps allow SAP Web IDE to run on desktops, tablets, smartphones, and hybrid devices. 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 SAPUI5 applications applying SAP Fiori UX guidelines using wizards and templates.
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 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. • UI Adaptation: The UI Adaptation editor provides an intuitive user interface to make changes to SAP Fiori element applications.
Storyboard	
Extensibility Capabilities	Extend SAP Fiori applications through a visual extensibility editor. SAP Fiori, cloud edition leverages SAP Web IDE to extend and customize SAP Fiori apps.

[Assisted Development \[page 153\]](#)

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

[Collaborative Development \[page 154\]](#)

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.

[User-Centric Customizability and Session Persistence \[page 155\]](#)

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.

Parent topic: [Overview \[page 150\]](#)

3.1.1 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.

Parent topic: [Features and Benefits \[page 151\]](#)

Related Information

[Collaborative Development \[page 154\]](#)

[User-Centric Customizability and Session Persistence \[page 155\]](#)

3.1.1.1 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.

3.1.1.2 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.

3.1.2 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.

Parent topic: [Features and Benefits \[page 151\]](#)

Related Information

[Assisted Development \[page 153\]](#)

[User-Centric Customizability and Session Persistence \[page 155\]](#)

3.1.2.1 Git Client Plugin

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

Related Information

[Set Up Git \[page 406\]](#)

[Using Source Control \(Git\) \[page 402\]](#)

3.1.2.2 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.

3.1.3 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.

Parent topic: [Features and Benefits \[page 151\]](#)

Related Information

[Assisted Development \[page 153\]](#)

[Collaborative Development \[page 154\]](#)

4 Getting Started

Set up SAP Web IDE by subscribing to it in the cloud, creating destinations, configuring Git, and connecting remote systems.

Prerequisites

- You must first sign up for an SAP Cloud Platform developer account. For more information, see [Getting an Account](#).
- In the SAP Cloud Platform cockpit, you can add users who develop and maintain HTML5 applications as account members.

[Open SAP Web IDE \[page 156\]](#)

You open SAP Web IDE in a web browser.

[Connect to ABAP Systems \[page 158\]](#)

Establish a connection to an ABAP back-end system by creating one destination for multi-usage.

4.1 Open SAP Web IDE

You open SAP Web IDE in a web browser.

Prerequisites

To restrict access to SAP Web IDE, make sure that you have assigned your roles to the `WebIDEPermission` permission. For more information, see [Assign Users Permission in SAP Web IDE \[page 495\]](#).

i Note

If no role is assigned, only account members with developer or administrator permissions have access to the protected resource. For more information, see [Managing Roles and Permissions](#).

Procedure

1. Log on with a user (who is an account member) to the SAP Cloud Platform cockpit, using the developer account at one of the following URLs:

Productive Use	Trial Use
<ul style="list-style-type: none">○ <code>https://account.hana.ondemand.com/cockpit</code> (Europe)○ <code>https://account.us1.hana.ondemand.com</code> (United States)○ <code>https://account.ap1.hana.ondemand.com</code> (Asia-Pacific)	<code>https://account.hanatrial.ondemand.com/cockpit</code>

2. Select the *Neo* environment.
3. In the navigation pane, choose *Services*.
4. On the *Services* pane, select *SAP Web IDE*.
5. Click *Open SAP Web IDE*.

The following browsers are supported:

- Microsoft Internet Explorer (version 11)
- Microsoft Edge
- Mozilla Firefox
- Google Chrome
- Safari

i Note

Opening multiple instances of SAP Web IDE in parallel may cause issues. For more information, see [Known Issues \[page 510\]](#).

Task overview: [Getting Started \[page 156\]](#)

Related Information

[Connect to ABAP Systems \[page 158\]](#)

4.2 Connect to ABAP Systems

Establish a connection to an ABAP back-end system by creating one destination for multi-use.

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 Cloud Connector and defined a virtual host mapping for the system.
See [Cloud Connector](#).
 - In the *Access Control* tab page of SAP Cloud Platform 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`
for extensibility 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-use 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 Cloud Platform cockpit:

i Note

You can access the SAP Cloud Platform cockpit from SAP Web IDE by selecting **Tools** **SAP Cloud Platform Cockpit**.

Option	Description
productive use	<code>https://account.hana.ondemand.com/cockpit</code> (Europe) <code>https://account.us1.hana.ondemand.com</code> (United States) <code>https://account.ap1.hana.ondemand.com</code> (Asia-Pacific)
trial use	<code>https://account.hanatrial.ondemand.com/cockpit</code>

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 SAP Cloud Platform connector)

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.

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 No Authentication or Basic Authentication depending on the authentication you need for the connection. For more information, see Creating HTTP Destinations .

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">○ odata_abap: for the OData functionality of Gateway (corresponds to URL path /sap/opu/odata)○ odata_gen:

Key	Value
	<p>for generic OData functionality (service URL must be provided manually in the New Project wizard)</p> <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 Cloud Platform ◦ <code>odata_hci</code>: for accessing OData services in a HANA Cloud Integration system on Cloud Platform. 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 Cloud Platform ◦ <code>smart_business_odata</code>: for smart business OData services ◦ <code>smart_business_gen</code>: for smart business annotation generation services

i Note

- In order to import an application from your ABAP system and execute it, the following `WebIDEUsage` property values are needed: `dev_abap`, `odata_abap`, `ui5_execute_abap`.
- When you enter multiple usages for a destination, separate them by commas without spaces (for example, `odata_abap,ui5_execute_abap`).

<code>WebIDEEEnabled</code>	<code>true</code>
<code>WebIDESystem</code>	<code><SAP system_ID></code>
<code>sap-client</code>	<code><SAP client from the ABAP system></code>

Key	Value
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 <p>i Note</p> <p>When specifying this value, the <i>URL</i> 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.</p>

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

Task overview: [Getting Started \[page 156\]](#)

Related Information

[Open SAP Web IDE \[page 156\]](#)

[SAP Cloud Platform connector](#)

[Creating HTTP Destinations](#)

4.2.1 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_INFRA` and `UI5_731`) are installed.

Related Information

[Connect to ABAP Systems \[page 158\]](#)

[Connect Remote Systems in SAP Web IDE Personal Edition \[page 486\]](#)

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, which 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 240]
Code completion settings to enable inline code completion and comment completion.	Configure Code Completion [page 223]
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 217]
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 406]
Customize SAP Web IDE keyboard shortcuts.	Customize Keyboard Shortcuts [page 169]
Enable plugins to create and develop applications.	Enable Plugins [page 474]

Preferences

More Information

Delete or ignore workspace persistence.

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.

[Navigate SAP Web IDE \[page 165\]](#)

In SAP Web IDE, you can choose from different types of menus and toolbars to perform your actions.

[Keyboard Shortcuts \[page 166\]](#)

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

[Workspace Search Options \[page 170\]](#)

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.

[Work with Files and Folders \[page 176\]](#)

Use keyboard shortcuts and context menus to work with files and folders.

[Resize Panes \[page 178\]](#)

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

6.1 Navigate SAP Web IDE

In SAP Web IDE, you can choose from different types of menus and toolbars to perform your actions.

SAP Web IDE UI is arranged in the following areas:

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

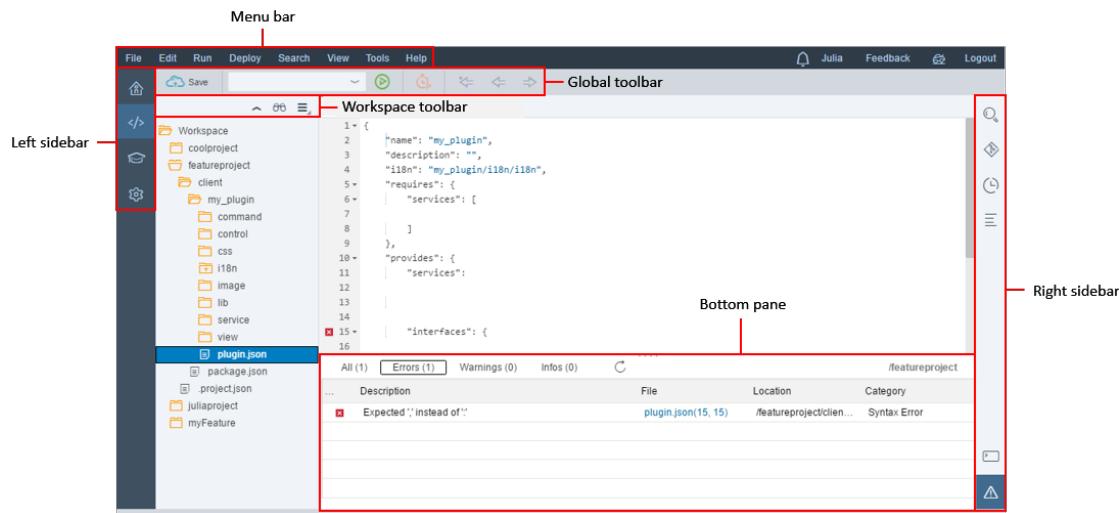
Global toolbar - Depending on the item that is activated in the workspace, you can choose from the buttons in the global toolbar (icons of actions that are not applicable are grayed out).

Left sidebar - Use the buttons to switch between the Welcome page, development workspace, 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, Problems View bottom pane, and so on).

Workspace toolbar - Use the buttons to collapse all projects with one click, search for any resource and quickly open it from the [Open Resource](#) dialog, and access additional options related to the workspace.

Bottom pane - The bottom pane displays the SAP Web IDE console and the Problems view. You can switch between these two using the bottom buttons in the right sidebar.



[Parent topic: SAP Web IDE Basics \[page 165\]](#)

Related Information

[Keyboard Shortcuts \[page 166\]](#)

[Workspace Search Options \[page 170\]](#)

[Work with Files and Folders \[page 176\]](#)

[Resize Panes \[page 178\]](#)

6.2 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 \[page 169\]](#).

Action	Microsoft Windows Keyboard Shortcut	Mac OS Keyboard Shortcut
New file	Ctrl + Alt + N	Command + Alt + N
New folder	Ctrl + Alt + Shift + N	Command + Alt + Shift + N

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

Action	Microsoft Windows Keyboard Shortcut	Mac OS Keyboard Shortcut
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]
Refactor	[Alt] + [J]	[Alt] + [J]
Open i18n	[Alt] + [I]	[Alt] + [I]

Parent topic: SAP Web IDE Basics [page 165]

Related Information

[Navigate SAP Web IDE \[page 165\]](#)

[Workspace Search Options \[page 170\]](#)

[Work with Files and Folders \[page 176\]](#)

[Resize Panes \[page 178\]](#)

6.2.1 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 \[page 166\]](#)

6.3 Workspace 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 \[page 171\]](#)

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

[Search for Files or Content in the Workspace \[page 172\]](#)

Perform a file or string search within a folder or across all projects in your workspace.

[Replace Strings Across Multiple Files \[page 173\]](#)

Perform an advanced find and replace across multiple files with the Search pane.

[Find References \[page 175\]](#)

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 \[page 165\]](#)

Related Information

[Navigate SAP Web IDE \[page 165\]](#)

[Keyboard Shortcuts \[page 166\]](#)

[Work with Files and Folders \[page 176\]](#)

[Resize Panes \[page 178\]](#)

6.3.1 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 [Replacing Strings Across Multiple Files \[page 173\]](#).

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
 - Whole words only
 - To replace text, enter both a search string and a *Replace with* string. To search and replace one by one, click *Replace*. If other instance exists, the next one will be selected. Otherwise, to automatically search and replace all instances, click *All*.
 -
4. If you replaced strings, save the file.

Task overview: [Workspace Search Options \[page 170\]](#)

Related Information

[Search for Files or Content in the Workspace \[page 172\]](#)

[Replace Strings Across Multiple Files \[page 173\]](#)

[Find References \[page 175\]](#)

6.3.2 Search for Files or Content in the Workspace

Perform a file or string search within a folder or across all projects in your workspace.

Context

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

i Note

SAP Web IDE uses index-based searching. The results of an index-based search may not reflect all the latest changes made to a file because the search index may take several minutes until changes to files are recognized by the search. A longer response time for a search may be the result of network conditions and a large-sized workspace with many files.

Procedure

1. Open the [Search](#) pane by choosing the search icon in the right sidebar.
2. In the [Search](#) pane, define the type of search:
 - To search for a file, enter a full or partial file name, then choose [File name](#).
The file name search is case sensitive. If you do not know whether the file name is capitalized, use a wildcard, such as an asterisk (*). For example, instead of searching for `file1`, search for `*ile1`.
 - To search for a string, enter the string, then choose [File content](#).
3. (Optional) Refine search results further:
 - Set [Search in](#) to determine where the search scope is to be limited: use [All](#) to search across the entire workspace, use [Project](#) to search across a single project, or use [Folder](#) to search within the current directory. If you choose [Folder](#) or [Project](#), type the path to the target that you want to search, or select it in the workspace.
 - Set [Find](#) to limit file or content searches to files with a specified extension.
4. Choose [Search](#).
5. In the results list at the bottom of the [Search](#) pane, you see search results, organized according to file.
 - Expand or collapse files with results, which is useful when lists are long.
 - Below each file, matches are highlighted.
 - Hover over a specific result to get the context of that item.If you do a search of file content for the string `test`, you might see results shown as in the following example. You can choose the `/Test/` folder as the target, then change [Find](#) from all file types to `*.js`. Search results are limited to two files. By hovering over the first result in the first file, you can quickly review the context without opening the file.
6. When you have identified the file and row that you require, double-click the specific search result to open the file with the appropriate row highlighted.

Task overview: [Workspace Search Options \[page 170\]](#)

Related Information

[Find and Replace in an Open File \[page 171\]](#)

[Replace Strings Across Multiple Files \[page 173\]](#)

[Find References \[page 175\]](#)

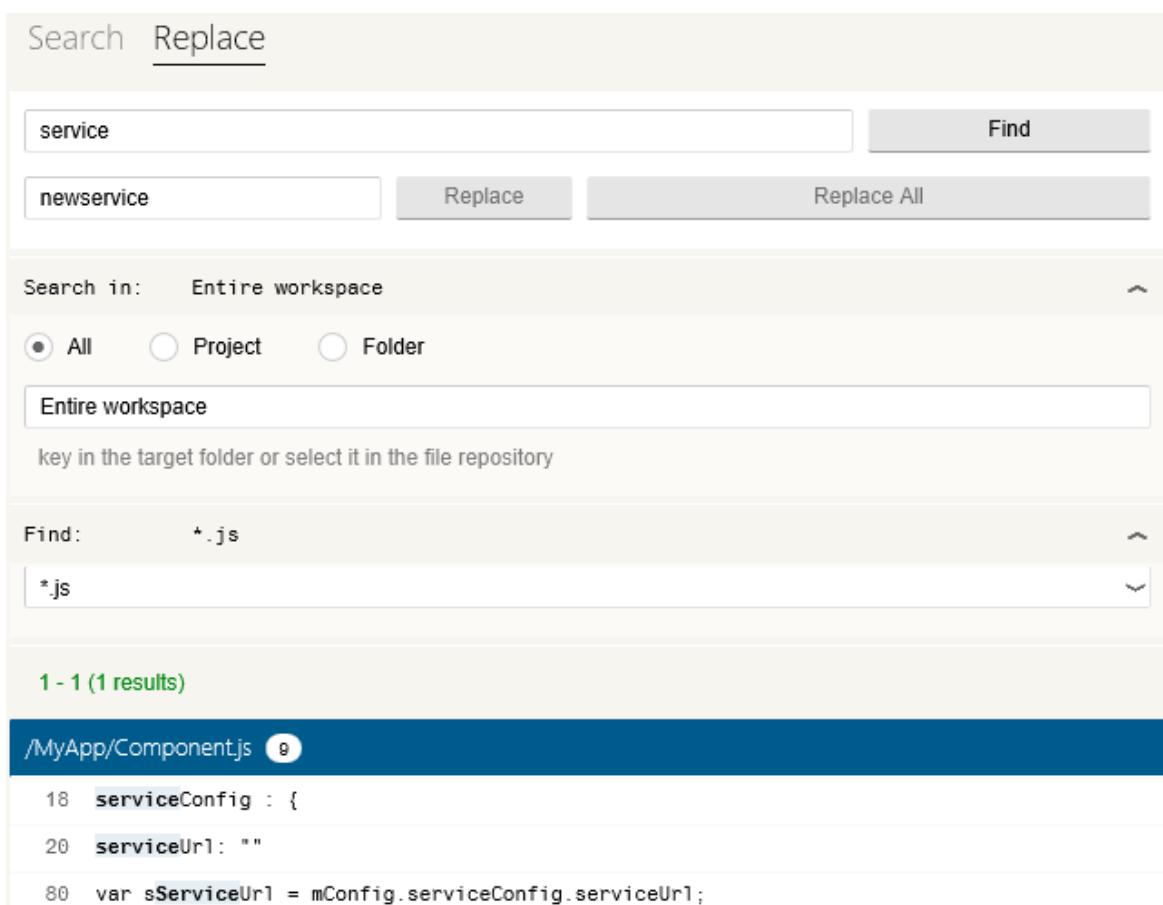
6.3.3 Replace Strings Across Multiple Files

Perform an advanced find and replace across multiple files with the Search pane.

Procedure

1. With or without any files open in the code editor, open the *Search* pane by choosing the search icon in the right sidebar.
2. In the *Search* pane, type the string you want to find.
3. (Optional) Refine search results further:
 - Set *Search in* to determine where the search scope is to be limited: use *All* to search across the entire workspace, use *Project* to search across a single project, or use *Folder* to search within the current directory. If you choose *Folder* or *Project*, enter the path to the target that you want to search.
 - Set *Find* to limit searches to files with a specified extension.
4. Choose *Search*.

In the results list at the bottom of the *Search* pane, you see search results, organized according to file.



5. In the *Replace* pane, enter the replace text and perform one of the following:

- A serial replace that allows you to evaluate each instance case-by-case. In the results list, hover over each result and click the replace icon at the end of the line to replace the string. As each instance is replaced, it disappears from the results list. Skip all results that you do not want to replace.
- An inline replace that allows you to select only specific file results, and choose the replace icon adjacent to the result itself.

- A global replace of all results in the list. Choose *Replace All*.

i Note

You cannot undo the global replace operation.

Task overview: [Workspace Search Options \[page 170\]](#)

Related Information

[Find and Replace in an Open File \[page 171\]](#)

[Search for Files or Content in the Workspace \[page 172\]](#)

[Find References \[page 175\]](#)

6.3.4 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: [Workspace Search Options \[page 170\]](#)

Related Information

[Find and Replace in an Open File \[page 171\]](#)

[Search for Files or Content in the Workspace \[page 172\]](#)

[Replace Strings Across Multiple Files \[page 173\]](#)

6.4 Work with Files and Folders

Use keyboard shortcuts and context menus to work with files and folders.

You can open multiple files in the SAP Web IDE code editor, and you can navigate between them. Before closing files, save or discard your changes. You can cut, copy, paste, and rename files and folders.

i Note

For keyboard shortcuts, see [Keyboard Shortcuts \[page 166\]](#).

Action	Context Menu or Toolbar Icon
Open selected files in the workspace.	 Open with  Code Editor
Search and open files in the workspace.	Open Resource
Navigate between open files in the workspace. For more navigation options in the code editor, see Working in the Code Editor [page 214] .	 (Back)  (Forward)  (Last Edit Location)
Close one or more files.	Tab menu: <ul style="list-style-type: none">• Close• Close All• Close Other (Closes all files except the active file.)
Import a file from the local file system or from an archive to a selected folder.	 Import  From File System
i Note File names can contain only letters, numbers, and the period (.) and underscore (_) characters.	
Check the Extract Archive checkbox to extract files in the archive into the selected folder	
Export selected files or folders.	Export
Archive a folder and save it as a compressed (.zip) file under the parent folder.	Archive
Compressed files larger than 20 MB cannot be archived.	
Refresh the view of the selected files and folders in the workspace.	Refresh

Action	Context Menu or Toolbar Icon
Link the code editor to the workspace to easily see the location of any active file or show the location of the current active file without linking.	Tab menu: <ul style="list-style-type: none">• Link Workspace to Editor• Show in Workspace

[Work with Multiple Open Files \[page 177\]](#)

You can open multiple files in the code editor, each of which appears in a separate tab.

Parent topic: [SAP Web IDE Basics \[page 165\]](#)

Related Information

[Navigate SAP Web IDE \[page 165\]](#)

[Keyboard Shortcuts \[page 166\]](#)

[Workspace Search Options \[page 170\]](#)

[Resize Panes \[page 178\]](#)

6.4.1 Work with Multiple Open Files

You can open multiple files in the code editor, each of which appears in a separate tab.

Tab Area

If more than one file is opened, each one is shown in a different editor tab in the code editor. Each tab shows the file name and its extension as well as an **x**, which allows you to close the file by clicking it. If a file is being edited, its file name is shown in bold and a small asterisk appears next to its name.

When you hover over the file name, the file path appears in a tooltip.

Maximizing the Editing Area

To enlarge the editing area, double-click any tab in the editor. All open SAP Web IDE panes disappear, and the complete current editor area is fitted to your screen. To restore, double-click the tab.

Tab Overflow

If you open more files in the editor than there is space to display their tabs in the editor area, a small menu appears on the right of the tab area:



Use the arrows to navigate through the tabs in either direction. You can also click the list icon on the right to see the names of all opened files and then choose one from the list to make it the active editor tab.

Parent topic: Work with Files and Folders [page 176]

6.5 Resize 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.

i Note

Double-click again to restore the editor window to its original size. This always redisplays 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 \[page 165\]](#)

Related Information

[Navigate SAP Web IDE \[page 165\]](#)

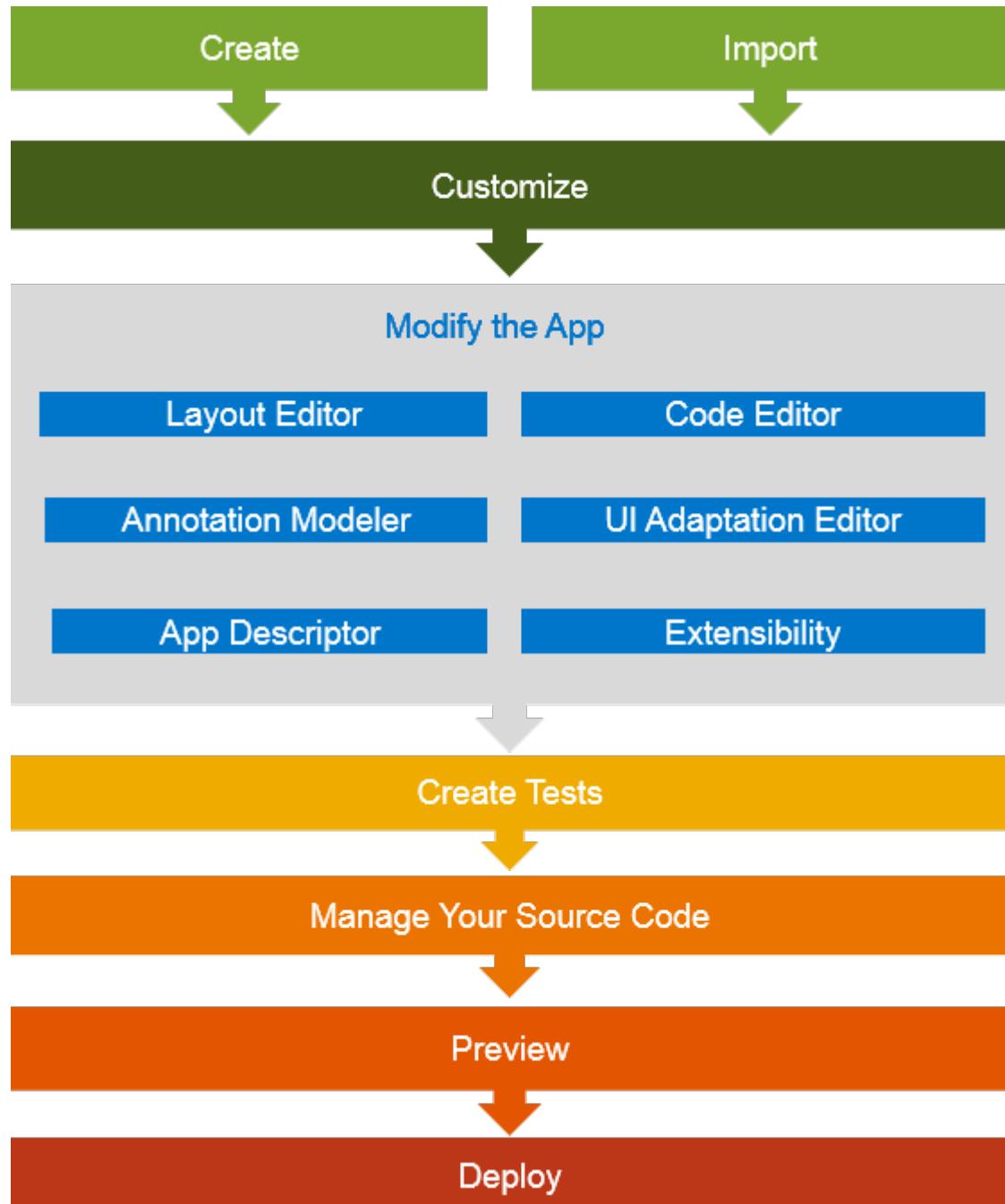
[Keyboard Shortcuts \[page 166\]](#)

[Workspace Search Options \[page 170\]](#)

[Work with Files and Folders \[page 176\]](#)

7 Developing

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



- [Customizing Your Project \[page 193\]](#)
- [Creating Projects \[page 180\]](#)
- [Importing Projects \[page 190\]](#)
- [Creating Projects \[page 180\]](#)
- [Customizing Your Project \[page 193\]](#)
- [Layout Editor \[page 332\]](#)
- [Developing Applications \[page 213\]](#)
- [Annotation Modeler \[page 369\]](#)
- [UI Adaptation Editor \[page 395\]](#)
- [Extending SAPUI5 Applications \[page 457\]](#)
- [Developing Application Tests \[page 331\]](#)
- [Using Source Control \(Git\) \[page 402\]](#)
- [Running Applications in Development Mode \[page 432\]](#)
- [Deploying Applications \[page 447\]](#)
- [Modifying the Application Descriptor Configuration \[page 205\]](#)

7.1 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 Templates \[page 181\]](#)

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

[Create Projects from Sample Applications \[page 188\]](#)

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

[Add a New Component \[page 189\]](#)

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

Related Information

[Create a Quick Start Application with the Layout Editor \[page 348\]](#)

7.1.1 Create Projects from Templates

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 and choose *Next*.

Note

You can click the heart at the top of any template tile to make it a favorite. It is then included in the *Favorites* category and is displayed by default the next time that you open the wizard.

You can 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.

3. Enter a name for your project and choose *Next*.

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* and select the desired data source. Next, select a service.

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.

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 \[page 183\]](#).

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

Note

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

i Note

If you select a data source 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 \[page 432\]](#).

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 will be 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:

- o 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.

- o Choose *File System* and browse for the relevant annotation file in your file system.
- o Choose *Workspace* and browse for the relevant annotation file in the SAP Web IDE system.
- o 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 subsequent 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 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](#).

7.1.1.2 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](#).

7.1.1.3 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 Cloud Platform account.

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 Cloud Platform
- `odata_hci`: for accessing OData services in an SAP Cloud Platform 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 the OData services on an API management system

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

For more information, see [Connect to ABAP Systems \[page 158\]](#).

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 \[page 158\]](#).

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

Service Details	Description
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

Entity Set Details	Description
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 \[page 187\]](#).
- 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 Templates \[page 181\]](#) for more information.

7.1.1.3.1 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:
 - a. 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.
 - b. 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.
 - c. 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*).

- d. 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*).

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

Related Information

[Select a Data Source \[page 183\]](#)

7.1.2 Create Projects from Sample Applications

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*.

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.

Note

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

7.1.3 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 \[page 432\]](#).

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.

7.2 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 \[page 190\]](#)

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

[Import Applications from SAP Cloud Platform \[page 191\]](#)

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

[Import Applications from the SAPUI5 ABAP Repository \[page 192\]](#)

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

7.2.1 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.

Procedure

- From the workspace, select the folder to which you want to import the project, for example, the top-level **Workspace** folder.
 - 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.
- In the **File** field, browse to the location of your archived project file, select it, and choose **Open**.
 - 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.

7.2.2 Import Applications from SAP Cloud Platform

You can import an existing application from SAP Cloud Platform 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 Cloud Platform.

Procedure

1. In SAP Web IDE, choose **File** **Import** **Application from SAP Cloud Platform**. The *Select Application from SAP Cloud Platform* 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.

Note

You cannot import an application to an already existing project.

Importing an application from SAP Cloud Platform 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.

7.2.3 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 [Requirements for Connecting to ABAP Systems \[page 161\]](#)

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.

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](#)

7.3 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.

[Configure Annotation Modeler to Use Mock Data \[page 194\]](#)

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

[Customize JavaScript Beautifier Properties \[page 195\]](#)

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

[Customize Code Checking Rules \[page 196\]](#)

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

[Customize Code Checking Triggers \[page 197\]](#)

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

[Define Application Languages \[page 197\]](#)

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.

[Use the Translation Hub \[page 198\]](#)

You can translate your project's `i18n.properties` file using the Translation Hub service.

[Configure Mock Data Usage \[page 199\]](#)

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

[Set Project Types \[page 200\]](#)

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

[Customize Performance Measures \[page 201\]](#)

Define the thresholds that trigger error and warning alerts in the Performance Results table.

[Set the SAPUI5 Version \[page 201\]](#)

Configure the SAPUI5 version for your project.

[Configure Run Configurations for the UI Adaptation Editor \[page 202\]](#)

Select the Run Configuration to be used when running your project in the UI Adaptation editor.

[Use the Smart Business Service \[page 203\]](#)

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

7.3.1 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 `f1pSandboxMockServer.html` file.
- e. In the *Mock Data* section, choose *Run with mock data*.

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

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

7.3.2 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 <i>2</i> .
<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 <i>if/then</i> 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.

7.3.3 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 \[page 245\]](#).
- Configure your own JavaScript code checking rules and use them for your project. For more information, see [Create JavaScript Code Checking Rules \[page 244\]](#).
- Create your own code checking plugin, then choose the plugin in the corresponding code checking configuration pane.

Related Information

[Configure Code Checking \[page 240\]](#)

[Checking Code \[page 239\]](#)

7.3.4 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.

Note

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.

7.3.5 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 Projects \[page 435\]](#)

7.3.6 Use the Translation Hub

You can translate your project's `i18n.properties` file using the Translation Hub service.

Procedure

Before you can translate your project's `i18n.properties` file, you must configure your system to consume the SAP Translation Hub.

1. Make sure your project is attached to a Git repository.
2. Enable the SAP Translation Hub service for your account.
 - a. In the SAP Cloud Platform cockpit, choose *Services* in the navigation tree.
 - b. Choose *SAP Translation Hub*.
 - c. Choose *Enable*.
3. Assign the roles of the users who will access the services in SAP Web IDE. For more information, see [User Authentication and Authorization](#).
4. Create a destination named `translationHubBeta` for your account with the following information:
 - Description=Translation
 - Type=HTTP
 - TrustAll=true
 - Authentication=AppToAppSSO
 - Name = `translationHubBeta`
 - WebIDEEnabled=true
 - CloudConnectorVersion=1
 - URL= <base URL of your account subscribed SAP Translation Hub>/uiTo obtain the URL:
 1. In the SAP Cloud Platform cockpit, choose *Services* in the navigation pane.
 2. Choose *SAP Translation Hub*.
 3. Choose *Go To Service* and copy the application URL.
 - ProxyType=Internal

Translating your Properties File

Procedure

1. Select the `i18n.properties` file of the project you want to translate.
2. Right-click and select *Generate Translation Files*.
3. Select the application domain and the desired target languages.
4. Choose *Generate*.
5. Enter your Git password and choose *Submit*. The Translation Hub Service will generate the translation files and push them back to your git repository. To get these files to your workspace, pull the changes from Git.

7.3.7 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 Cloud

Platform. 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.

7.3.8 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.

7.3.9 Customize Performance Measures

Define the thresholds that trigger error and warning alerts in the Performance Results table.

Context

i Note

This feature is not available in SAP Web IDE personal edition.

When you view performance measurements in the application preview, alerts are displayed for measurements that exceed specified warning and error thresholds. You can change the default thresholds for the project.

Procedure

1. From the context menu of any file in your project, select  .
2. In the *Performance Measures* pane, edit the threshold values for warning alerts and error alerts for each measure.

7.3.10 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*.

- 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.

7.3.11 Configure Run Configurations for the UI Adaptation Editor

Select the Run Configuration to be used when running your project in the UI Adaptation editor.

Context

When an application is running in the UI Adaptation 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

- From the project context menu, select *Project Settings*.
- From the Project Settings options, select *UI Adaptation*.

i Note

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

The *UI Adaptation* pane displays the configurations that the application currently has in its Run Configuration.

- Select the configuration to be used when running the app in the UI Adaptation editor.
- Save your changes.

i Note

The following configurations will not apply in the UI Adaptation editor:

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

7.3.12 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 Platfrom 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`

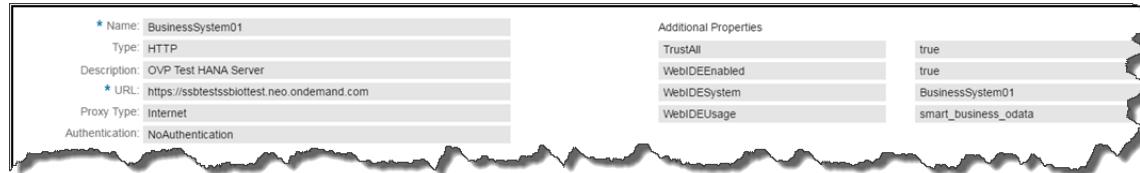
This destination is used to generate the smart business annotation.

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 Buisness service destination:



3. Create an application using the Smart Business service. Use *Service URL* as the source for your destination.

For example:

The screenshot shows the SAP Web IDE Service Catalog interface. At the top, a message says "Service: SMART_BUSINESS.xsodata selected." Below it, a note says "Choose a service from one of the sources listed below." On the left, a sidebar titled "Sources" lists "Service Catalog", "Workspace", "File System", and "Service URL". The "Service URL" option is highlighted with a blue background. To its right, a section titled "Choose a system to connect to the required service" shows "OVP Test HANA Server" selected in a dropdown, with the URL "/sap/smartsbusiness/core/odata/modeler/SMART_BUSINESS.xsodata" entered in a text input field. A "Test" button and a "Show Details" link are also present. The main content area is titled "Service" and displays a tree view of the service structure under "SMART_BUSINESS.xsodata", including nodes like "Chips", "Catalogs", "INDICATORS", etc.

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.

The screenshot shows the SAP Web IDE Annotation Selection interface. The title is "Annotation Selection". It contains a note: "Select the desired annotation files and rank them in the order in which they will be loaded. Note: If the annotation files overlap, the one loaded last will overwrite the previous ones." Below this is a table with a header row: "Rank", "Name", and "Source". There is one entry in the table: Rank 1, Name "Smart_Business", and Source "Remote".

Rank	Name	Source
1	Smart_Business	Remote

7.4 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
- *Routing* tab
- *Navigation* tab

2. Change the settings in each tab as required.

[Settings Tab Options \[page 206\]](#)

[Data Sources Tab Options \[page 208\]](#)

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

[Models Tab Options \[page 208\]](#)

Define the data models for your application.

[Routing Tab Options \[page 209\]](#)

Define routes and targets for the application.

[Navigation Tab Options \[page 211\]](#)

7.4.1 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 ({{ }}).
Description	Enter the description of the application inside double curly brackets ({{ }}).
Source Template	The ID and version of the template from which the application was generated.
I18n 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 ({{ }}).
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.

Option	Description
Application Icons	<p>The icons used in the application:</p> <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	<p>JavaScript and CSS resources that are required by your application.</p> <p>To add a resource, click +, and in the popup, choose the resource type and enter its URI.</p>
Dependencies	<p>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.</p> <p>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.</p>
Content Densities	<p>You can define different content densities for certain controls that allow your application to adapt to specific devices:</p> <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.

7.4.2 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.
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 .

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.

7.4.3 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 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 \[page 208\]](#)

7.4.4 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 View Path is <code>myApp</code> , the created view is <code>myApp.myView</code> .
View Type	Type of view that is created: <code>XML</code> , <code>JSON</code> , <code>JS</code> , <code>HTML</code> , or <code>Template</code> .
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 View Level is used to define the transition direction when navigating to this view.
Control Aggregation	Name of an aggregation of the Control ID (target control) that contains views. For example, a NavContainer 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: <code>slide</code> , <code>flip</code> , <code>fade</code> , or <code>show</code> .
Target Parent	ID of the view that contains the control specified by the Control ID 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.

7.4.5 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 ({{ }}).
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 sap.app 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"> • plain - the parameter value is taken as a literal string value. • reference - the parameter value is a reference to another parameter.
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.

Option	Description
Filter Format	<p>The format of the filter value:</p> <ul style="list-style-type: none"> • <i>plain</i> - the filter value must match the actual parameter value. • <i>regexp</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.

7.5 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 \[page 214\]](#)

Use keyboard shortcuts and context menus to easily edit and navigate through your code and code comments.

[Using Code Completion \[page 220\]](#)

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 \[page 239\]](#)

SAP Web IDE performs code checking, also known as validation, and displays errors as annotations.

[Developing Apps Using SAP Fiori Elements \[page 325\]](#)

This section contains information about creating SAP Fiori apps using SAP Fiori elements in SAP Web IDE.

[Using the Outline Pane for JavaScript Files \[page 329\]](#)

The Outline pane helps you to understand the structure of JavaScript files and to navigate through the code.

[Creating an HTML5 Application Descriptor File \[page 330\]](#)

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.

7.5.1 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

The following keyboard shortcuts are for Microsoft Windows. For Mac OS, see [Keyboard Shortcuts \[page 166\]](#).

Basic Navigating and Editing

Action	Keyboard Shortcut	Context Menu
Move one tab to the right	[Alt] + [R]	-
Move one tab to the left	[Alt] + [Q]	-
Close all tabs to the right of the open tab		Close Tabs to the Right
Undo edit	[Ctrl] + [Z]	Undo
Redo edit	[Ctrl] + [Y]	Redo
Show and hide control characters, such as space, tab, newline, and paragraph	[Ctrl] + [I] (uppercase i)	-
Move a line up	[Alt] + [Up arrow]	-
Move a line down	[Alt] + [Down arrow]	-
Move to a specific line	[Ctrl] + [L]	Go to Line
Copy a line to the line above or below	[Alt] + [Shift] + [Up arrow] [Alt] + [Shift] + [Down arrow]	-

Action	Keyboard Shortcut	Context Menu
Expand the entire hierarchy of file elements	[Alt] + [Shift] + [F2]	Gutter context menu: <i>Expand All</i>
Collapse the entire hierarchy of file elements	[Ctrl] + [Alt] + [F2]	Gutter context menu: <i>Collapse All</i>
Search for a string	[Ctrl] + [F]	<i>Find</i>
Search and replace a string	[Ctrl] + [H]	<i>Find and Replace</i>

Commenting

Action	Keyboard Shortcut	Context Menu
For JavaScript and XML files	[Ctrl] + [/]	<i>Toggle Line Comment</i>
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 // . For example:		
<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 /* and */ to wrap the comment block. For example:	[Ctrl] + [Shift] + [/]	<i>Toggle Block Comment</i>
<pre>/* This is a comment block. Use this block comment style when comments span multiple lines.*/ //Use the comment style for short comments.</pre>		
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.	[Ctrl] + [Alt] + [T]	<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 \[page 195\]](#).

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.

Git Annotations

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.

Use the context menu [Show Git Annotations](#) to open a new read-only editor tab with the Git information.

In the new tab with the Git information, you can right-click a commit and open the [Git History](#) pane for that commit.

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.

7.5.1.1 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 and behavior.

Option	Description
Appearance	
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. 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.

Option	Description
Show invisible characters	Shows white-space characters such as spaces, tabs, and new lines.
Input Behavior	
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.

Test your settings in the preview.

3. Change the [Autosave](#) setting, if required.
4. Click [Save](#).

7.5.1.2 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.

7.5.1.3 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.

i 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
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>
```

7.5.2 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 \[page 223\]](#).
- Manual code completion
Supported for various file types including `i18n` and `messagebundle` property files.

- i18 and messagebundle
- JavaScript
- XML

No 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).

i 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.

• Note

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.

The screenshot shows a code editor with the following code:

```
formatConcatenateNameId : function(sName, sID) {
    /* @type sap.m.Button */
    var x = sap.ui.getCore().byId("btnBack");
    x.s
}
```

A tooltip for `setIconDensityAware(bIconDensityAware)` is displayed, containing the following information:

setIconDensityAware(bIconDensityAware) : sap.m.Button

Sets a new value for property `iconDensityAware`. By default, this is set to true but then one or more requests are sent trying to get the density perfect version of image if this version of image doesn't exist on the server. If only one version of image is provided, set this value to false to avoid the attempt of fetching density perfect image. When called with a value of null or undefined, the default value of the property will be restored. Default value is true.

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.

The screenshot shows a code editor with the following code:

```
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
10 - <Page backgroundDesign="|">
11   |
12   |
13   </Page>
14 </core:View>
```

A tooltip for the `backgroundDesign` attribute is displayed, containing the following information:

List

Page background color when a List is set as the Page content.

7.5.2.1 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*.

7.5.2.2 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.

7.5.2.3 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](#).

7.5.2.4 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.

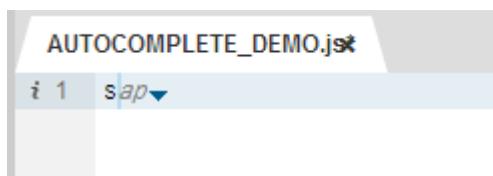
Context

This example demonstrates the following code completion features:

- JavaScript code completion
 - Element hints
 - Snippet insertion

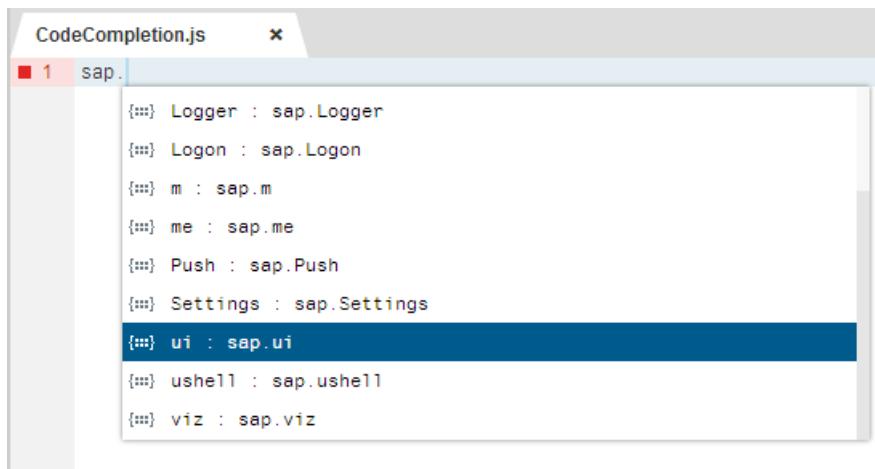
Procedure

1. In the JavaScript editor, type **s**, and notice that the autocomplete suggestions automatically appear adjacent to the letter in context.



- When you see `sap`, press `Enter`.
 - 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.



7. 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.

A screenshot of the SAP Web IDE interface. The title bar says "AUTOCOMPLETE_DEMO.js*". The code editor shows the following JavaScript code:

```
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2     ... 
3     ...
4     }
5 );
```

A code completion dropdown menu is open at the cursor position, showing the following suggestions:

- (x) valueOf() : Object
- var - variable declaration
- var - variable declaration with value
- VariantLayoutData in sap.ui.core
- View in sap.ui.core.mvc
- ValueHelpField in sap.ui.commons
- VerticalLayout in sap.ui.commons.layout

8. 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.

A screenshot of the SAP Web IDE interface. The title bar says "CodeCompletion.js*". The code editor shows the following JavaScript code:

```
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2     ... 
3     ...
4     }
5 ));
```

The word "var" is being typed by the user, and the code completion dropdown is visible, showing the suggestion "var - variable declaration".

9. Name the var as **fcl = form**, then press **Ctrl** + **Space** to retrieve a list of form templates for it.

A screenshot of the SAP Web IDE interface. The title bar says "AUTOCOMPLETE_DEMO.js*". The code editor shows the following JavaScript code:

```
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2   ...
3   var fc1 = form
4 }
```

A code completion dropdown menu is open over the word "form". It lists several suggestions, each preceded by a small icon:

- FormattedText in sap.ui.commons
- Form in sap.ui.commons.form
- FormContainer in sap.ui.commons.form
- FormElement in sap.ui.commons.form
- FormLayout in sap.ui.commons.form
- Form in sap.ui.layout.form
- FormContainer in sap.ui.layout.form
- FormElement in sap.ui.layout.form
- FormLayout in sap.ui.layout.form

The suggestion "FormContainer in sap.ui.layout.form" is highlighted with a blue selection bar.

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.

A screenshot of the SAP Web IDE interface. The title bar says "CodeCompletion.js*". The code editor shows the following JavaScript code:

```
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2   ...
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});
```

A code completion dropdown menu is open over the "id" property of the "FormContainer" object. The "id" property is highlighted with a yellow selection bar. The dropdown menu lists the following properties for "FormContainer":

- id: "id", // sap.ui.core.ID
- expanded: true, // boolean
- expandable: false, // boolean
- visible: true, // boolean
- tooltip: undefined, // sap.ui.core.TooltipBase
- customData: [], // sap.ui.core.CustomData
- formElements: [], // sap.ui.layout.form.FormElement
- title: undefined // sap.ui.core.Title

7.5.2.5 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

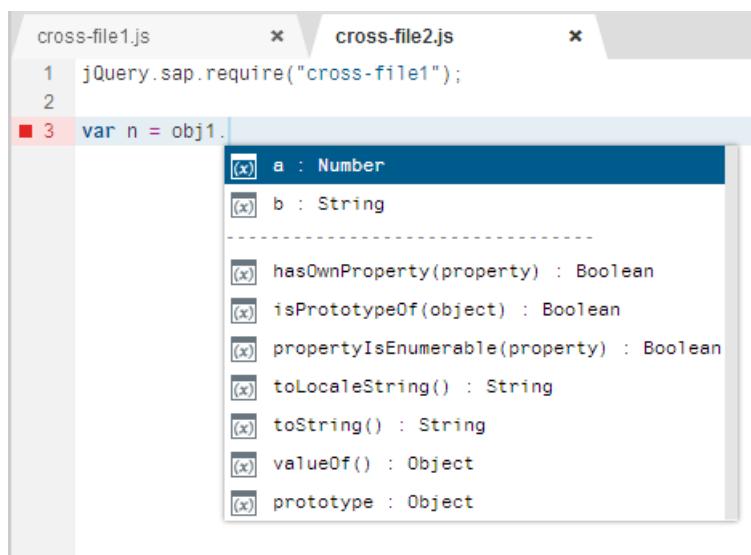
SAP Web IDE code completion automatically displays user-defined objects in the same project.

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
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 to use code completion:
 - Either type `var n = obj1`, then press `Ctrl + Space` to get this hint from the first file:



- Or type `var f = fun`, then press `Ctrl + Space` to get this hint from the first file:



7. Select the first result and press `Enter`.

Results

The code updates accordingly in the second file.

7.5.2.6 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.

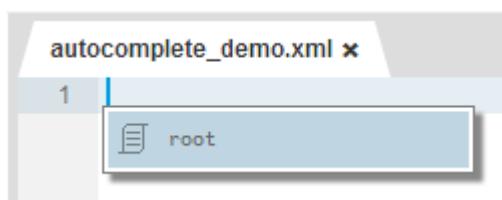
Context

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.

```
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
```

A screenshot of the SAP Web IDE showing the generated XML code for the 'root' node. The code is as follows:

```
<core:View
  xmlns:core="sap.ui.core"
  xmlns="sap.m"
  xmlns:commons="sap.ui.commons"
  xmlns:layout="sap.ui.layout"
  controllerName=""
  viewName="">
</core:View>
```

2. In row 8, type `<p`.

The screenshot shows a code editor window titled "autocomplete_demo.xml*x". The code is as follows:

```
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>
```

The cursor is at the end of the line "8 <panel>".

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.

The screenshot shows a code editor window titled "autocomplete_demo.xml*x". The code is as follows:

```
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
```

The cursor is at the end of the line "8 <Page". A context menu is open, listing the following suggestions:

- <> />
- <> >
- 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
```

7.5.2.7 Try It: XML Code Snippet Insertion

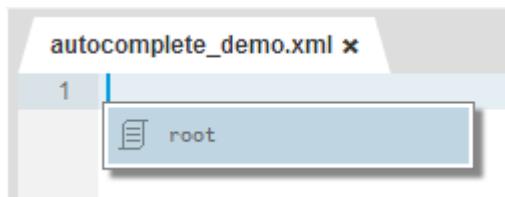
A self-guided example that demonstrates how to insert an XML code snippet.

Prerequisites

Open a new XML file in a new or existing project.

Procedure

1. Press **Ctrl** + **Space** and select the `root` node suggestion.



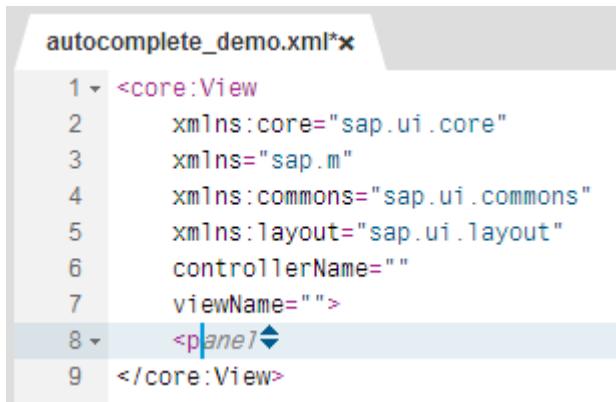
The `root` node is automatically populated with required elements and values.

```
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
```

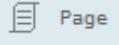
A screenshot of the SAP Web IDE showing the completed XML code for the 'root' node. The code is:

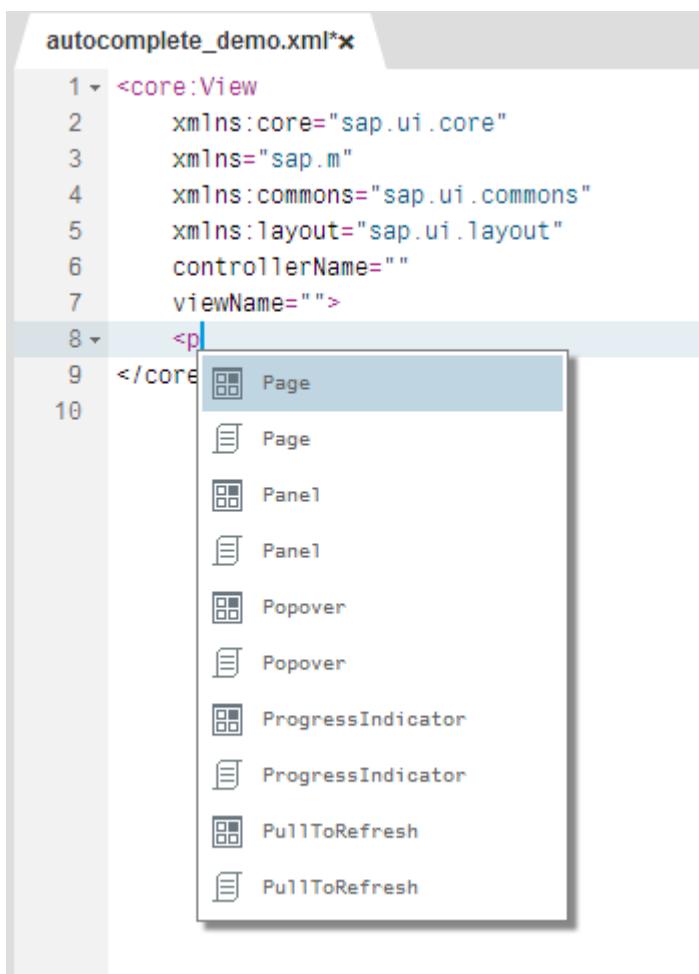
```
<core:View
  xmlns:core="sap.ui.core"
  xmlns="sap.m"
  xmlns:commons="sap.ui.commons"
  xmlns:layout="sap.ui.layout"
  controllerName=""
  viewName="">
</core:View>
```

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>
```

3. Press the arrow keys to scroll through options. Select Page with the snippet icon (), 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:View>
10
```

The snippet dropdown menu contains the following items:

- 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>
```

7.5.2.8 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, autocomplete 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.



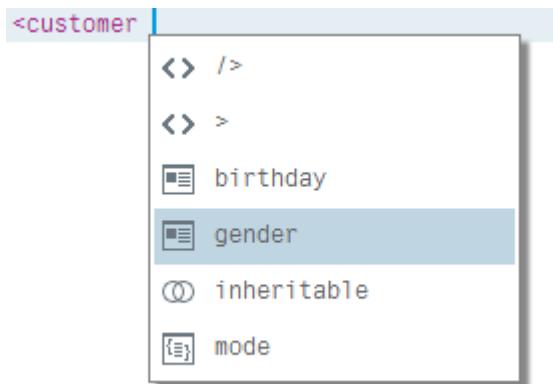
Because `<xs:anyAttribute/>` is defined for the `employee` element, the `gender` attribute from `demo2b.xsd` appears in this list.

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.

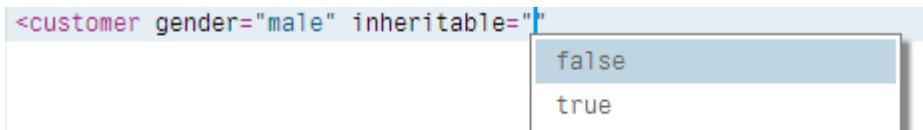


i Note

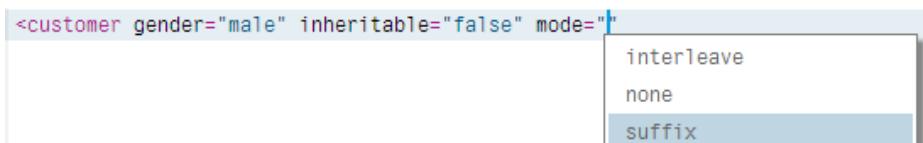
Note the icons in this example: the `mode` attribute's icon shows an enum type, and the `inheritable` attribute's icon shows a boolean type.

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



7.5.2.9 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

7.5.3 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 \[page 240\]](#)

Configure when to trigger code checking, also known as code validation, and the level of messages to display.

[Code Checking Annotations \[page 241\]](#)

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 \[page 243\]](#)

Review the default JavaScript validator configuration. Customizations always override these defaults.

[XML Validation \[page 297\]](#)

You can configure which XML validator to use in your project.

[Validation of neo-app.json Files \[page 323\]](#)

A project's neo-app.json file is validated on opening in the code editor.

[Using the Problems View \[page 323\]](#)

View information about problems in the projects in your workspace.

[Validation of manifest.json Files \[page 324\]](#)

Application descriptor files (manifest.json) for SAP Fiori projects are validated on opening in the code editor.

Related Information

[Customize Code Checking Rules \[page 196\]](#)

7.5.3.1 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 \[page 239\]](#).

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 \[page 239\]](#)

Related Information

[Code Checking Annotations \[page 241\]](#)

[JavaScript Validation \[page 243\]](#)

[XML Validation \[page 297\]](#)

[Validation of neo-app.json Files \[page 323\]](#)

[Using the Problems View \[page 323\]](#)

[Validation of manifest.json Files \[page 324\]](#)

[Customize Code Checking Rules \[page 196\]](#)

7.5.3.2 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 \[page 239\]](#)

Related Information

[Configure Code Checking \[page 240\]](#)

[JavaScript Validation \[page 243\]](#)

[XML Validation \[page 297\]](#)

[Validation of neo-app.json Files \[page 323\]](#)

[Using the Problems View \[page 323\]](#)

[Validation of manifest.json Files \[page 324\]](#)

7.5.3.2.1 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

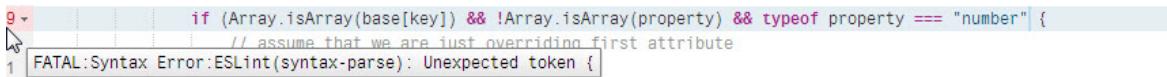
Configure validation to run on change as described in [Configure Code Checking \[page 240\]](#). Then follow the steps in [Opening and Reading Files with Annotations \[page 241\]](#) to open a file named Linter.js.

Procedure

- Once the Linter.js file is open, use the colored square flags to determine which potential issues are the most serious.

```
*Linter.js
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 HIGH:Possible Error:ESLint(semi): Missing semicolon.
800     if (!Array.isArray(base[key]) && !Array.isArray(property) && typeof property == "number") {
801         // assume that we are just overriding first attribute
802         base[key][0] = custom[key];
803         return;
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});
```

- 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.



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.

7.5.3.3 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	See these resources for additional support: <ul style="list-style-type: none">• For ESLint configuration details, see Configuring ESLint.• For ESLint rule information, see Rules.
ESLint Rule Metadata Defaults	Attributes can have multiple supported values. Use the following: <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 \[page 239\]](#)

Related Information

[Configure Code Checking \[page 240\]](#)

[Code Checking Annotations \[page 241\]](#)

[XML Validation \[page 297\]](#)

[Validation of neo-app.json Files \[page 323\]](#)

[Using the Problems View \[page 323\]](#)

[Validation of manifest.json Files \[page 324\]](#)

7.5.3.3.1 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 .
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 .
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.

7.5.3.3.2 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*. 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.

7.5.3.3.3 Fiori JavaScript Validator Rules

The Fiori JavaScript validator rules are used when building Fiori projects.

[sap-cross-application-navigation \[page 247\]](#)

No static cross-application navigation targets are allowed.

[sap-forbidden-window-property \[page 249\]](#)

Detects the usage of forbidden window properties.

[sap-no-navigator \[page 250\]](#)

Detects `window.navigator` usage.

[sap-no-override-rendering \[page 252\]](#)

Override of control methods is not allowed.

[sap-no-override-storage-prototype \[page 253\]](#)

Override of `csap-no-override-storage-prototype` is not allowed.

[sap-no-proprietary-browser-api \[page 254\]](#)

Discourage usage of proprietary browser APIs.

[sap-no-sessionstorage \[page 256\]](#)

Usage of session storage is not allowed.

[sap-no-ui5-prop-warning \[page 257\]](#)

Usage of private members of SAPUI5 objects is not allowed.

[sap-no-ui5base-prop \[page 258\]](#)

Usage of private members of SAPUI5 objects is not allowed

[sap-timeout-usage \[page 260\]](#)

Discourage usage of `setTimeout`.

[sap-ui5-no-private-prop \[page 262\]](#)

Detects usage of private properties of SAPUI5 objects.

[sap-usage-basemastercontroller \[page 264\]](#)

Detects usage of `BaseMasterController`.

[sap-message-toast \[page 266\]](#)

Wrong usage of `sap.m.MessageToast` is not allowed.

[sap-no-absolute-component-path \[page 267\]](#)

Absolute paths to component `includes` are not allowed.

[sap-no-br-on-return \[page 269\]](#)

Detects the usage of `document.queryCommandSupported`.

[sap-no-dom-access \[page 270\]](#)

Usage of certain methods of `document` is discouraged.

[sap-no-dom-insertion \[page 272\]](#)

Usage of DOM insertion methods is not allowed.

[sap-no-dynamic-style-insertion \[page 273\]](#)

Detects dynamic style insertion.

[sap-no-element-creation \[page 275\]](#)

Direct DOM insertion is not allowed.

[sap-no-encode-file-service \[page 276\]](#)

Detects the usage of `encode_file` service

[sap-no-exec-command \[page 278\]](#)

Detects direct DOM manipulation.

[sap-no-global-define \[page 279\]](#)

Detects definition of globals via window object.

[sap-no-global-event \[page 280\]](#)

Detects global event handling overrides.

[sap-no-global-selection \[page 282\]](#)

Usage of global selection is discouraged.

[sap-no-global-variable \[page 283\]](#)

Global variables should not be used in SAP Fiori applications.

[sap-no-hardcoded-color \[page 284\]](#)

Usage of hard coded colors is not allowed.

[sap-no-hardcoded-url \[page 286\]](#)

Use of hardcoded URLs is not allowed.

[sap-no-history-manipulation \[page 287\]](#)

Direct history manipulation is discouraged.

[sap-no-jquery-device-api \[page 289\]](#)

Usage of the jQuery device APIs is not allowed.

[sap-no-localhost \[page 290\]](#)

Usage of localhost is not allowed.

[sap-no-localstorage \[page 292\]](#)

Usage of local storage is not allowed.

[sap-no-location-reload \[page 293\]](#)

Detects location reload.

[sap-no-location-usage \[page 295\]](#)

Override of location properties and methods is not allowed.

7.5.3.3.1 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 245]

Related Information

[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]

[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.3.2 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]

[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.3 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 245]

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.4 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)

[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.5 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]

[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.6 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 245]

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.7 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)

[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.8 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)

[sap-no override-storage-prototype](#) [page 253]
[sap-no proprietary-browser-api](#) [page 254]
[sap-no sessionstorage](#) [page 256]
[sap-no ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.9 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:

sap.ui.base.ManagedObject	mProperties, mAggregations, mAssociations, mMethods, oParent, aDelegates, aBeforeDelegates, iSuppressInvalidate, oPropagatedProperties, oModels, oBindingContexts, mBindingInfos, sBindingPath, mBindingParameters, mBoundObjects
sap.ui.base.EventProvider	mEventRegistry, oEventPool
sap.ui.base.Event	oSource, mParameters, sId
sap.ui.model.odata.ODataModel, sap.ui.model.odata.v2.ODataModel	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

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 \[page 245\]](#)

Related Information

- [sap-cross-application-navigation \[page 247\]](#)
- [sap-forbidden-window-property \[page 249\]](#)
- [sap-no-navigator \[page 250\]](#)
- [sap-no-override-rendering \[page 252\]](#)
- [sap-no-override-storage-prototype \[page 253\]](#)
- [sap-no-proprietary-browser-api \[page 254\]](#)
- [sap-no-sessionstorage \[page 256\]](#)

[sap-no-ui5-prop-warning](#) [page 257]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.10 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 245]

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.3.11 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.common
- 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)

[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.12 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.13 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)

[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.14 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', "  
    + "'/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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.15 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)

[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.16 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');
```

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.17 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)

[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.18 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 245]

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.19 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)

[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.3.20 sap-no-encode-file-service

Detects the usage of `encode_file` service

The `encode_file` service is deprecated and not available on SAP Cloud Platform.

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 245]

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.21 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)

[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.3.22 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]

sap-forbidden-window-property [page 249]
sap-no-navigator [page 250]
sap-no-override-rendering [page 252]
sap-no-override-storage-prototype [page 253]
sap-no-proprietary-browser-api [page 254]
sap-no-sessionstorage [page 256]
sap-no-ui5-prop-warning [page 257]
sap-no-ui5base-prop [page 258]
sap-timeout-usage [page 260]
sap-ui5-no-private-prop [page 262]
sap-usage-basemastercontroller [page 264]
sap-message-toast [page 266]
sap-no-absolute-component-path [page 267]
sap-no-br-on-return [page 269]
sap-no-dom-access [page 270]
sap-no-dom-insertion [page 272]
sap-no-dynamic-style-insertion [page 273]
sap-no-element-creation [page 275]
sap-no-encode-file-service [page 276]
sap-no-exec-command [page 278]
sap-no-global-event [page 280]
sap-no-global-selection [page 282]
sap-no-global-variable [page 283]
sap-no-hardcoded-color [page 284]
sap-no-hardcoded-url [page 286]
sap-no-history-manipulation [page 287]
sap-no-jquery-device-api [page 289]
sap-no-localhost [page 290]
sap-no-localstorage [page 292]
sap-no-location-reload [page 293]
sap-no-location-usage [page 295]

7.5.3.3.23 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)

[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.3.24 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]

[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.25 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]

[sap-no override-rendering](#) [page 252]
[sap-no override-storage-prototype](#) [page 253]
[sap-no proprietary-browser-api](#) [page 254]
[sap-no sessionstorage](#) [page 256]
[sap-no ui5-prop-warning](#) [page 257]
[sap-no ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.26 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-url \[page 286\]](#)

[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.3.27 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]

[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.28 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]

[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.29 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 considered warnings:

```
if (!sap.ui.Device.system.desktop) {  
    this.getView().byId("factSheetButton").setVisible(false);  
}
```

[Parent topic: Fiori JavaScript Validator Rules](#) [page 245]

Related Information

[sap-cross-application-navigation](#) [page 247]

[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]
[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-localhost](#) [page 290]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.30 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)

[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localstorage](#) [page 292]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.3.31 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 245]

Related Information

[sap-cross-application-navigation](#) [page 247]
[sap-forbidden-window-property](#) [page 249]
[sap-no-navigator](#) [page 250]
[sap-no-override-rendering](#) [page 252]
[sap-no-override-storage-prototype](#) [page 253]
[sap-no-proprietary-browser-api](#) [page 254]
[sap-no-sessionstorage](#) [page 256]
[sap-no-ui5-prop-warning](#) [page 257]
[sap-no-ui5base-prop](#) [page 258]
[sap-timeout-usage](#) [page 260]
[sap-ui5-no-private-prop](#) [page 262]
[sap-usage-basemastercontroller](#) [page 264]
[sap-message-toast](#) [page 266]
[sap-no-absolute-component-path](#) [page 267]
[sap-no-br-on-return](#) [page 269]
[sap-no-dom-access](#) [page 270]

[sap-no-dom-insertion](#) [page 272]
[sap-no-dynamic-style-insertion](#) [page 273]
[sap-no-element-creation](#) [page 275]
[sap-no-encode-file-service](#) [page 276]
[sap-no-exec-command](#) [page 278]
[sap-no-global-define](#) [page 279]
[sap-no-global-event](#) [page 280]
[sap-no-global-selection](#) [page 282]
[sap-no-global-variable](#) [page 283]
[sap-no-hardcoded-color](#) [page 284]
[sap-no-hardcoded-url](#) [page 286]
[sap-no-history-manipulation](#) [page 287]
[sap-no-jquery-device-api](#) [page 289]
[sap-no-localhost](#) [page 290]
[sap-no-location-reload](#) [page 293]
[sap-no-location-usage](#) [page 295]

7.5.3.3.3.32 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 245]

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)
[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-usage \[page 295\]](#)

7.5.3.3.33 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 \[page 245\]](#)

Related Information

[sap-cross-application-navigation \[page 247\]](#)
[sap-forbidden-window-property \[page 249\]](#)
[sap-no-navigator \[page 250\]](#)
[sap-no-override-rendering \[page 252\]](#)
[sap-no-override-storage-prototype \[page 253\]](#)
[sap-no-proprietary-browser-api \[page 254\]](#)
[sap-no-sessionstorage \[page 256\]](#)
[sap-no-ui5-prop-warning \[page 257\]](#)
[sap-no-ui5base-prop \[page 258\]](#)
[sap-timeout-usage \[page 260\]](#)
[sap-ui5-no-private-prop \[page 262\]](#)
[sap-usage-basemastercontroller \[page 264\]](#)
[sap-message-toast \[page 266\]](#)
[sap-no-absolute-component-path \[page 267\]](#)

[sap-no-br-on-return \[page 269\]](#)
[sap-no-dom-access \[page 270\]](#)
[sap-no-dom-insertion \[page 272\]](#)
[sap-no-dynamic-style-insertion \[page 273\]](#)
[sap-no-element-creation \[page 275\]](#)
[sap-no-encode-file-service \[page 276\]](#)
[sap-no-exec-command \[page 278\]](#)
[sap-no-global-define \[page 279\]](#)
[sap-no-global-event \[page 280\]](#)
[sap-no-global-selection \[page 282\]](#)
[sap-no-global-variable \[page 283\]](#)
[sap-no-hardcoded-color \[page 284\]](#)
[sap-no-hardcoded-url \[page 286\]](#)
[sap-no-history-manipulation \[page 287\]](#)
[sap-no-jquery-device-api \[page 289\]](#)
[sap-no-localhost \[page 290\]](#)
[sap-no-localstorage \[page 292\]](#)
[sap-no-location-reload \[page 293\]](#)

7.5.3.3.4 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 Cloud Platform account.

Context

i Note

You must be an administrator of your SAP Cloud Platform account.

Procedure

1. Open SAP Cloud Platform cockpit.

i Note

You can access SAP Cloud Platform cockpit from SAP Web IDE by selecting .

2. Open the SAP Web IDE page for JavaScript rules.

- a. In the SAP Cloud Platform cockpit, go to [Services](#).
 - b. Select the [SAP Web IDE](#) service.
 - c. Select [JavaScript Rules](#).
3. Select [Import](#), and upload your ESLint rules file.

7.5.3.4 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 \[page 297\]](#)

SAP Web IDE performs semantic validation on XML files.

[SAP Fiori XML Validator Rules \[page 299\]](#)

The SAP Fiori XML validator rules are used when building SAP Fiori projects.

Task overview: [Checking Code \[page 239\]](#)

Related Information

[Configure Code Checking \[page 240\]](#)

[Code Checking Annotations \[page 241\]](#)

[JavaScript Validation \[page 243\]](#)

[Validation of neo-app.json Files \[page 323\]](#)

[Using the Problems View \[page 323\]](#)

[Validation of manifest.json Files \[page 324\]](#)

7.5.3.4.1 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 \[page 297\]](#)

Related Information

[SAP Fiori XML Validator Rules \[page 299\]](#)

7.5.3.4.2 SAP Fiori XML Validator Rules

The SAP Fiori XML validator rules are used when building SAP Fiori projects.

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)

Detects buttons with text and icons.

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)

Detects lists and tables with `showNoData` attributes that are not set to `true`.

[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)

Detects duplicate icons in `IconTabBar` tags.

[DG_XML_NO_SINGLE_TAB \[page 304\]](#)

Detects a single `IconTabFilter` tag in `IconTabBar` tags.

[XML_COMMONS_USAGE \[page 306\]](#)

Detects the usage of `sap.ui.commons` objects.

[XML_DEPRECATED \[page 307\]](#)

Checks for deprecated controls.

[XML_DIALOG_IN_VIEW \[page 308\]](#)

Checks `Dialog`, `Popover`, `ResponsivePopover`, and `ActionSheet` tags in views.

[XML_FORM_USAGE \[page 309\]](#)

Checks the usage of the `sap.ui.commons.form` tags.

[XML_ICON_ACCESSIBILITY \[page 310\]](#)

Checks accessibility for icons.

[XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)

Checks accessibility for icons.

[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)

Checks accessibility for images.

[XML_LAYOUT_USAGE \[page 314\]](#)

Detects the usage of the `sap.ui.commons.layout` tags.

[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)

Checks usage of absolute `media_src` URLs.

[XML_MISSING_STABLE_ID \[page 316\]](#)

Checks for stable IDs for controls.

[XML_PAGE_ACCESSIBILITY \[page 317\]](#)

Detects missing `title` attributes.

[XML_TABLE_ACCESSIBILITY \[page 318\]](#)

Detects missing `title` attributes.

[XML_TITLE_ACCESSIBILITY \[page 319\]](#)

Detects missing title attributes.

[XML_UPLOAD_IN_VIEW \[page 320\]](#)

Checks usage of FileUpload and AddPicture tags.

[XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

Checks your setting of the serviceRefreshInterval attribute.

Parent topic: [XML Validation \[page 297\]](#)

Related Information

[XML Semantic Validation \[page 297\]](#)

7.5.3.4.2.1 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>
```

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 299]

Related Information

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
[DG_XML_NO_SINGLE_TAB \[page 304\]](#)
[XML_COMMONS_USAGE \[page 306\]](#)
[XML_DEPRECATED \[page 307\]](#)
[XML_DIALOG_IN_VIEW \[page 308\]](#)
[XML_FORM_USAGE \[page 309\]](#)
[XML_ICON_ACCESSIBILITY \[page 310\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
[XML_LAYOUT_USAGE \[page 314\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
[XML_MISSING_STABLE_ID \[page 316\]](#)
[XML_PAGE_ACCESSIBILITY \[page 317\]](#)
[XML_TABLE_ACCESSIBILITY \[page 318\]](#)
[XML_TITLE_ACCESSIBILITY \[page 319\]](#)
[XML_UPLOAD_IN_VIEW \[page 320\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.2 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)

[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)

[DG_XML_NO_SINGLE_TAB](#) [page 304]
[XML_COMMONS_USAGE](#) [page 306]
[XML_DEPRECATED](#) [page 307]
[XML_DIALOG_IN_VIEW](#) [page 308]
[XML_FORM_USAGE](#) [page 309]
[XML_ICON_ACCESSIBILITY](#) [page 310]
[XML_ICON_BUTTON_ACCESSIBILITY](#) [page 311]
[XML_IMAGE_ACCESSIBILITY](#) [page 313]
[XML_LAYOUT_USAGE](#) [page 314]
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 315]
[XML_MISSING_STABLE_ID](#) [page 316]
[XML_PAGE_ACCESSIBILITY](#) [page 317]
[XML_TABLE_ACCESSIBILITY](#) [page 318]
[XML_TITLE_ACCESSIBILITY](#) [page 319]
[XML_UPLOAD_IN_VIEW](#) [page 320]
[XML_BOOKMARK_PERFORMANCE](#) [page 321]

7.5.3.4.2.3 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 cases, please ignore the warning.

Parent topic: SAP Fiori XML Validator Rules [page 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON](#) [page 300]
[DG_XML_LIST_BASE_SHOW_NO_DATA](#) [page 301]
[DG_XML_NO_SINGLE_TAB](#) [page 304]
[XML_COMMONS_USAGE](#) [page 306]
[XML_DEPRECATED](#) [page 307]
[XML_DIALOG_IN_VIEW](#) [page 308]
[XML_FORM_USAGE](#) [page 309]
[XML_ICON_ACCESSIBILITY](#) [page 310]
[XML_ICON_BUTTON_ACCESSIBILITY](#) [page 311]
[XML_IMAGE_ACCESSIBILITY](#) [page 313]
[XML_LAYOUT_USAGE](#) [page 314]
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 315]
[XML_MISSING_STABLE_ID](#) [page 316]
[XML_PAGE_ACCESSIBILITY](#) [page 317]
[XML_TABLE_ACCESSIBILITY](#) [page 318]
[XML_TITLE_ACCESSIBILITY](#) [page 319]
[XML_UPLOAD_IN_VIEW](#) [page 320]
[XML_BOOKMARK_PERFORMANCE](#) [page 321]

7.5.3.4.2.4 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 `IIconTabBar` should have more than one `IIconTab`

Rule Details

The rule checks each `IIconTabBar` and finds all tags with just one `IIconTabFilter`.

The following patterns are considered warnings:

```
<IIconTabBar
    id="idIIconTabBar"
    select="handleIIconTabBarSelect"
    class="sapUiResponsiveContentPadding">
    <items>
        <IIconTabFilter
            showAll="true"
            count="{/ProductCollectionStats/Counts/Total}"
            text="Products"
            keys="All" />
        <IIconTabSeparator />
    </items>
</IIconTabBar>
```

False Positives

If there are multiple `IIconTabFilter` tags, and some of them are invisible, the rule does not detect a violation if just one `IIconTabFilter` is visible.

Parent topic: [SAP Fiori XML Validator Rules \[page 299\]](#)

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)

[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)

[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)

[XML_COMMONS_USAGE \[page 306\]](#)

[XML_DEPRECATED \[page 307\]](#)

[XML_DIALOG_IN_VIEW \[page 308\]](#)

[XML_FORM_USAGE \[page 309\]](#)

[XML_ICON_ACCESSIBILITY \[page 310\]](#)

[XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)

[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)

[XML_LAYOUT_USAGE \[page 314\]](#)

[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 315]
[XML_MISSING_STABLE_ID](#) [page 316]
[XML_PAGE_ACCESSIBILITY](#) [page 317]
[XML_TABLE_ACCESSIBILITY](#) [page 318]
[XML_TITLE_ACCESSIBILITY](#) [page 319]
[XML_UPLOAD_IN_VIEW](#) [page 320]
[XML_BOOKMARK_PERFORMANCE](#) [page 321]

7.5.3.4.2.5 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON](#) [page 300]
[DG_XML_LIST_BASE_SHOW_NO_DATA](#) [page 301]
[DG_XML_NO_DUPLICATE_ICONS](#) [page 303]
[DG_XML_NO_SINGLE_TAB](#) [page 304]
[XML_DEPRECATED](#) [page 307]
[XML_DIALOG_IN_VIEW](#) [page 308]
[XML_FORM_USAGE](#) [page 309]

[XML_ICON_ACCESSIBILITY](#) [page 310]
[XML_ICON_BUTTON_ACCESSIBILITY](#) [page 311]
[XML_IMAGE_ACCESSIBILITY](#) [page 313]
[XML_LAYOUT_USAGE](#) [page 314]
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 315]
[XML_MISSING_STABLE_ID](#) [page 316]
[XML_PAGE_ACCESSIBILITY](#) [page 317]
[XML_TABLE_ACCESSIBILITY](#) [page 318]
[XML_TITLE_ACCESSIBILITY](#) [page 319]
[XML_UPLOAD_IN_VIEW](#) [page 320]
[XML_BOOKMARK_PERFORMANCE](#) [page 321]

7.5.3.4.2.6 XML_DEPRECATED

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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON](#) [page 300]
[DG_XML_LIST_BASE_SHOW_NO_DATA](#) [page 301]
[DG_XML_NO_DUPLICATE_ICONS](#) [page 303]
[DG_XML_NO_SINGLE_TAB](#) [page 304]
[XML_COMMONS_USAGE](#) [page 306]
[XML_DIALOG_IN_VIEW](#) [page 308]

[XML_FORM_USAGE \[page 309\]](#)
[XML_ICON_ACCESSIBILITY \[page 310\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
[XML_LAYOUT_USAGE \[page 314\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
[XML_MISSING_STABLE_ID \[page 316\]](#)
[XML_PAGE_ACCESSIBILITY \[page 317\]](#)
[XML_TABLE_ACCESSIBILITY \[page 318\]](#)
[XML_TITLE_ACCESSIBILITY \[page 319\]](#)
[XML_UPLOAD_IN_VIEW \[page 320\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.7 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
[DG_XML_NO_SINGLE_TAB \[page 304\]](#)
[XML_COMMONS_USAGE \[page 306\]](#)
[XML_DEPRECATED \[page 307\]](#)
[XML_FORM_USAGE \[page 309\]](#)
[XML_ICON_ACCESSIBILITY \[page 310\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
[XML_LAYOUT_USAGE \[page 314\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
[XML_MISSING_STABLE_ID \[page 316\]](#)
[XML_PAGE_ACCESSIBILITY \[page 317\]](#)
[XML_TABLE_ACCESSIBILITY \[page 318\]](#)
[XML_TITLE_ACCESSIBILITY \[page 319\]](#)
[XML_UPLOAD_IN_VIEW \[page 320\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.8 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON](#) [page 300]
[DG_XML_LIST_BASE_SHOW_NO_DATA](#) [page 301]
[DG_XML_NO_DUPLICATE_ICONS](#) [page 303]
[DG_XML_NO_SINGLE_TAB](#) [page 304]
[XML_COMMONS_USAGE](#) [page 306]
[XML_DEPRECATED](#) [page 307]
[XML_DIALOG_IN_VIEW](#) [page 308]
[XML_ICON_ACCESSIBILITY](#) [page 310]
[XML_ICON_BUTTON_ACCESSIBILITY](#) [page 311]
[XML_IMAGE_ACCESSIBILITY](#) [page 313]
[XML_LAYOUT_USAGE](#) [page 314]
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 315]
[XML_MISSING_STABLE_ID](#) [page 316]
[XML_PAGE_ACCESSIBILITY](#) [page 317]
[XML_TABLE_ACCESSIBILITY](#) [page 318]
[XML_TITLE_ACCESSIBILITY](#) [page 319]
[XML_UPLOAD_IN_VIEW](#) [page 320]
[XML_BOOKMARK_PERFORMANCE](#) [page 321]

7.5.3.4.2.9 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
[DG_XML_NO_SINGLE_TAB \[page 304\]](#)
[XML_COMMONS_USAGE \[page 306\]](#)
[XML_DEPRECATED \[page 307\]](#)
[XML_DIALOG_IN_VIEW \[page 308\]](#)
[XML_FORM_USAGE \[page 309\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
[XML_LAYOUT_USAGE \[page 314\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
[XML_MISSING_STABLE_ID \[page 316\]](#)
[XML_PAGE_ACCESSIBILITY \[page 317\]](#)
[XML_TABLE_ACCESSIBILITY \[page 318\]](#)
[XML_TITLE_ACCESSIBILITY \[page 319\]](#)
[XML_UPLOAD_IN_VIEW \[page 320\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.10 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
[DG_XML_NO_SINGLE_TAB \[page 304\]](#)
[XML_COMMONS_USAGE \[page 306\]](#)
[XML_DEPRECATED \[page 307\]](#)
[XML_DIALOG_IN_VIEW \[page 308\]](#)
[XML_FORM_USAGE \[page 309\]](#)
[XML_ICON_ACCESSIBILITY \[page 310\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
[XML_LAYOUT_USAGE \[page 314\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
[XML_MISSING_STABLE_ID \[page 316\]](#)
[XML_PAGE_ACCESSIBILITY \[page 317\]](#)
[XML_TABLE_ACCESSIBILITY \[page 318\]](#)
[XML_TITLE_ACCESSIBILITY \[page 319\]](#)
[XML_UPLOAD_IN_VIEW \[page 320\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.11 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 299]

Related Information

- [DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
- [DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
- [DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
- [DG_XML_NO_SINGLE_TAB \[page 304\]](#)
- [XML_COMMONS_USAGE \[page 306\]](#)
- [XML_DEPRECATED \[page 307\]](#)
- [XML_DIALOG_IN_VIEW \[page 308\]](#)
- [XML_FORM_USAGE \[page 309\]](#)
- [XML_ICON_ACCESSIBILITY \[page 310\]](#)
- [XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
- [XML_LAYOUT_USAGE \[page 314\]](#)
- [XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
- [XML_MISSING_STABLE_ID \[page 316\]](#)
- [XML_PAGE_ACCESSIBILITY \[page 317\]](#)
- [XML_TABLE_ACCESSIBILITY \[page 318\]](#)
- [XML_TITLE_ACCESSIBILITY \[page 319\]](#)
- [XML_UPLOAD_IN_VIEW \[page 320\]](#)
- [XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.12 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 299]

Related Information

- [DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
- [DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
- [DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
- [DG_XML_NO_SINGLE_TAB \[page 304\]](#)
- [XML_COMMONS_USAGE \[page 306\]](#)
- [XML_DEPRECATED \[page 307\]](#)
- [XML_DIALOG_IN_VIEW \[page 308\]](#)
- [XML_FORM_USAGE \[page 309\]](#)
- [XML_ICON_ACCESSIBILITY \[page 310\]](#)
- [XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
- [XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
- [XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
- [XML_MISSING_STABLE_ID \[page 316\]](#)
- [XML_PAGE_ACCESSIBILITY \[page 317\]](#)
- [XML_TABLE_ACCESSIBILITY \[page 318\]](#)
- [XML_TITLE_ACCESSIBILITY \[page 319\]](#)
- [XML_UPLOAD_IN_VIEW \[page 320\]](#)
- [XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.13 XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER

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 299]

Related Information

- [DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
- [DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
- [DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
- [DG_XML_NO_SINGLE_TAB \[page 304\]](#)
- [XML_COMMONS_USAGE \[page 306\]](#)
- [XML_DEPRECATED \[page 307\]](#)
- [XML_DIALOG_IN_VIEW \[page 308\]](#)
- [XML_FORM_USAGE \[page 309\]](#)
- [XML_ICON_ACCESSIBILITY \[page 310\]](#)
- [XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
- [XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
- [XML_LAYOUT_USAGE \[page 314\]](#)
- [XML_MISSING_STABLE_ID \[page 316\]](#)
- [XML_PAGE_ACCESSIBILITY \[page 317\]](#)
- [XML_TABLE_ACCESSIBILITY \[page 318\]](#)
- [XML_TITLE_ACCESSIBILITY \[page 319\]](#)
- [XML_UPLOAD_IN_VIEW \[page 320\]](#)
- [XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.14 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON](#) [page 300]
[DG_XML_LIST_BASE_SHOW_NO_DATA](#) [page 301]
[DG_XML_NO_DUPLICATE_ICONS](#) [page 303]
[DG_XML_NO_SINGLE_TAB](#) [page 304]
[XML_COMMONS_USAGE](#) [page 306]
[XML_DEPRECATED](#) [page 307]
[XML_DIALOG_IN_VIEW](#) [page 308]
[XML_FORM_USAGE](#) [page 309]
[XML_ICON_ACCESSIBILITY](#) [page 310]
[XML_ICON_BUTTON_ACCESSIBILITY](#) [page 311]
[XML_IMAGE_ACCESSIBILITY](#) [page 313]
[XML_LAYOUT_USAGE](#) [page 314]
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 315]
[XML_PAGE_ACCESSIBILITY](#) [page 317]
[XML_TABLE_ACCESSIBILITY](#) [page 318]
[XML_TITLE_ACCESSIBILITY](#) [page 319]
[XML_UPLOAD_IN_VIEW](#) [page 320]
[XML_BOOKMARK_PERFORMANCE](#) [page 321]

7.5.3.4.2.15 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON](#) [page 300]
[DG_XML_LIST_BASE_SHOW_NO_DATA](#) [page 301]
[DG_XML_NO_DUPLICATE_ICONS](#) [page 303]
[DG_XML_NO_SINGLE_TAB](#) [page 304]
[XML_COMMONS_USAGE](#) [page 306]
[XML_DEPRECATED](#) [page 307]
[XML_DIALOG_IN_VIEW](#) [page 308]
[XML_FORM_USAGE](#) [page 309]
[XML_ICON_ACCESSIBILITY](#) [page 310]
[XML_ICON_BUTTON_ACCESSIBILITY](#) [page 311]
[XML_IMAGE_ACCESSIBILITY](#) [page 313]
[XML_LAYOUT_USAGE](#) [page 314]
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 315]
[XML_MISSING_STABLE_ID](#) [page 316]
[XML_TABLE_ACCESSIBILITY](#) [page 318]
[XML_TITLE_ACCESSIBILITY](#) [page 319]
[XML_UPLOAD_IN_VIEW](#) [page 320]
[XML_BOOKMARK_PERFORMANCE](#) [page 321]

7.5.3.4.2.16 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON](#) [page 300]
[DG_XML_LIST_BASE_SHOW_NO_DATA](#) [page 301]
[DG_XML_NO_DUPLICATE_ICONS](#) [page 303]
[DG_XML_NO_SINGLE_TAB](#) [page 304]
[XML_COMMONS_USAGE](#) [page 306]
[XML_DEPRECATED](#) [page 307]
[XML_DIALOG_IN_VIEW](#) [page 308]
[XML_FORM_USAGE](#) [page 309]
[XML_ICON_ACCESSIBILITY](#) [page 310]
[XML_ICON_BUTTON_ACCESSIBILITY](#) [page 311]
[XML_IMAGE_ACCESSIBILITY](#) [page 313]
[XML_LAYOUT_USAGE](#) [page 314]
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER](#) [page 315]
[XML_MISSING_STABLE_ID](#) [page 316]
[XML_PAGE_ACCESSIBILITY](#) [page 317]
[XML_TITLE_ACCESSIBILITY](#) [page 319]
[XML_UPLOAD_IN_VIEW](#) [page 320]
[XML_BOOKMARK_PERFORMANCE](#) [page 321]

7.5.3.4.2.17 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
[DG_XML_NO_SINGLE_TAB \[page 304\]](#)
[XML_COMMONS_USAGE \[page 306\]](#)
[XML_DEPRECATED \[page 307\]](#)
[XML_DIALOG_IN_VIEW \[page 308\]](#)
[XML_FORM_USAGE \[page 309\]](#)
[XML_ICON_ACCESSIBILITY \[page 310\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
[XML_LAYOUT_USAGE \[page 314\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
[XML_MISSING_STABLE_ID \[page 316\]](#)
[XML_PAGE_ACCESSIBILITY \[page 317\]](#)
[XML_TABLE_ACCESSIBILITY \[page 318\]](#)
[XML_UPLOAD_IN_VIEW \[page 320\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.18 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 299]

Related Information

[DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
[DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
[DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
[DG_XML_NO_SINGLE_TAB \[page 304\]](#)
[XML_COMMONS_USAGE \[page 306\]](#)
[XML_DEPRECATED \[page 307\]](#)
[XML_DIALOG_IN_VIEW \[page 308\]](#)
[XML_FORM_USAGE \[page 309\]](#)
[XML_ICON_ACCESSIBILITY \[page 310\]](#)
[XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
[XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
[XML_LAYOUT_USAGE \[page 314\]](#)
[XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
[XML_MISSING_STABLE_ID \[page 316\]](#)
[XML_PAGE_ACCESSIBILITY \[page 317\]](#)
[XML_TABLE_ACCESSIBILITY \[page 318\]](#)
[XML_TITLE_ACCESSIBILITY \[page 319\]](#)
[XML_BOOKMARK_PERFORMANCE \[page 321\]](#)

7.5.3.4.2.19 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.common" 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" />
```

Parent topic: SAP Fiori XML Validator Rules [page 299]

Related Information

- [DG_XML_FOOTER_BUTTON_TEXT_ICON \[page 300\]](#)
- [DG_XML_LIST_BASE_SHOW_NO_DATA \[page 301\]](#)
- [DG_XML_NO_DUPLICATE_ICONS \[page 303\]](#)
- [DG_XML_NO_SINGLE_TAB \[page 304\]](#)
- [XML_COMMONS_USAGE \[page 306\]](#)
- [XML_DEPRECATED \[page 307\]](#)
- [XML_DIALOG_IN_VIEW \[page 308\]](#)
- [XML_FORM_USAGE \[page 309\]](#)
- [XML_ICON_ACCESSIBILITY \[page 310\]](#)
- [XML_ICON_BUTTON_ACCESSIBILITY \[page 311\]](#)
- [XML_IMAGE_ACCESSIBILITY \[page 313\]](#)
- [XML_LAYOUT_USAGE \[page 314\]](#)
- [XML_METADATA_MEDIA_SRC_WITHOUT_FORMATTER \[page 315\]](#)
- [XML_MISSING_STABLE_ID \[page 316\]](#)
- [XML_PAGE_ACCESSIBILITY \[page 317\]](#)
- [XML_TABLE_ACCESSIBILITY \[page 318\]](#)
- [XML_TITLE_ACCESSIBILITY \[page 319\]](#)
- [XML_UPLOAD_IN_VIEW \[page 320\]](#)

7.5.3.5 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 \[page 239\]](#)

Related Information

[Configure Code Checking \[page 240\]](#)

[Code Checking Annotations \[page 241\]](#)

[JavaScript Validation \[page 243\]](#)

[XML Validation \[page 297\]](#)

[Using the Problems View \[page 323\]](#)

[Validation of manifest.json Files \[page 324\]](#)

7.5.3.6 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.

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 \[page 239\]](#)

Related Information

[Configure Code Checking \[page 240\]](#)

[Code Checking Annotations \[page 241\]](#)

[JavaScript Validation \[page 243\]](#)

[XML Validation \[page 297\]](#)

[Validation of `neo-app.json` Files \[page 323\]](#)

[Validation of `manifest.json` Files \[page 324\]](#)

7.5.3.7 Validation of `manifest.json` Files

Application descriptor files (`manifest.json`) for SAP Fiori projects are validated on opening in the code editor.

When you open a `manifest.json` file in the code editor for an SAP Fiori project, the following validation is performed:

- Validation of the json syntax. Code issues are displayed as annotations as for JavaScript file code checking.
- If the `manifest.json` file has no syntax issues, 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 \[page 239\]](#)

Related Information

[Configure Code Checking \[page 240\]](#)

[Code Checking Annotations \[page 241\]](#)

[JavaScript Validation \[page 243\]](#)

[XML Validation \[page 297\]](#)

[Validation of neo-app.json Files \[page 323\]](#)

[Using the Problems View \[page 323\]](#)

7.5.4 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**

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**

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.

- **Overview Page**

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](#).

7.5.4.1 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 \[page 156\]](#).
- 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 Cloud Platform, so that the OData services are available in SAP Web IDE.
- For more information, see [Destinations](#) and [SAP Cloud Platform connector](#).

7.5.4.2 Create the List Report and Object Page Application

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> .

Step	Action
2. Basic Information	<p>The following fields are mandatory:</p> <ul style="list-style-type: none"> ○ <i>Project Name</i> ○ <i>Title</i> ○ <i>Application Component Hierarchy</i> <p>Choose <i>Next</i>.</p>
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 option 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.

7.5.4.3 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 \[page 328\]](#)

7.5.4.3.1 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 .
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. (Optional) Supports multiple views in one card.
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. (Optional) Supports multiple views in one card.
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. (Optional) Supports multiple views in one card.

i Note

Overview Page lets you to configure view switch for List, Table, and Analytic Chart cards. Selecting the checkbox *Select to enable view switch for this card* lets you configure multiple views and apply different filtering and sorting options in the card.

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*.

7.5.5 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.

i 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.

7.5.6 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 \[page 158\]](#).

```
{
    "path": "/sap/opu/odata",
    "target": {
        "type": "destination",
        "name": "myRemoteDestination",
        "entryPath": "/sap/opu/odata"
    },
    "description": "Target OData system"
}
```

7.6 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](#)

[Create Tests \[page 332\]](#)

You can create tests in your project using a wizard.

7.6.1 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*  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.

7.7 Layout Editor

Display the content of an XML view in the SAP Web IDE layout editor to see it in a way that closely corresponds to how it will appear in your finished application.

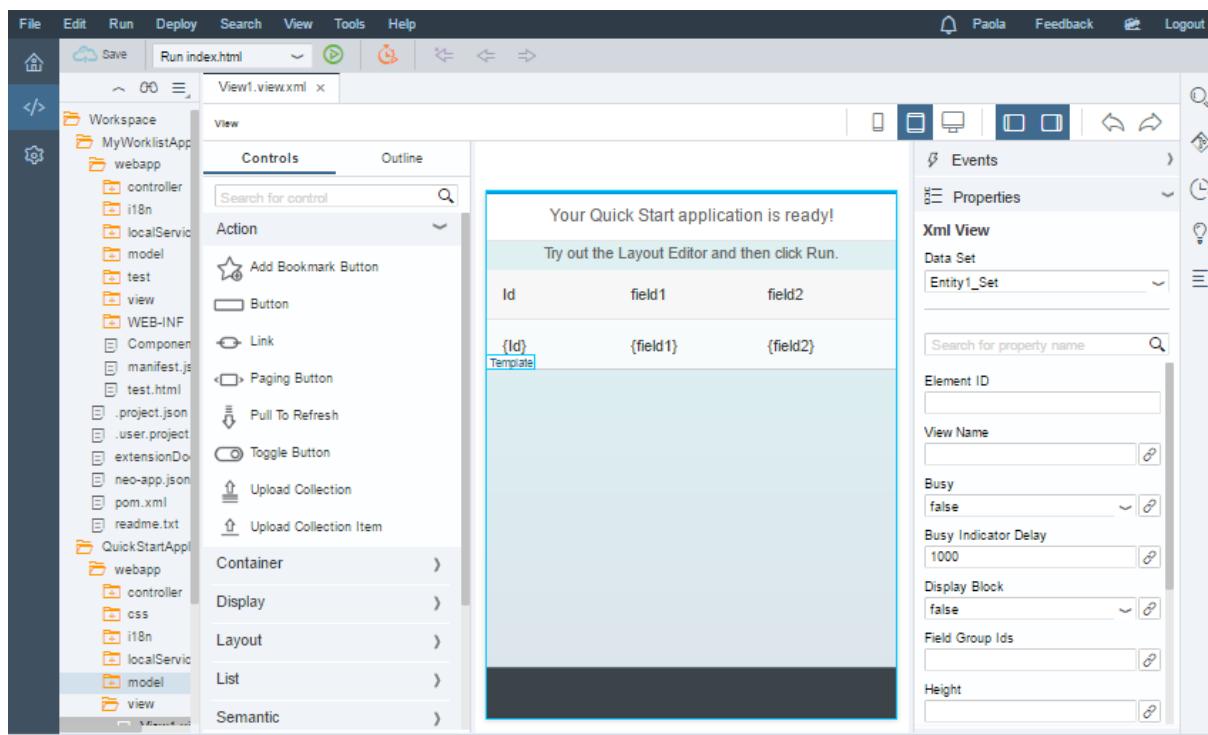
i Note

The layout editor is not supported in the Safari browser.

For more information on known issues regarding the layout editor, see [Known Issues \[page 510\]](#).

Layout Editor Landscape

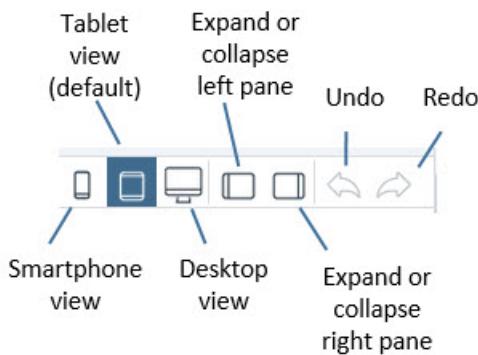
The layout editor is composed of a canvas, a pane on the left that includes the *Controls* and *Outline* tabs, and a pane on the right that includes the *Events* and *Properties* panes.



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.

i 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 \[page 341\]](#).

Controls	Outline
<input type="text" value="Search for control"/>	
Action	›
Container	›
Display	›
Layout	›
List	›
Tile	›
User Input	›

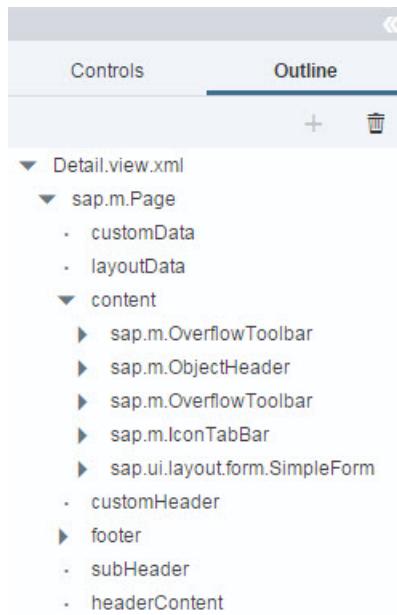
You can find the list of available controls in [SAPUI5 Controls Supported in the Layout Editor \[page 349\]](#).

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.

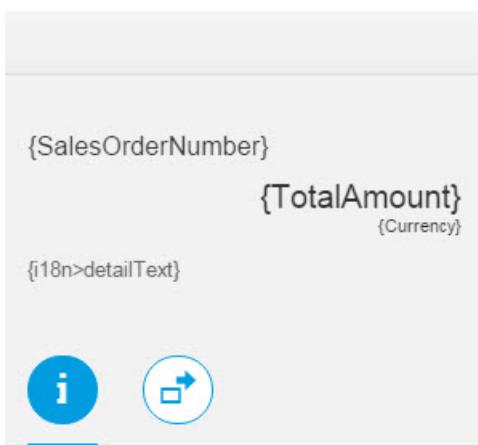
For more information, see [Add Controls from the Outline Tab \[page 339\]](#).



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:

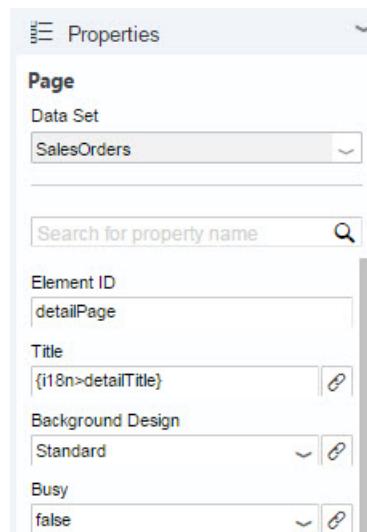
Events Pane

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 \[page 343\]](#), [Bind Data to a Simple Control \[page 344\]](#), and [Bind Data to an Aggregate-Type Control \[page 345\]](#).



i 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](#) on SAP Help Portal at <http://help.sap.com/nw-uiaddon>. Under *Application Help*, open *SAP Library*, and search for *deprecation*.

[Work with the Layout Editor \[page 338\]](#)

An overview of the steps required to edit a project using the layout editor.

[Create a Quick Start Application with the Layout Editor \[page 348\]](#)

Quickly create a new application using the layout editor.

[SAPUI5 Controls Supported in the Layout Editor \[page 349\]](#)

Provides a list of SAPUI5 controls that are supported in the layout editor.

[Try It: Build an Application with the Layout Editor \[page 362\]](#)

Get an overview of the features that are available with the layout editor by following this tutorial for building an application.

Related Information

[Work with the Layout Editor \[page 338\]](#)

[SAPUI5 Controls Supported in the Layout Editor \[page 349\]](#)

7.7.1 Work 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 views of your application as follows:
 - Add controls to your view using drag and drop functionality.
 - Delete controls from your view.
 - Rearrange controls in your view using drag and drop functionality.
 - Use the keyboard to navigate within the canvas. Click the same control twice 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 \[page 339\]](#)

You can add controls to the canvas from the *Outline* tab.

[Delete Controls from the Outline Tab \[page 339\]](#)

You can remove controls from the *Outline* tab.

[Create a New Function \[page 340\]](#)

You can create a new function for a controller from the *Events* pane.

[Add Controls from the Controls Tab \[page 341\]](#)

Add controls to the canvas by using drag and drop functionality.

[Keyboard Support \[page 342\]](#)

Use the keyboard to move selected controls or navigate within the view that you opened with the layout editor.

[Layout Editor Binding Capabilities \[page 343\]](#)

In the layout editor, you can bind properties of controls or control aggregations to data fields, to i18n models, and to label annotations.

7.7.1.1 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  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: [Work with the Layout Editor \[page 338\]](#)

Related Information

[Delete Controls from the Outline Tab \[page 339\]](#)

[Create a New Function \[page 340\]](#)

[Add Controls from the Controls Tab \[page 341\]](#)

[Keyboard Support \[page 342\]](#)

[Layout Editor Binding Capabilities \[page 343\]](#)

[Delete Controls from the Outline Tab \[page 339\]](#)

[SAPUI5 Controls Supported in the Layout Editor \[page 349\]](#)

7.7.1.2 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  button.

The control is removed from the view.

Task overview: [Work with the Layout Editor \[page 338\]](#)

Related Information

[Add Controls from the Outline Tab \[page 339\]](#)

[Create a New Function \[page 340\]](#)

[Add Controls from the Controls Tab \[page 341\]](#)

[Keyboard Support \[page 342\]](#)

[Layout Editor Binding Capabilities \[page 343\]](#)

[Add Controls from the Outline Tab \[page 339\]](#)

[SAPUI5 Controls Supported in the Layout Editor \[page 349\]](#)

7.7.1.3 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: [Work with the Layout Editor \[page 338\]](#)

Related Information

[Add Controls from the Outline Tab \[page 339\]](#)

[Delete Controls from the Outline Tab \[page 339\]](#)
[Add Controls from the Controls Tab \[page 341\]](#)
[Keyboard Support \[page 342\]](#)
[Layout Editor Binding Capabilities \[page 343\]](#)
[Layout Editor \[page 332\]](#)

7.7.1.4 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.

• Note

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: [Work with the Layout Editor \[page 338\]](#)

Related Information

[Add Controls from the Outline Tab \[page 339\]](#)
[Delete Controls from the Outline Tab \[page 339\]](#)
[Create a New Function \[page 340\]](#)
[Keyboard Support \[page 342\]](#)
[Layout Editor Binding Capabilities \[page 343\]](#)

7.7.1.5 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]+ control**: 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: [Work with the Layout Editor \[page 338\]](#)

Related Information

[Add Controls from the Outline Tab \[page 339\]](#)

[Delete Controls from the Outline Tab \[page 339\]](#)

[Create a New Function \[page 340\]](#)

[Add Controls from the Controls Tab \[page 341\]](#)

[Layout Editor Binding Capabilities \[page 343\]](#)

7.7.1.6 Layout Editor Binding Capabilities

In the layout editor, you can bind properties of controls or control aggregations to data fields, to i18n models, and to label annotations.

Parent topic: [Work with the Layout Editor \[page 338\]](#)

Related Information

[Add Controls from the Outline Tab \[page 339\]](#)

[Delete Controls from the Outline Tab \[page 339\]](#)

[Create a New Function \[page 340\]](#)

[Add Controls from the Controls Tab \[page 341\]](#)

[Keyboard Support \[page 342\]](#)

7.7.1.6.1 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.

[Bind Data to a Simple Control \[page 344\]](#)

You can bind data to a simple control.

[Bind Data to an Aggregate-Type Control \[page 345\]](#)

You can bind data to an aggregate-type control, which creates a template.

[Unbind Data from a Simple Control Property \[page 346\]](#)

You can unbind data from a simple control property.

[Unbind Data from a Template \[page 346\]](#)

You can unbind data from a template (aggregate-type control).

Related Information

[SAPUI5 API Reference](#)

7.7.1.6.1.1 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 \[page 343\]](#)

Related Information

[Bind Data to an Aggregate-Type Control \[page 345\]](#)

[Unbind Data from a Simple Control Property \[page 346\]](#)

[Unbind Data from a Template \[page 346\]](#)

7.7.1.6.1.2 Bind Data to an Aggregate-Type Control

You can bind data to an aggregate-type control, which creates a template.

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 \[page 343\]](#)

Related Information

- [Bind Data to a Simple Control \[page 344\]](#)
- [Unbind Data from a Simple Control Property \[page 346\]](#)
- [Unbind Data from a Template \[page 346\]](#)
- [Aggregation Binding](#)

7.7.1.6.1.3 Unbind Data from a Simple Control Property

You can unbind data from a simple control property.

Procedure

1. In the canvas, select the control whose data you want to unbind.
2. In the *Properties* pane, do one of the following:
 - In the property field or dropdown list, delete the string.
 - Click the *Data Binding*  button to open the the *Data Binding* dialog box. Click *Clear* and then click *OK*.

Task overview: [Binding Data \[page 343\]](#)

Related Information

- [Bind Data to a Simple Control \[page 344\]](#)
- [Bind Data to an Aggregate-Type Control \[page 345\]](#)
- [Unbind Data from a Template \[page 346\]](#)

7.7.1.6.1.4 Unbind Data from a Template

You can unbind data from a template (aggregate-type control).

Procedure

1. In the canvas, select the aggregate-type control whose data you want to unbind.

- In the properties pane to the right of the canvas, do one of the following:
 - Clear the *Set as template* checkbox.
 - In the dropdown list of the data set, select *Unbind*.

Task overview: [Binding Data \[page 343\]](#)

Related Information

[Bind Data to a Simple Control \[page 344\]](#)

[Bind Data to an Aggregate-Type Control \[page 345\]](#)

[Unbind Data from a Simple Control Property \[page 346\]](#)

7.7.1.6.2 Bind to the i18n Model

You can bind a control property to the i18n model or create a new i18n entry.

Procedure

- On the canvas, select the desired control for which you want to define i18n model binding.
- 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:
 - 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.
 - 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 i18n entry.

7.7.1.6.3 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 {...}.

7.7.2 Create 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* .

- 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.

- 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.
- In the properties pane to the left of the canvas, bind data to each control as required.

Related Information

[Layout Editor \[page 332\]](#)

[Add Controls from the Outline Tab \[page 339\]](#)

[Binding Data \[page 343\]](#)

7.7.3 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 \[page 339\]](#).

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.

SAPUI5 Control Name	Description
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. App provides certain header tags to the HTML page that are relevant for mobile apps.
Bar sap.m.Bar	Centers a control like a title while having other controls on its left and right.
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 DatePickers.
Calendar Legend sap.ui.unified.CalendarLegend	Legend for the Calendar control. Displays special date colors with their corresponding description.
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.
Column sap.m.Column	Allows definition of column-specific properties that are applied when rendering a List 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 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 List Item sap.m.CustomListItem	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>i Note</p> <p>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>
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.
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.
Flex Box sap.m.FlexBox	Builds the container for a flexible box layout.
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.
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.
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> .

SAPUI5 Control Name	Description
HBox	Builds the container for a horizontal flexible box layout.
sap.m.HBox	
Horizontal Layout	Provides support for horizontal alignment of controls.
sap.ui.layout.HorizontalLayout	
Icon	Uses an embedded font instead of a pixel image.
sap.ui.core.Icon	
Icon Tab Bar	Represents a collection of tabs with associated content.
sap.m.IconTabBar	
Icon Tab Filter	Represents a selectable item inside an Icon Tab Bar control.
sap.m.IconTabFilter	
Icon Tab Header	Displays a number of Icon Tab Filter and Icon Tab Separator controls.
sap.m.IconTabHeader	
Icon Tab Separator	Icon used to separate two Icon Tab Filter controls.
sap.m.IconTabSeparator	
Image	Wrapper around the IMG tag.
sap.m.Image	
Input	Allows users to input data.
sap.m.Input	
Input List Item	List item used for a label and an input field.
sap.m.InputListItem	
Invisible Text	Used to bring hidden texts to the UI for screen reader support.
sap.ui.core.InvisibleText	
Item	Control base type.
sap.ui.core.Item	
Label	Used in SAPUI5 mobile applications to provide label text for other controls.
sap.m.Label	
Link	Used to trigger actions or to navigate to other applications or web pages.
sap.m.Link	

SAPUI5 Control Name	Description
List	Provides a container for all types of list items.
sap.m.List	
List Item	Used in lists or list-like controls, such as <code>DropdownBox</code> .
sap.ui.core.ListItem	
Master Page	An <code>sap.m.semantic.SemanticPage</code> control that supports certain semantic
sap.m.semantic.MasterPage	buttons that have default semantic-specific properties and are eligible for content aggregation.
Message Strip	Allows the embedding of application-related messages in the application.
sap.m.MessageStrip	
Nav Container	Handles hierarchical navigation between Page controls or other fullscreen controls.
sap.m.NavContainer	
Object Attribute	Displays a text field that can be normal or active.
sap.m.ObjectAttribute	
Object Header	Allows the user to easily identify a special object.
sap.m.ObjectHeader	
Object Identifier	Display control that allows the user to easily identify a specific object.
sap.m.ObjectIdentifier	
Object List Item	Display control that provides summary information about an object as an item in a list.
sap.m.ObjectListIem	
Object Number	Displays number and number unit properties for an object.
sap.m.ObjectNumber	
Object Status	Status information that can be either text with a value state, or an icon.
sap.m.ObjectStatus	
Overflow Toolbar	Container based on <code>sap.m.Toolbar</code> that provides overflow when its content does
sap.m.OverflowToolbar	not fit in the visible area.
Page	Basic container for a mobile application screen.
sap.m.Page	
Paging Button	Allows users to navigate between items and entities.
sap.m.PagingButton	

SAPUI5 Control Name	Description
Panel	Container for controls that has a header and content.
sap.m.Panel	
Progress Indicator	Shows the progress of a process in a graphical way.
sap.m.ProgressIndicator	
Pull To Refresh	Triggers the refresh event.
sap.m.PullToRefresh	
Radio Button	Control similar to CheckBox, but it allows the user to choose only one of a predefined set of options.
sap.m.RadioButton	
Radio Button Group	Used as a wrapper for a group of sap.m.RadioButton controls, which then can be used as a single UI element.
sap.m.RadioButtonGroup	
Rating Indicator	Used to rate content.
sap.m.RatingIndicator	
Search Field	Allows users to input a search string.
sap.m.SearchField	
Segmented Button	Horizontal control made of multiple buttons, which can display a title or an image.
sap.m.SegmentedButton	
Select	Provides a list of items that allows the user to select an item.
sap.m.Select	
Select List	Displays a list of items that allows the user to select an item.
sap.m.SelectList	
Semantic Page	An enhanced sap.m.Page control that can contain controls with semantic meaning.
sap.m.semantic.SemanticPage	Content specified in sap.m.semantic.SemanticPage 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	An sap.m.semantic.SemanticPage control that supports a Share menu in the footer.
sap.m.semantic.ShareMenuPage	
Simple Form	Provides an API for creating simple forms. Inside a SimpleForm control, a Form control is created along with its FormContainers control and FormElements control, but the complexity in the API is removed.
sap.ui.layout.form.SimpleForm	

SAPUI5 Control Name	Description
Slider sap.m.Slider	User interface control that allows the user to adjust values within a specified numerical range.
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 Tile 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.
Title sap.ui.core.Title	Used to aggregate other controls.
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.
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.

SAPUI5 Control Name	Description
Upload Collection Item	Provides information about uploaded files.
sap.m.UploadCollectionItem	
VBox	Builds the container for a vertical flexible box layout.
sap.m.VBox	
Vertical Layout	Layout in which the content controls are rendered one below the other.
sap.ui.layout.VerticalLayout	

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
sap.m.semantic.AddAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.CancelAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.DiscussInJamAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage.
sap.m.semantic.EditAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.FavoriteAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.FilterAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.

SAPUI5 Control Name	Description
sap.m.semantic.FlagAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.ForwardAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.GroupAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.MainAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.MessagesIndicator	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.MultiSelectAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.NegativeAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.OpenInAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.PositiveAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.PrintAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.SaveAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.SendEmailAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.

SAPUI5 Control Name	Description
sap.m.semantic.SendMessageAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage.
sap.m.semantic.ShareInJamAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
sap.m.semantic.SortAction	Has default semantic-specific properties and is eligible to be included in the aggregation content of an sap.m.semantic.SemanticPage control.
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.
sap.suite.ui.commons.ComparisonChart	Displays a comparison chart.
sap.suite.ui.commons.ComparisonData	Comparison tile value holder.
sap.suite.ui.commons.CountingNavigationItem	Extends the sap.ui.ux3.NavigationItem 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.
sap.suite.ui.commons.DateRangeScroller	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.
sap.suite.ui.commons.DateRangeSliderInternal	Provides the user with a RangeSlider control that is optimized for use with dates.
sap.suite.ui.commons.DeltaMicroChart	Displays a delta of two values as a chart.
sap.suite.ui.commons.DynamicContainer	Displays multiple GenericTile controls as changing slides.
sap.suite.ui.commons.FacetOverview	Used in UnifiedThingInspector to display a preview of facet content.
sap.suite.ui.commons.GenericTile	Tile control that displays a title, description, and customizable main area.

SAPUI5 Control Name	Description
sap.suite.ui.commons.HarveyBallMicroChart	Chart that shows a comparative part to a total.
sap.suite.ui.commons.HarveyBallMicroChartItem	Configuration of a slice on a pie chart.
sap.suite.ui.commons.HeaderCell	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 HeaderCellItem.
sap.suite.ui.commons.HeaderCellItem	Object that contains an instance of a control and information about its height. It should be used inside sap.suite.ui.commons.HeaderCell.
sap.suite.ui.commons.HeaderContainer	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.
sap.suite.ui.commons.JamContent	Displays SAP Jam content text, subheader, and numeric value in a tile.
sap.suite.ui.commons.KpiTile	Used in UnifiedThingInspector 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 NoteTaker 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.

SAPUI5 Control Name	Description
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.
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.

SAPUI5 Control Name	Description
sap.ui.commonsTreeNode	Tree node element.
sap.ui.layout.FixFlex	Builds the container for a layout with a fixed and a flexible part.
sap.ui.layout.form.Form	Structured into FormContainer controls, each of which consists of FormElement controls.
sap.ui.layout.form.FormContainer	Group inside a Form.
sap.ui.layout.form.FormElement	Row in a FormContainer control.
sap.ui.layout.form.FormLayout	Base layout used to render a Form control.
sap.ui.layout.form.GridLayout	Renders a Form control using an HTML table-based grid.
sap.ui.layout.form.ResponsiveGridLayout	Renders a Form control using a responsive grid.
sap.ui.layout.form.ResponsiveLayout	Renders a Form control with a responsive layout.
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>
sap.ui.unified.FileUploaderParameter	Represents a parameter for the FileUploader, 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 sap.ui.unified.Shell 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 Feed control.
sap.ui.ux3.Feeder	Lean common feed, or a comment feed, with a text commit function.

SAPUI5 Control Name	Description
sap.uxap.ObjectPageHeader	Static part of an Object page header.
sap.uxap.ObjectPageHeaderContent	Dynamic part of an Object page header.

7.7.4 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 \[page 362\]](#)

[Create an OData Model File \[page 363\]](#)

[Create a Project for Your New Application \[page 364\]](#)

[Add Controls to Your New Application \[page 365\]](#)

7.7.4.1 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 \[page 363\]](#).
2. You have created a new project from a template according to the instructions in the topic [Create a Project for Your New Application \[page 364\]](#).

Related Information

[Create an OData Model File \[page 363\]](#)

[Create a Project for Your New Application \[page 364\]](#)

[Add Controls to Your New Application \[page 365\]](#)

7.7.4.1.1 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`.

7.7.4.1.2 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.
 4. 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.

5. 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 \[page 363\]](#). Choose the *Next* button.
6. 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>

Field or Dropdown List	Value
	<p>i Note</p> <p>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.</p>

<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>

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>

7. Choose the *Next* button.
8. In the *Confirmation* wizard step, choose the *Finish* button.

Your new project is now created in your workspace.

7.7.4.2 Add Controls to Your New Application

Steps for adding controls to your new sales order tracking application.

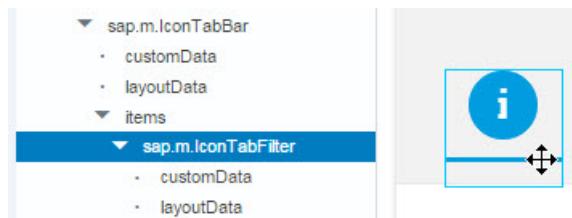
Procedure

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 \[page 364\]](#), then expand the `view` folder and right-click the `Detail.view.xml` view.
2. From the context menu, choose  *Open With 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:

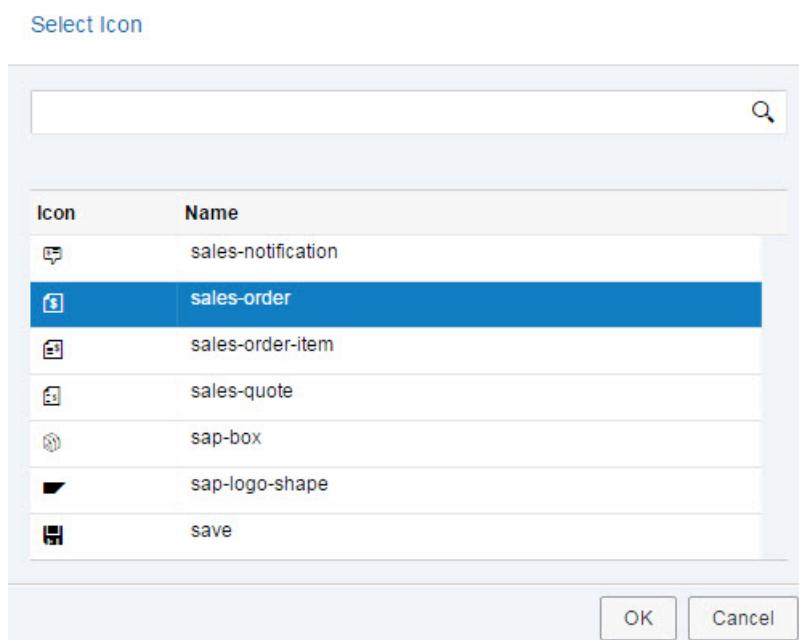


- a. On the canvas, select the first *Icon Tab Filter* control on the left side of the *Icon Tab Bar* control.

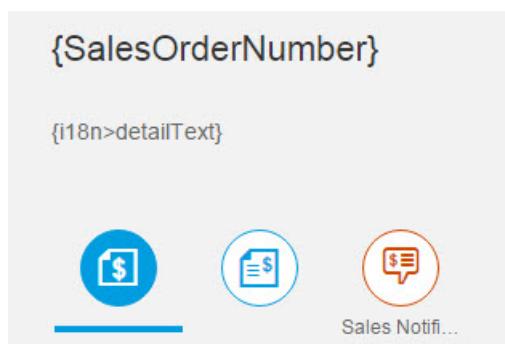
In the *Outline* tab on the left side of the canvas, *sap.m.IconTabFilter* is selected.



- 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.

My Sales Notifications

Sales Order N...:	<input type="text"/>	
Label:	<input type="text"/>	Label
Label 2:	<input type="text"/>	

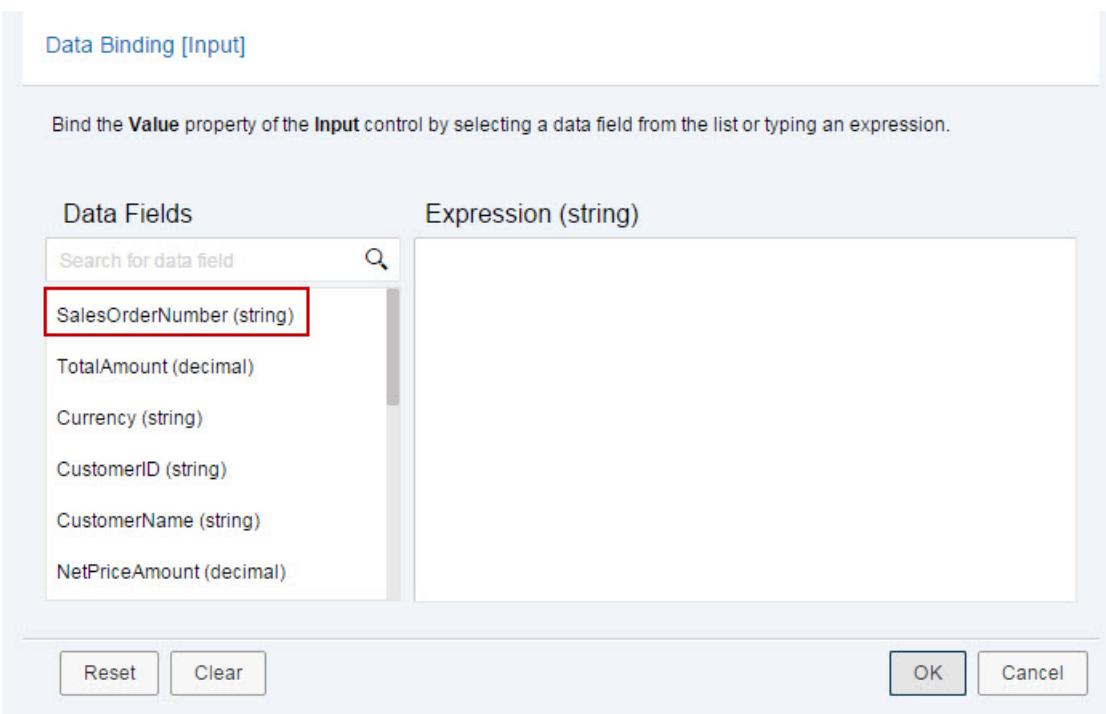
- g. In the canvas, in the *Simple Form* control, select the label that you just created and then in the properties pane to the right of the canvas, change its text to **Order Date**.
- h. Change the text of the *Label 2* control of the third input field in the same way to **Status**.

My Sales Notifications

Sales Order Number:	<input type="text"/> {SalesOrderNumber}
Order Date:	<input type="text"/> {OrderDate}
Status:	<input type="text"/> {Status}

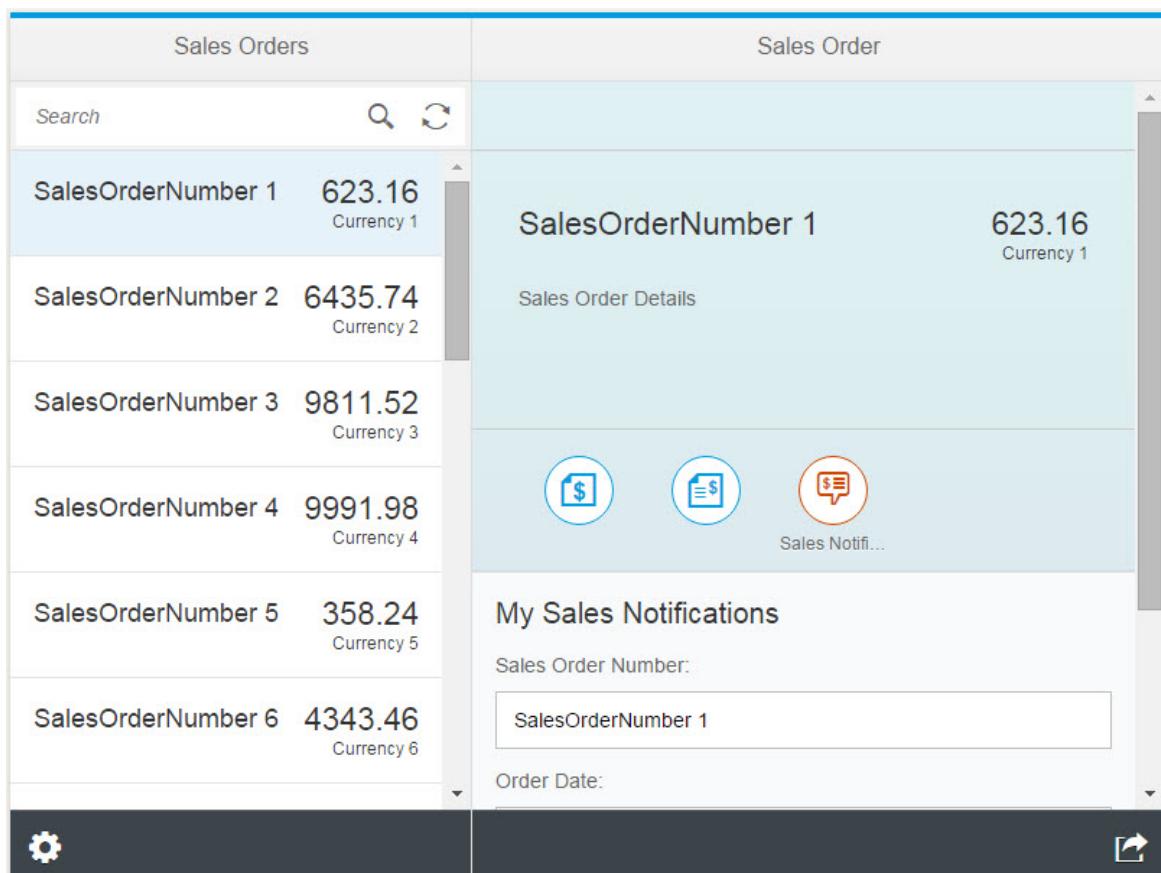
7. Bind the *Value* property of the input fields to elements from the OData service that you created:

- 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.



7.8 Annotation Modeler

SAP Web IDE 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 back-end sources 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 \[page 374\]](#).

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 \[page 194\]](#).

Use UI Adaptation 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 [UI Adaptation Editor \[page 395\]](#).

For more information about annotation modeler, watch the following video:

[Getting started with the SAP Web IDE annotation modeler](#) 

Related Information

[Architecture \[page 370\]](#)

[User Interface \[page 372\]](#)

[Prerequisites for Working with Annotation Modeler \[page 374\]](#)

[Adding Local Annotation Files to the Project \[page 376\]](#)

[Working with Annotation Modeler \[page 377\]](#)

7.8.1 Architecture

Infrastructure

Annotation modeler is embedded in the SAP Web IDE infrastructure and can be used directly in SAP Web IDE. For a detailed infrastructure overview, see [Overview \[page 150\]](#).

Supported OData Vocabularies

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.

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.

Annotation modeler supports only standard vocabularies. The following vocabularies are supported:

- **OData.org:**
 - [Aggregation](#)
 - [Capabilities](#)
 - [Core](#)
 - [Measures](#)
 - [Validation](#)
 - [Authorization](#)
- **SAP:**
 - [UI](#)
 - [Common](#)
 - [Communication](#)

i Note

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

Annotation modeler supports applying annotations to the following targets:

- Entity Type
- Entity Type Property
- Entity Set
- Function Import
- Function Import Parameters

Annotation modeler also allows you to annotate the collection records and property values of the annotations.

Limitations

- Annotation modeler only supports the services of OData version 2.0.

7.8.2 User Interface

The screenshot shows the SAP Web IDE Annotation Modeler interface. The title bar says 'annotations.xml'. The main area has a header 'Annotation Structure' and 'OData Data Source: mainService'. Below is a 'Select Targets' button and a table with columns: Node, Edit Qualifier, Key Information, Expression Type, Value, and Actions. The table data is as follows:

Node	Edit Qualifier	Key Information	Expression Type	Value	Actions
Entity Types					
SEPMRA_C_PD_ProductType					
Local Annotations		Source: localAnnotations			+ Edit
UI Facets					+ Edit
UI CollectionFacet		ID: GeneralInformation, Label: General Information			+ Edit
ID			String	GeneralInformation	+ Edit
Label			String (I18n)	General Information	+ Edit
*Facets					+ Edit
UI ReferenceFacet		Label: General Information, Target: {@UI.Identificati...			+ Edit
UI ReferenceFacet		Label: Second Facet, Target: {to_ProductSalesData...			+ Edit
External Annotations		Source: SEPMRA_PROD_MAN_ANNO_MDL			
External Annotations		Source: mainService			
Properties					
Product					
Local Annotations		Source: localAnnotations			+ Edit
External Annotations		Source: SEPMRA_PROD_MAN_ANNO_MDL			
Price					
Currency					

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 \[page 390\]](#).

- You can edit *Local Annotations* from the currently open file in annotation modeler. See [Edit Annotations \[page 379\]](#). 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 \[page 386\]](#).

- 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 \[page 387\]](#).

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 \[page 392\]](#).

- 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 \[page 388\]](#).
- You can concatenate several values for all terms containing the *Value* property. See [Concatenate Values \[page 389\]](#).
- You can insert parameters in any part of a URL. See [Use Parameters in URLs \[page 389\]](#).

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 \[page 381\]](#).

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 \[page 390\]](#).

→ 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 \[page 163\]](#).

7.8.3 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 a single */edmx:DataServices/Schema* definition. 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 file cannot be loaded. In this case, annotation modeler does not open. See [Troubleshooting \[page 393\]](#).

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 \[page 194\]](#).

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 \[page 377\]](#) and [Import Existing Annotation Files \[page 376\]](#).

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 \[page 377\]](#)

7.8.4 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 \[page 376\]](#).
- Create a new annotation file. See [Create New Annotation Files \[page 377\]](#).
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 \[page 376\]](#)

[Create New Annotation Files \[page 377\]](#)

7.8.4.1 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*  *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 `annotations` folder and is available for editing if the prerequisites are met. See [Prerequisites for Working with Annotation Modeler \[page 374\]](#).

7.8.4.2 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.

7.8.5 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 \[page 374\]](#) 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 \[page 378\]](#).
- You can preview the annotated data if your project supports running the application from the workspace. See [Run Applications from the Workspace \[page 433\]](#).

Related Information

[Open Local Annotation Files in Annotation Modeler \[page 378\]](#)

[Edit Annotations \[page 379\]](#)

[Validation of the Annotation File \[page 391\]](#)

7.8.5.1 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 \[page 163\]](#).

Open Annotation Modeler from the Context Menu of the Annotation File

1. Right-click the local annotation file of your project.
2. Choose  [Open With > Annotation Modeler](#).

7.8.5.2 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 \[page 390\]](#).

⚠ 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](#).

Procedure

- 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 \[page 380\]](#).

By default, the mandatory and the already defined optional properties are displayed under this term.
- In the *Actions* column, choose the *Add* icon and select the annotation properties to be defined.
- 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.

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.

- 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.

→ Tip

To find out more about how facets annotations are used in applications based on the ObjectPage template, see [Sections](#).

Related Information

[Add Annotation Terms, Items, and Records to the Local Annotation File \[page 381\]](#)

[Change the Order of Annotations and Collection Items \[page 387\]](#)

[Create Internationalized Labels \[page 388\]](#)

[Concatenate Values \[page 389\]](#)

[Use Parameters in URLs \[page 389\]](#)

[Edit External Annotations \[page 390\]](#)

7.8.5.2.1 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 \[page 370\]](#), section *Scope*.

Procedure

1. Choose the *Select Targets* button.

The top-level target types are displayed in alphabetical order. By default, one entity type is 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 addition to the existing ones.

7.8.5.2.2 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 \[page 380\]](#).
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 \[page 382\]](#)

7.8.5.2.2.1 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*.

→ 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 \[page 378\]](#).

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 \[page 380\]](#).
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					
SEPMRA_C_PD_ProductType					
Local Annotations		Source: localAnnotations			+ + Delete
UI.Facets	Edit				+ + Delete
UI.CollectionFacet		ID: ProductInfo; Label: Product Information			+ + Delete
ID			String	ProductInfo	+ + Delete
Label			String (i18n)	Product Information	+ + Delete
*Facets					+ +
UI.ReferenceFacet		Label: Second Facet; Target: {@UI.Chart#...}			+ + Delete
UI.ReferenceFacet					+ + Delete
*Target			AnnotationPath	Navigation + Annotation +	+ +
External Annotations		Source: SEPMRA_PROD_MAN_ANNO_M...			
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*.

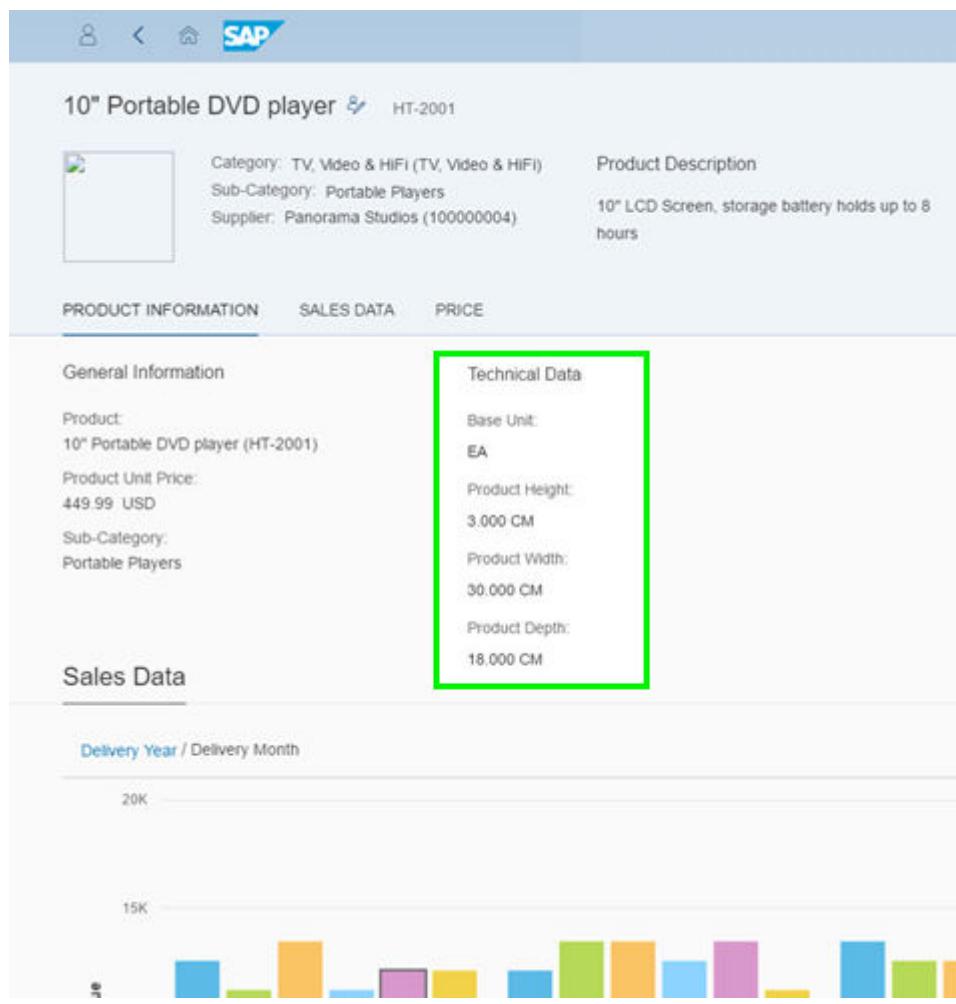
The term following @ defines the annotation type. The term following # defines the qualifier. This means that @*UI.FieldGroup#TechnicalData* references the *UI.FieldGroup* annotation with the qualifier *TechnicalData*.

i Note

In the backend of this sample project, a *UI.FieldGroup* that contains all necessary technical data already exists. To display the content on the UI of the application, we reference this *UI.FieldGroup* in our new *UI.ReferenceFacet*.

- 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, navigate to 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 shows a SAP Fiori application interface for a product. At the top, there's a header with a user icon, back arrow, home icon, and the SAP logo. Below the header, the product name "10" Portable DVD player" and code "HT-2001" are displayed. To the left is a thumbnail image. To the right, there are sections for "Category: TV, Video & HiFi (TV, Video & HiFi)", "Sub-Category: Portable Players", and "Supplier: Panorama Studios (100000004)". A "Product Description" section follows, stating "10" LCD Screen, storage battery holds up to 8 hours". Below this, there are three tabs: "PRODUCT INFORMATION", "SALES DATA", and "PRICE". The "PRODUCT INFORMATION" tab is selected. Under this tab, there are two sections: "General Information" and "Technical Data". The "General Information" section contains details like Product (10" Portable DVD player (HT-2001)), Product Unit Price (449.99 USD), and Sub-Category (Portable Players). The "Technical Data" section, which is highlighted with a green border, lists Base Unit (EA), Product Height (3.000 CM), Product Width (30.000 CM), and Product Depth (18.000 CM). Below the "PRODUCT INFORMATION" tab, there's a "Sales Data" section. Under "Sales Data", there's a chart titled "Delivery Year / Delivery Month" showing sales volume over time. The Y-axis has markers for 15K and 20K. The X-axis shows months from January to December. Sales volumes are represented by colored bars: January (blue), February (green), March (orange), April (yellow), May (purple), June (light blue), July (green), August (orange), September (purple), October (blue), November (green), and December (orange).

7.8.5.2.3 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 \[page 380\]](#).
2. In the search field, enter the text you want to search for, and press `[ENTER]`.
For example, enter **Technical Data**, and press `[ENTER]`.
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.

7.8.5.2.4 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.

7.8.5.2.5 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](#)

7.8.5.2.6 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](#)

7.8.5.2.7 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 `Enter`.
For each parameter, a new node is created.
3. For each parameter, in the *Value* column, enter the target of the URL parameter.

7.8.5.2.8 Edit External Annotations

Annotations defined in the back end 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 back end 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 \[page 374\]](#).

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 \[page 393\]](#).

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 \[page 380\]](#).
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 \[page 379\]](#).

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.

7.8.5.2.9 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 \[page 392\]](#)

[Check Error Descriptions in the Problems View \[page 392\]](#)

7.8.5.2.9.1 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 field is highlighted in red and error icons appear at the node.
- Errors and warnings are propagated to the parent elements in the tree structure. If there are both errors and warnings in the child nodes, the error icon is displayed on the parent node.

→ Tip

To obtain an overview of all errors and warnings in your local annotation file, open the *Problem* view in SAP Web IDE.

7.8.5.2.9.2 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 annotation modeler, open the `annotations.xml` file in annotation modeler.

i Note

- The *Problems* view displays issues specific to annotation modeler only if the `annotations.xml` file is opened in annotation modeler.
- The *Problems* view displays only errors and warnings specific to annotation modeler.

2. Open the *Problems* view. See [Using the Problems View \[page 323\]](#).

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.

7.8.6 Troubleshooting

Error Message	Issue Description and Solution
<p>OData metadata cannot be loaded from destination <destination name>. This error can appear for several reasons:</p> <ul style="list-style-type: none">• The login data is incorrect: provide the correct credentials.• The OData Service URI is missing or defined incorrectly. Please check the OData Service URI in the manifest.json file (app descriptor) of this project.• The connection to the backend is not available. Please check the destination in the neo-app.json file of the project and make sure that the backend system is available.	<p>The annotation file cannot be loaded because the OData metadata file cannot be loaded from the back end.</p>

i Note

Annotation modeler loads metadata based on the URI entry defined in the manifest.json file. This is usually a URI pointing to a back-end destination. The localUri setting is not used, which prevents inconsistency between back-end metadata and its local copy.

- Make sure that you have entered the correct credential for accessing the back-end system.
- Make sure that the back-end system is available. If the back-end system is not available, annotation modeler cannot load the OData metadata file.
- Check the relevant entry of the dataSources setting of type OData in the manifest.json file (app descriptor). Make sure that the OData Service is correct and that there is an entry in the neo-app.json file of the same project that represents at least a part of the string of the URI of your data source.
- Check the destination that is referred to by the neo-app.json entry. The name of the destination must be the same as defined in SAP Cloud Platform Cockpit. Also make sure that the WebIDEUsage property contains either `odata_abap` or `odata_xs`, but not both. You can also use destinations of WebIDEUsage `odata_gen`. If your destination addresses a dedicated service with its full URL, you must specify the additional WebIDEAdditionalData property with the `full_url` value.

For more information, see [Prerequisites for Working with Annotation Modeler \[page 374\]](#) and [Connect to ABAP Systems \[page 158\]](#).

Error Message	Issue Description and Solution
<p>The file <file name> cannot be edited, because it is not registered in the manifest.json file (app descriptor).</p>	<ul style="list-style-type: none"> The local annotation file cannot be opened. The annotation file cannot be loaded because it is not registered in the app descriptor (<code>manifest.json</code> 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 <code>manifest.json</code> file. See Modifying the Application Descriptor Configuration [page 205] 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 390]. 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 377].
<p>The annotation xml file cannot be loaded because it contains xml format errors. Open the Problems view to view and fix the errors.</p>	<p>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.</p> <p>For more information, see Warnings and Errors in Annotation Modeler [page 392].</p>
<p>Invalid file: DataServices definition is missing.</p>	<p>Check that the annotation xml file includes an / <code>edmx : DataServices / Schema</code> node. You can create a valid file by creating an annotation file. For more information, see Create New Annotation Files [page 377].</p>
<p>There is no OData data source defined in the manifest.json. Please define an OData data source.</p>	<p>Check the relevant entry of the <code>dataSources</code> setting of type OData in the <code>manifest.json</code> file (app descriptor).</p>

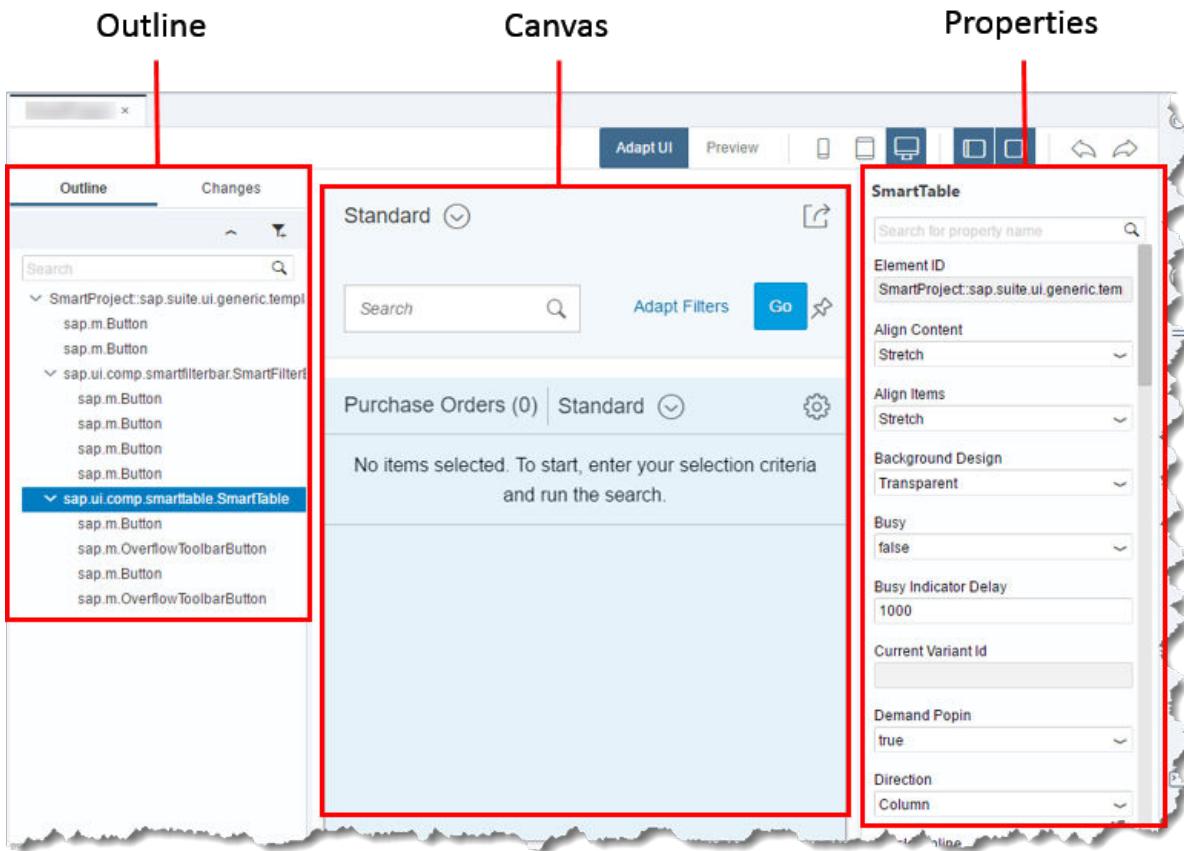
Error Message	Issue Description and Solution
<p>OData metadata cannot be loaded from destination <destination name>. The destination is not correctly maintained at the SAP Cloud Platform.</p>	<p>Check the destination that is referred to by the neo-app.json entry. The name of the destination must be the same as defined in SAP Cloud Platform cockpit. Also make sure that the WebIDEUsage property contains either <code>odata_abap</code> or <code>odata_xs</code>, but not both.</p> <p>You can also use destinations of WebIDEUsage <code>odata_gen</code>. If your destination addresses a dedicated service with its full URL, you must specify the additional WebIDEAdditionalData property with the <code>full_url</code> value.</p>

7.9 UI Adaptation Editor

UI Adaptation allows developers to adapt the user interface of their applications.

The UI Adaptation editor provides an intuitive user interface to make changes to SAP Fiori element applications. For example, you can add, remove, or move fields and group. You can also view the properties of the controls in the application and modify the ones that support modification.

It is composed of an Outline pane, the application runtime (canvas), and a Properties pane.



The buttons on the UI adaptation toolbar allow you to:

- Navigate in the application using the Preview mode.
- Modify the application using the UI adaptation mode. When in this mode, if you click a UI element in the previewed application, the element is selected and highlighted in the Outline pane and its properties are displayed and vice-versa. You can deselect it by clicking it again.

i Note

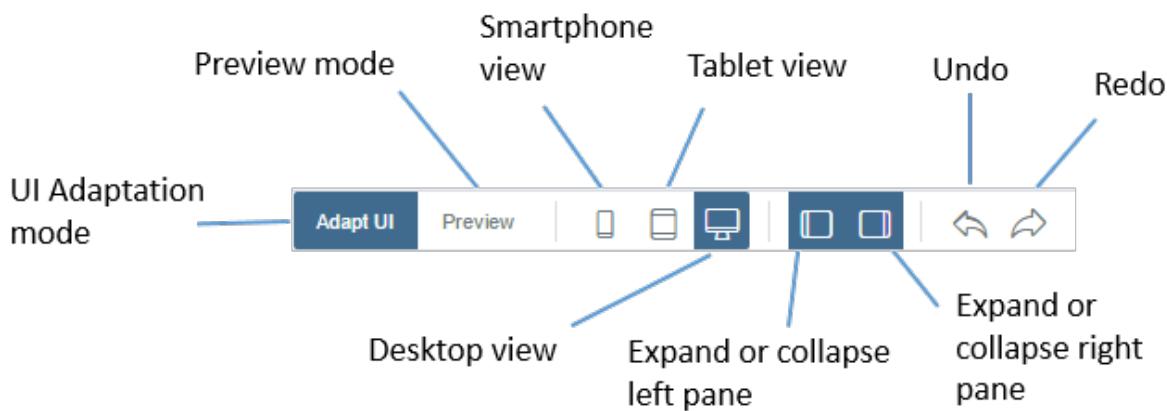
If you switch between modes, your changes are saved and copied to the workspace.

- 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 and copied to the workspace.

- Expand and collapse the panes to the right and left of the canvas.
 - The pane on the left side contains the *Outline* tab.
 - The pane on the right side contains the *Properties* pane.



Related Information

[Change Applications with the UI Adaptation Editor \[page 397\]](#)

[UI Adaptation Options \[page 399\]](#)

[Developing Apps with SAP Fiori Elements](#)

7.9.1 Change Applications with the UI Adaptation Editor

At runtime, you can make changes to 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 you have an annotation file. For more information see [Create Projects from Templates \[page 181\]](#).
- From the project's context menu select [Project Settings](#) [Project Types](#) and make sure the [UI Adaptation](#) project type is selected.

Procedure

1. From the generated project's context menu, select [UI Adaptation Editor](#). The application opens 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 [Adapt UI](#) to make changes.

- From the menu bar, select *Preview* to navigate to the page containing the UI element you want to change.
- 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

i Note

UI Adaptation can only be used for smart controls with stable IDs and for making property changes to non-smart controls.

The *Properties* pane displays the properties of the selected control.

i Note

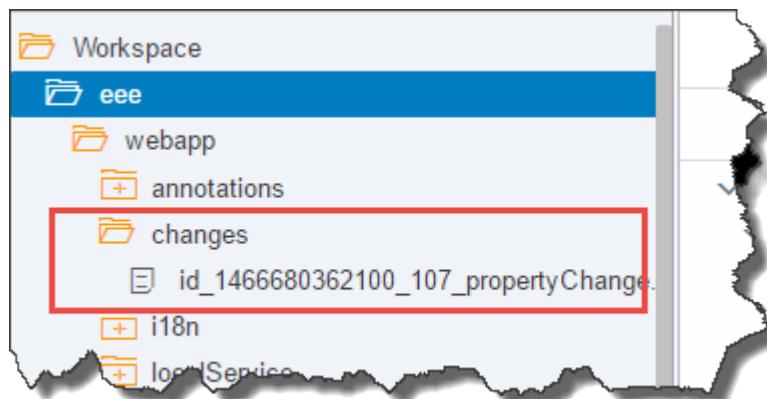
UI Adaptation can only be used for smart controls with stable IDs and for making property changes to non-smart controls.

- 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 are subject to change. 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 the device.

- Open the *Changes* tab or open your project in the workspace 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.



- Run the application to preview the changes.
- To incorporate these changes in the application residing in your SAPUI5 ABAP repository, deploy the application as described in [Deploy Applications to the SAPUI5 ABAP Repository \[page 447\]](#).

Related Information

[Configure Run Configurations for the UI Adaptation Editor \[page 202\]](#)

7.9.2 UI Adaptation Options

List of possible changes you can make to your application at runtime using the UI adaptation mode.

Change properties

1. Select the UI element you want to change.
2. Change the element's properties as needed.

i Note

Not all properties are subject to change.

Add new fields

1. Hover over or select a group or a field and choose *Add Field* from the context menu (right-click).
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 new groups

1. Hover over or select a group or select the form it's contained in and choose *Add Group* from the context menu (right-click).
Its default title is *New Group*, but you can rename it to whatever you want.
2. To apply your adaptations, press `[ENTER]` or `[ESC]`.

Add sections to an object page

1. Hover over or select a section and choose *Add Section* from the context menu (right-click).

i Note

If all available sections are placed 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 (right-click).
 2. Rename the field label or group title.
 3. To apply your adaptations, press **[ENTER]**; to quit, press **[ESC]**.
-

Move fields, groups, and object page sections

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.
If you drop the field, group, or section outside a dashed box, it's moved to this space and you exit the dragging mode.

Cut and paste fields and groups

1. Hover over or select a field or group and choose *Cut* from the context menu (right-click). 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.
2. To paste a cut field, hover over or select a highlighted group or a field in a highlighted group and choose *Paste* from the context menu. 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 get rid of 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]** 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 ID:
HT-1011

Product Name:
Notebook Professional 17

Time Stamp:
Jun 26, 2016, 12:31:40 AM

Rename
Add Field
Remove
Cut
Paste
Combine

Product Info

Product ID/Product Name:
HT-1011 Notebook Professional 17

Split combined fields

1. Hover over or select the combined fields.
2. Choose *Split* from the context menu (right-click).

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. Either choose *Remove Field*, *Remove Group*, or *Remove Section* from the context menu (right-click) 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.

7.10 Using Source Control (Git)

The SAP Web IDE includes the 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  *File*  *Git*. 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.

The screenshot shows the SAP Web IDE's Git pane. At the top, it displays the repository name "myTest" and the current branch "testBranch". Below the branch selection are several circular icons representing Git operations: Pull, Fetch, Rebase, Merge, Show Stash, Reset, and Fetch from Gerrit. A "Commit" section follows, containing a "Status table" which lists two files: "webapp/Component.js" and "webapp/index.html", both marked as modified (M). To the right of the status table are "Stage" and "Discard" checkboxes. Below the status table is a "Commit Description" input field with placeholder text "Insert commit description". At the bottom, there is a note about Change ID, a checkbox for "Amend Changes", and three buttons: "Commit and Push", "Commit", "Push", and "Stash".

Status	Name	Stage	Discard
M	webapp/Component.js	<input type="checkbox"/>	<input checked="" type="checkbox"/>
M	webapp/index.html	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Status table

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:

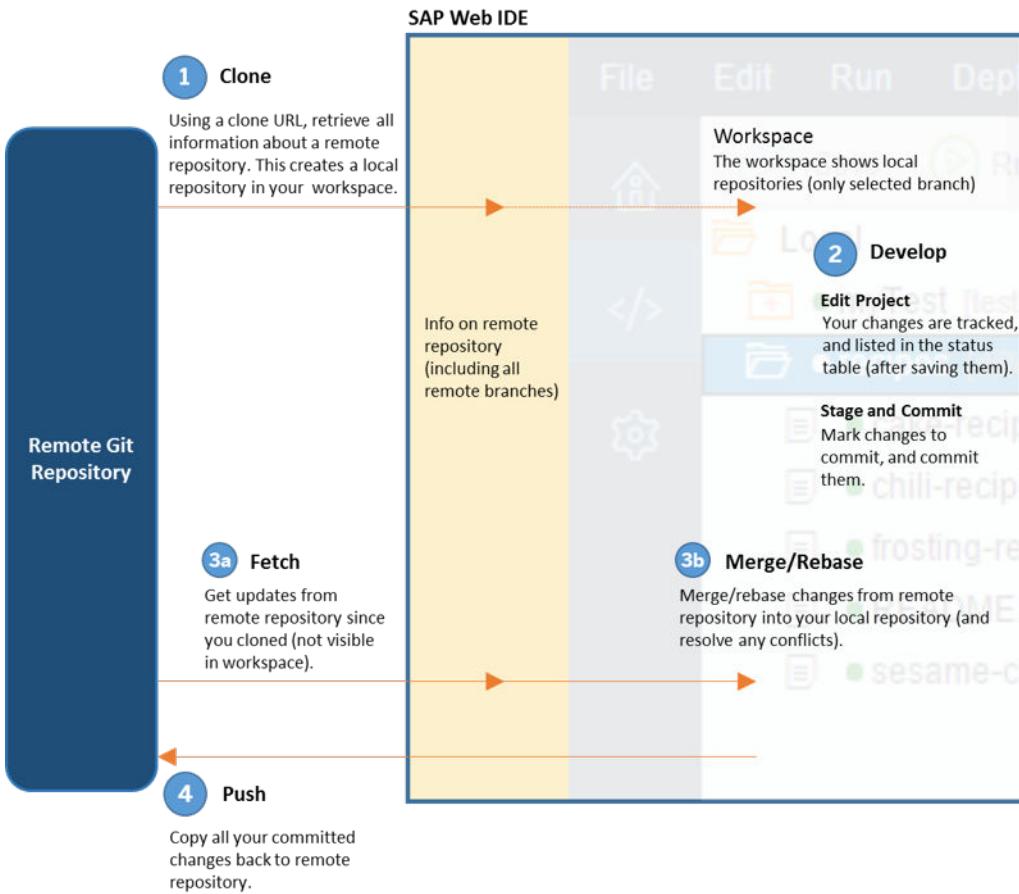
1. From the workspace, select a Git repository

2. From the right sidebar, choose  (Git History pane).
- **Git Annotate:** When working in a code editor, you can select *Show Git Annotations* 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. This is similar to the `git blame` command.
For more information, see [Working in the Code Editor \[page 214\]](#).

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* . You can then connect the local repository to a remote repository by choosing  *Git* 
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.



[Set Up Git \[page 406\]](#)

To use source control in your SAP Web IDE project, your user name and email address must be set for your Git account.

[Connect to your Corporate Git System \[page 408\]](#)

You can subscribe to the corporate Git connectivity service and manage the connectivity to your on-premise Git repository.

[Clone Repositories \[page 410\]](#)

You can clone an existing Git repository into your workspace.

[Initialize a Local Git Repository \[page 411\]](#)

You can initialize a local repository for any project that is not already connected to a Git repository.

[Set a Remote Repository \[page 412\]](#)

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.

[Fetch Changes \[page 413\]](#)

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.

[Rebase Changes \[page 413\]](#)

Rebasing enables you to take all the committed changes from one branch and incorporate them into a different branch.

[Merge Changes \[page 415\]](#)

You can incorporate all the changes from one branch into another in a single commit.

[Pull Changes \[page 415\]](#)

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.

[Stage Files \[page 416\]](#)

The status table shows changed files, and lets you select files to stage.

[Commit Changes \[page 421\]](#)

You can commit changes to the repository locally.

[Push Changes \[page 422\]](#)

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.

[Use Multiple Local Branches \[page 423\]](#)

From the Git pane, you can check out a local branch, add a new local branch, and remove a local branch.

[Reset a Local Branch \[page 425\]](#)

You can delete all new objects and references that were added to an existing local branch to make it identical to its remote branch.

[Create a Remote Branch \[page 425\]](#)

You can create a new branch in the remote repository.

[Git History \[page 426\]](#)

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.

[Set Up Git to Work with Gerrit \[page 429\]](#)

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.

7.10.1 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.

 **Note**

The email address field is case-sensitive.

 **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*.

Related Information

[Git Client Plugin \[page 154\]](#)

7.10.1.1 Git Decorations

Any change in a file's status is reflected in the workspace by decorations.

The table below shows the meaning of these decorations:

Decoration	Meaning
	Committed file
	Modified file that has not been staged
	Modified file that has been staged
	New file
	Folder containing deleted files
	File with merge conflicts

7.10.2 Connect to your Corporate Git System

You can subscribe to the corporate Git connectivity service and 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 an SAP Cloud Platform connector. For more information, see [SAP Cloud Platform 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>Virtual Host / Port</i>	Enter a virtual host and port for your Git system. You can use the same host and port as for the internal 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>

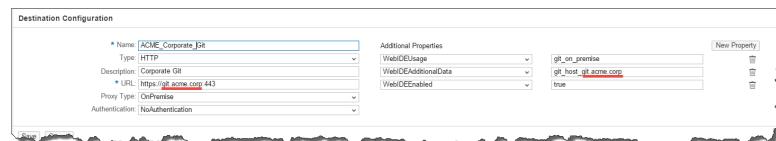
3. Upload your organization's Git server certificate to the cloud connector (if your Git server is using certificate-based authentication).

- Define your corporate Git destination. For more information, see [Connect to ABAP Systems \[page 158\]](#).
 - In the SAP Cloud Platform cockpit, select [Connectivity](#) [Destinations](#).
 - Select [New Destination](#).
 - In the *Destination Configuration* section, set the *Proxy Type* to *OnPremise*.
 - In the *Additional Properties* section, configure the following:

Property	Value
WebIDEEnabled	true
WebIDEUsage	git_on_premise
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`.

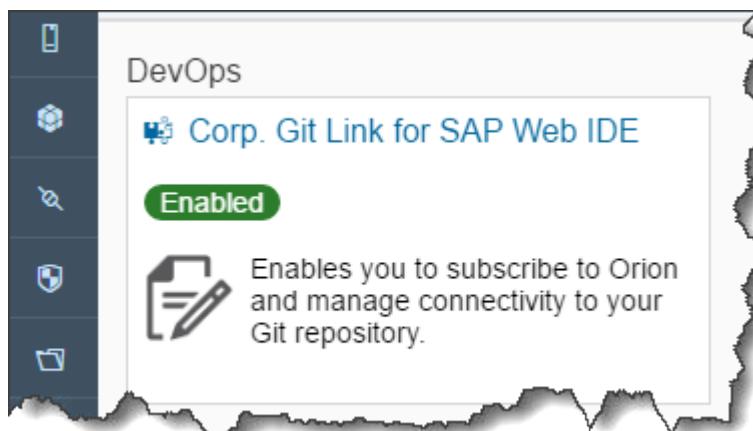
So you would enter in the field: `git_host_git.acme.corp`



Note

When you define the cloud connector, there are two types of hosts: Internal and Virtual. Make sure to use the internal host.

- Subscribe to the SAP Web IDE backend system.
 - In the SAP Cloud Platform cockpit, select [Services](#).
 - Select the [Corp. Git Link for SAP Web IDE](#) tile.



- Click *Enable*.

Next Steps

Test your new corporate Git system by cloning a repository from your corporate Git system (see [Clone Repositories \[page 410\]](#)) or by initializing a local repository, setting the remote repository, fetching and pushing to the remote repository (see [Initialize a Local Git Repository \[page 411\]](#)).

7.10.3 Clone Repositories

You can clone an existing Git repository into your workspace.

Procedure

1. From the *File* menu, choose *Git* .

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.

7.10.3.1 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 .
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).

7.10.4 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 \[page 412\]](#)

7.10.5 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 .
2. Right-click and select .
3. Enter the remote repository URL, and a name for the remote repository.

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>

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 \[page 411\]](#)

7.10.6 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. 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 all the available branches. The *Changes Fetched* table shows the fetched changes for each branch.

5. Choose *OK*.

7.10.7 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 \[page 414\]](#)

7.10.7.1 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. SAP Web 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.
For more information, see [Compare Code Versions \[page 417\]](#).

7.10.8 Merge Changes

You can incorporate all the changes from one branch into another in a single commit.

Procedure

1. In the *Git Pane*, choose *Merge*.
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 SAP Web IDE 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.

7.10.9 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.

7.10.10 Stage Files

The status table shows changed files, and lets you select files to stage.

Context

Whenever a file is updated, added, or deleted, it appears in the status table but is not staged. After staging, you can commit the staged files.

Procedure

In the status table of the *Git Pane*, choose the *Stage* checkbox in the row that contains the change that you want to stage.

i Note

To stage all files in the status table, select the *Stage All* checkbox at the top of the table.

You can edit a file listed in the status table by right-clicking the file name and choosing *Edit*. This option is enabled for all file statuses, except for deleted files (status D).

You can delete a file listed in the status table by right-clicking the file name and choosing *Delete*. This removes the file from the status table and from the workspace. This option is only enabled for new files (status N) that have not been staged.

Related Information

[Compare Code Versions \[page 417\]](#)

[Discard Changes \[page 418\]](#)

[Ignore Files \[page 418\]](#)

[Stash Changes \[page 419\]](#)

7.10.10.1 Compare Code Versions

You can compare different versions of your code.

Context

Use the SAP Web IDE compare editor to compare a modified version of your code with a version from the staging tables in the *Git* pane.

You can also compare two committed versions from the *Git History* pane.

Procedure

1. In the status table in the *Git Pane*, double-click the file in the staging table row or right-click it and choose *Compare* from the context menu.

The title of each pane indicates whether it contains the latest editable version or the previous read-only version. The read-only file is indicated by background shading.

i Note

You can only compare files that are modified (status M) or that have conflicts (status C).

2. Compare your modified code with the original code as follows:
 - Choose *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.
 - Choose *Copy from right to left* to move selected code from the original version to the modified version. The highlighted lines on the right side will replace the highlighted lines on the left side.
3. Edit your code and resolve any conflicts that may have occurred, especially after a rebase or merge operation.

i Note

If there is a conflict, decide which version of code to continue with. You may want to use a combination of both versions. The modified version is editable and you will eventually push this version.

7.10.10.2 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 status table of the *Git Pane*, choose  in the row that contains the change that you want to discard.

All changes that you made to the file are removed.

i Note

To discard all files, choose *Discard All* at the top of the table. All unstaged files in the table in the *Commit* section are discarded.

7.10.10.3 Ignore Files

You can mark specific files to not be tracked by Git.

Context

Ignored files will not be pushed to a remote repository. This option is enabled only for new files (status N) that have not been staged.

Procedure

In the status table of the *Git Pane*, right-click the file name and choose *Ignore*.

When ignoring a file, the path of the file is added to the `.gitignore` file (the `.gitignore` file is created if it does not exist). The `.gitignore` file is added to the status table, since it was changed.

Untracking Files

Untracking files lets you ignore files that have already been committed.

Procedure

1. In the workspace, select the file.
2. Choose [Untrack and Ignore](#).

The command adds the following files to the status table:

- `.gitignore` as modified, since your selected file will now be ignored.
- Your selected file as deleted, since you want to remove the tracked file from git.
- Your selected file as untracked, since you still want to have the file locally.

Here's an example of how the staging area looks after choosing [Untrack and Ignore](#) for the file `test.txt`.

Status	Name	Stage	Discard
M	.gitignore	<input type="checkbox"/>	
U	test.txt	<input type="checkbox"/>	
D	test.txt	<input checked="" type="checkbox"/>	

3. Commit the changes.

7.10.10.4 Stash Changes

You can stash changes that you've made to your local repository, cleaning your workspace but storing the changes so you can restore them and work on them later.

Context

i Note

This feature is not available in SAP Web IDE personal edition.

You may want to stash changes, for example, when you want to save an unfinished feature, but you are not ready to commit.

When you stash changes, all changes are assigned to a stash, and the stash is pushed to a stack. You can later push additional stashes to the stack. You only see the name of the latest stash when choosing [Show Stash](#).

Procedure

1. In the [Git Pane](#), choose [Stash](#).
2. Optional - Modify the description for the stash. By default, the description is a timestamp.
3. Choose [OK](#).

Results

The changes are stored in a new stash and the list of changed files is emptied.

i Note

You should avoid using stash in certain, very rare, workflows, as follows:

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.

Applying Stashes

Context

If you have stashed changes, the [Show Stash](#) button in the Git pane is enabled, and you can apply the stashed changes to the files in your workspace.

i Note

You may need to resolve conflicts between the version of a file in your stash and the version in your workspace.

- If the file in your workspace has been committed and has conflicts with the stashed changes, the conflicts are shown in the file in the workspace.

- If the file in your workspace has not been committed, you may not be able to apply the stash. Either commit or reset the changes to the file in your workspace, and then apply the stash again.

Procedure

1. Choose [Show Stash](#).

The description of the current stash on the stack is displayed.

2. Do one of the following:

- Choose [Apply](#). The stashed changes are applied to your workspace but the stash remains on the stack.
- Choose [Pop](#). The stashed changes are applied to your workspace and the stash is removed from the stack.
- Choose [Drop](#). The stashed changes are deleted and can no longer be applied to your workspace. The workspace is unaffected.

7.10.11 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. SAP Web 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.

7.10.12 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.

Note

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.

7.10.13 Use Multiple Local Branches

From the Git pane, you can check out a local branch, add a new local branch, and remove a local branch.

Check Out a Local Branch

Procedure

In the [Git Pane](#), select the desired local branch.

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.

Create 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*.

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.

Delete 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.

7.10.14 Reset a Local Branch

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.

7.10.15 Create a Remote Branch

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*  or click the plus sign in the *Git* pane.

Procedure

1. Select your project.
2. From the menu, select *Git* *Create Remote Branch*.

Related Information

[Use Multiple Local Branches \[page 423\]](#)

7.10.16 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

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

Status	Description
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 \[page 427\]](#)

From the Git *History* pane, you can execute a number of Git commands.

Related Information

[Git Commands from the Git History Pane \[page 427\]](#)

7.10.16.1 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.	<ol style="list-style-type: none"> Choose <i>Cherry-Pick</i>. Choose <i>OK</i> to confirm the commit. If there is a conflict, a warning is displayed. From the Git pane, resolve any conflicts as follows: <ol style="list-style-type: none"> Open the Git pane and find the conflicting files in the staging table. Edit the files to resolve the conflicts. Stage the edited files. Commit the changes.
Revert a commit	Allows you to undo all the changes that were incorporated in the selected commit.	<ol style="list-style-type: none"> Select the commit in the list of commits and choose <i>Revert</i>. 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.	<ol style="list-style-type: none"> Select the commit in the list of commits and choose <i>Check Out</i>. Enter a name for the new branch. Choose <i>OK</i>. The new branch appears as checked out in the Git pane and all the commit content is refreshed in the workspace.

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 Code Versions [page 417] . To compare two committed version: <ol style="list-style-type: none">1. Select two commits from the list.2. Select the common file to compare.3. Choose <i>Compare</i>.
-----------------	--	--

Task overview: [Git History \[page 426\]](#)

7.10.17 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**  menu, choose *Clone Repository* or *Set Remote*.

If you select *Set Remote*, your project must have been initialized as a local repository with  **File**  *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.

Note

Make sure to only select the checkbox if your Git uses Gerrit.

Related Information

[Clone Repositories \[page 410\]](#)

[Initialize a Local Git Repository \[page 411\]](#)
[Set a Remote Repository \[page 412\]](#)
[View Changes in the Gerrit Pane \[page 430\]](#)
[Fetch Changes from Gerrit \[page 431\]](#)

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](#).

7.10.17.1 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.

 Note

If the change is ready to be merged, you can click on *Submit*.

Related Information

[Set Up Git to Work with Gerrit \[page 429\]](#)

7.10.17.2 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*.

The changes are fetched into a new local branch, and the branch is checked out.

 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.

7.11 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 Projects \[page 435\]](#).

If your HTML5 application reuses applications that are contained in your workspace or SAP Cloud Platform, 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](#).

[Run Applications from the Workspace \[page 433\]](#)

Run and test your application using different run configurations.

[Create Run Projects \[page 435\]](#)

Create run configurations that define how your project or unit test is executed.

[Run Applications in a Frame \[page 440\]](#)

Check the localizations and functionality of your application by running it in a frame..

[Run Applications in the SAP Fiori Launchpad Environment \[page 441\]](#)

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 \[page 443\]](#)

You can run your application as a unit test to determine if it is working properly.

[Run Applications with Mock Data \[page 443\]](#)

Run a web application using a client mock server to test your application without connecting to the OData provider.

7.11.1 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 Projects \[page 435\]](#).

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 Projects [page 435].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 <code>OK</code> to create a new web application run configuration in the background. The file runs automatically.	<ul style="list-style-type: none">Open without a frameRun without mock dataRun with all existing URL parameters
SAP Fiori Launchpad Sandbox	<ul style="list-style-type: none">SAP Web IDE uses the selected component <code>.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 441].If there is no selected component <code>.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 frameRun without mock dataRun with all existing URL parameters

Run As	Default Behavior	Default Settings
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 Projects [page 435]. 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 <code>OK</code> 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](#).

7.11.2 Create Run Projects

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

- 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 \[page 200\]](#).

- 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 \[page 436\]](#)

Define general settings for the run configuration.

[URL Components Tab \[page 437\]](#)

Define navigation information within the application.

[Advanced Settings Tab \[page 438\]](#)

Define advanced settings for the run configurations.

7.11.2.1 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 Run Configurations from the context menu of a file that runs the application, this file name is automatically displayed.• If you chose Run Configurations 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 With Frame to open your application in a frame with configurable viewing options. For more information, see Run Applications in a Frame [page 440].</p>

i Note

This option is not available for unit tests.

Setting	Description
Mock Data	<p>Select Run with mock data 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 199].• The application that you want to run uses the SAPUI5 OData model with JSON format. <p>i Note</p> <p>This option is not available for unit tests.</p>

Parent topic: [Create Run Projects \[page 435\]](#)

Related Information

[URL Components Tab \[page 437\]](#)

[Advanced Settings Tab \[page 438\]](#)

7.11.2.2 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">1. To add a new row for an additional parameter, choose Add Parameter.2. In the new row, enter a parameter name and its value.

Setting	Description
URL Hash Fragment	To define a hash fragment (also known as a fragment identifier) for the URL, in the <i>URL Hash Fragment</i> field, enter the fragment identifier without the hash delimiter. In the full navigation URL, the fragment identifier is appended after the URL parameters, and is preceded by a hash (#) delimiter.

Parent topic: [Create Run Projects \[page 435\]](#)

Related Information

[General Tab \[page 436\]](#)

[Advanced Settings Tab \[page 438\]](#)

7.11.2.3 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> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"><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></div> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"><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></div>
Application Destination Resources	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 Cloud Platform account.
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 Cloud Platform is automatically used.</p>

Parent topic: [Create Run Projects \[page 435\]](#)

Related Information

[General Tab \[page 436\]](#)

[URL Components Tab \[page 437\]](#)

7.11.3 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

- Select a file in the project that you want to run.
- 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.

7.11.4 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 \[page 441\]](#)

Test your SAP Fiori application in a simplified SAP Fiori launchpad environment.

[Test Multiple SAP Fiori Applications \[page 442\]](#)

Test the interaction of multiple SAP Fiori applications in a simplified SAP Fiori launchpad environment.

7.11.4.1 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 \[page 200\]](#).

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 Projects \[page 435\]](#).

Task overview: [Run Applications in the SAP Fiori Launchpad Environment \[page 441\]](#)

Related Information

[Test Multiple SAP Fiori Applications \[page 442\]](#)

7.11.4.2 Test Multiple SAP Fiori Applications

Test the interaction of multiple SAP Fiori applications in a simplified SAP Fiori launchpad environment.

Prerequisites

- You have created several SAP Fiori applications in SAP Web IDE. These applications are located in the same project.
- You have selected SAP Fiori as a project type. For more information, see [Set Project Types \[page 200\]](#).

Context

To test multiple SAP Fiori applications, carry out the following configuration steps:

Procedure

1. Create a local JSON file in which to store the configuration data.

i Note

Use the following naming convention: *fioriSandboxConfig*.json.

2. Configure a set of SAP Fiori applications in the JSON file.

For more information, go to <http://help.sap.com/nw-uiaddon> and select *Creating a Local Configuration File*.

3. In SAP Web IDE, navigate to the *fioriSandboxConfig*.json file that you created in step 1 and select it.
4. 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 applications run 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 Projects \[page 435\]](#).

Task overview: [Run Applications in the SAP Fiori Launchpad Environment \[page 441\]](#)

Related Information

[Test a Single SAP Fiori Application \[page 441\]](#)

7.11.5 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 Projects \[page 435\]](#).
- Using a default run configuration.
For more information, see [Run Applications from the Workspace \[page 433\]](#).

7.11.6 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 \[page 199\]](#).

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 \[page 445\]](#)

[Create Run Projects \[page 435\]](#)

7.11.6.1 Edit Mock Data

You can model the service data that you want to use as mock data in your application.

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.
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.

7.11.6.2 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 \[page 199\]](#).

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.

7.12 Running Applications with Performance Measurements

View performance measurements as you run your application.

Context

You can run an application and view performance measurements for each user interaction in the application preview, such as buttons and links. Alerts are displayed for measurements that exceed specified thresholds for your application. You can configure alert thresholds for the project in Project Settings.

Procedure

1. In your project, select the `index.html` file that runs the application.
2. In the toolbar, choose the *Measure Performance* button.
3. When your application is up and running, activate user controls.

The performance measurements for each user interaction are displayed in a table below the application.

Field	Description
Status	Whether the action generated an alert.
Start Time	The start time of the interaction.
Initiator	The control that triggered the interaction; for example, the Search button.
Event	The event that triggered the interaction; for example, click.
E2E Duration [ms]	The duration of the interaction from start to end.
Client duration	The duration of client processing during the interaction.
Round trips	The number of round trips during the interaction.
Round Trips Duration	The duration from first request sent to the end of last received response

Field	Description
Backend Requests	The number of backend requests (OData calls)
Backend Duration [ms]	The total duration of backend requests (OData calls) including network duration

Measurements that exceed the configured warning and error alert values are indicated.

7.13 Deploying Applications

You can deploy new applications from SAP Web IDE to different servers.

[Deploy Applications to the SAPUI5 ABAP Repository \[page 447\]](#)

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 \[page 450\]](#)

You can update your applications that reside in the SAPUI5 ABAP repository.

[Deploy Applications to SAP Cloud Platform \[page 451\]](#)

You can deploy any project from SAP Web IDE to SAP Cloud Platform as a new application or as an update to a previously deployed application in any of your SAP Cloud Platform accounts.

[Register Applications to SAP Fiori Launchpad \[page 454\]](#)

You can register your SAP Cloud Platform 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 \[page 456\]](#)

You can check whether your application has been deployed to SAPUI5 ABAP Repository and/or SAP Cloud Platform.

[Application Build \[page 456\]](#)

Applications in production environments generally require a build to package resources in an optimal format, for example, to improve loading and delivery.

7.13.1 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 \[page 161\]](#).

Procedure

1. In your workspace, right-click the desired project.
2. Select **Deploy to SAPUI5 ABAP Repository**.
3. From the **System** dropdown list, select the desired system.

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 \[page 449\]](#).

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**.

Note

The **Browse** button is enabled only after providing a name for the application.

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.

Note

The request ID is automatically generated.

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.

If the application is configured to be built upon deployment (by setting the *SAPUI5 Client Build* project type in the Project Settings), only the content of the build target folder (called `dist` by default) is deployed.

If this configuration is not set, but the application contains a `webapp` folder, only the content within this folder is deployed.

For more information, see [Application Build \[page 456\]](#).

Related Information

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

7.13.1.1 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 \[page 438\]](#)).
- Always test your application on the server after deployment.

7.13.2 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*.

Note

If the project was already deployed, the fields will be automatically populated.

3. From the *System* dropdown list, select the desired system.

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.

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.

Note

The request ID is automatically generated.

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.

7.13.3 Deploy Applications to SAP Cloud Platform

You can deploy any project from SAP Web IDE to SAP Cloud Platform as a new application or as an update to a previously deployed application in any of your SAP Cloud Platform accounts.

Prerequisites

- You must be working in the cloud edition of SAP Web IDE.
- Make sure that you are a member of the account with at least a developer role.

Procedure

1. In the workspace, right-click the project and choose ► *Deploy* ► *Deploy to SAP Cloud Platform* ▾.
2. If you are deploying a new application, your default SAP Cloud Platform 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 HCP 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 Cloud Platform.

4. Choose *Deploy*.
The new application is deployed to SAP Cloud Platform. 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 Cloud Platform.

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 Cloud Platform.

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 Cloud Platform.

SAP Fiori applications can be run from SAP Fiori launchpad on SAP Cloud Platform. In this case, they don't have to include an executable HTML file. For more information, see [Register Applications to SAP Fiori Launchpad \[page 454\]](#).

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 [Updating Site Apps](#).

Results

When you deploy a project, an application build is performed in the background. The artifact deployed to HCP 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.

If the application is configured to be built upon deployment (by setting the *SAPUI5 Client Build* project type in the Project Settings), only the content of the build target folder (called `dist` by default) is deployed.

For more information, see [Application Build \[page 456\]](#).

Related Information

[Register Applications to SAP Fiori Launchpad \[page 454\]](#)

Connecting a Project to the SAP Cloud Platform Git Repository

Prerequisites

You must be an account administrator.

Context

The application source code should be managed in Git. When you deploy the application to HCP, the source code is not automatically pushed to the HCP Git. You can choose to connect your project to any Git repository (including the HCP Git) and push your changes there later.

Procedure

1. Go to Tools > SAP Cloud Platform Cockpit > Git Repositories .

The screenshot shows the SAP HANA Cloud Platform Cockpit interface. On the left, a sidebar menu lists various project categories: Java Applications, Database Systems, Databases & Schemas, HTML5 Applications, Virtual Machines (BETA), Destinations, Connectivity, Subscriptions, Services, Members, Trust, Authorizations, OAuth, Document Repositories, **Git Repositories** (which is selected and highlighted with a red border), Resource Consumption, Useful Links, and Legal Information. The main content area is titled "SAP HANA Cloud Platform Cockpit" and displays a table of Git repositories. The table has columns for Status (ACTIVE), Repository Name, Last Commit, Progress (1%), and a progress bar. The listed repositories include fioriapp, fioriapp076633, fioriappl, fioriappnewone, firebase, fioricarriew, fioridetail, fiorilib, fiorimaster, fiorimd, fiorimdapp, fiorimdapp12, fioritest, firolike, flights, flightss, fromworkspace, fs, full, fullexmounika, fullscreen, and gaia.

Status	Repository Name	Last Commit	Progress	
ACTIVE	fioriapp	fioriapp	1%	<div style="width: 10%;">1%</div>
ACTIVE	fioriapp076633	fioriapp076633	1%	<div style="width: 10%;">1%</div>
ACTIVE	fioriappl	fioriappl	1%	<div style="width: 10%;">1%</div>
ACTIVE	fioriappnewone	fioriappnewone	1%	<div style="width: 10%;">1%</div>
ACTIVE	firebase	firebase	1%	<div style="width: 10%;">1%</div>
ACTIVE	fioricarriew	fioricarriew	1%	<div style="width: 10%;">1%</div>
ACTIVE	fioridetail	fioridetail	1%	<div style="width: 10%;">1%</div>
ACTIVE	fiorilib	fiorilib	1%	<div style="width: 10%;">1%</div>
ACTIVE	fiorimaster	fiorimaster	1%	<div style="width: 10%;">1%</div>
ACTIVE	fiorimd	fiorimd	1%	<div style="width: 10%;">1%</div>
ACTIVE	fiorimdapp	fiorimdapp	1%	<div style="width: 10%;">1%</div>
ACTIVE	fiorimdapp12	fiorimdapp12	1%	<div style="width: 10%;">1%</div>
ACTIVE	fioritest	fioritest	1%	<div style="width: 10%;">1%</div>
ACTIVE	firolike	firolike	1%	<div style="width: 10%;">1%</div>
ACTIVE	flights	flights	1%	<div style="width: 10%;">1%</div>
ACTIVE	flightss	flightss	1%	<div style="width: 10%;">1%</div>
ACTIVE	fromworkspace	fromworkspace	1%	<div style="width: 10%;">1%</div>
ACTIVE	fs	fs	1%	<div style="width: 10%;">1%</div>
ACTIVE	full	full	1%	<div style="width: 10%;">1%</div>
ACTIVE	fullexmounika	fullexmounika	1%	<div style="width: 10%;">1%</div>
ACTIVE	fullscreen	fullscreen	1%	<div style="width: 10%;">1%</div>
ACTIVE	gaia	gaia	1%	<div style="width: 10%;">1%</div>

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 \[page 411\]](#).
4. Use the Git operations to fetch, commit, and push your changes. For more information, see [Using Source Control \(Git\) \[page 402\]](#).

7.13.4 Register Applications to SAP Fiori Launchpad

You can register your SAP Cloud Platform 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 Cloud Platform. For more information, see [Deploy Applications to SAP Cloud Platform \[page 451\]](#).
- You must be assigned the `TENANT_ADMIN` role in the SAP Cloud Platform cockpit. For more information, see [Accessing Services](#).
- You must have at least one site created in advance. For more information, see [About Sites](#).

i Note

You can access the SAP Cloud Platform cockpit from SAP Web IDE by selecting ► [Tools](#) ► [SAP Cloud Platform 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. Optionally, 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.

Related Information

[Deploy Applications to SAP Cloud Platform \[page 451\]](#)

7.13.5 Check the Application Status

You can check whether your application has been deployed to SAPUI5 ABAP Repository and/or SAP Cloud Platform.

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 Cloud Platform, its account, name, state, URL, link to the app page on SAP Cloud Platform, and versions are displayed.

i 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.

7.13.6 Application Build

Applications in production environments generally require a build to package resources in an optimal format, for example, to improve loading and delivery.

Any project with the project type `SAPUI5 Client Build` is automatically built when it is deployed. Building a project includes the following:

- A `resources.json` file listing all resources in the project is created to enable running SAPUI5 applications in mobile devices.
- In projects with a `Component.js` file, a minified JavaScript file (`Component-preload.js`) is created to improve loading performance.
- In projects with a `webapp` folder, a `dist` folder with the distributable application is created.

i Note

The `dist` folder is not created in projects that are created using the older (deprecated) templates: *Master-Detail*, *Master-Master-Detail*, and *Full Screen*.

- The `build` section of the `project.json` file is updated with the latest settings and the time of the most recent build.

Excluding Files from the Build

You can exclude specific files and folders from the build (and the `dist` folder) by changing the `build` settings in the `project.json` file. For example, the following excludes the `test` folder, and the `test1.html` and `test2.html` files:

↳ Source Code

```
"build": {  
    "excludedFolders": [ "/test" ],  
    "excludedFiles": [ "/webapp/test1.html", "/webapp/test2.html" ]  
}
```

Specifying the Source and Target Folders

For projects created from a template, the source and target folders are configured automatically. You can modify the source and target folders by setting the `targetFolder` and `sourceFolder` fields in the `build` settings in the `project.json` file.

↳ Source Code

```
"build": {  
    "targetFolder": "buildResults",  
    "sourceFolder": "sourceToBeBuild"  
}
```

Related Information

[Set Project Types \[page 200\]](#)

7.14 Extending SAPUI5 Applications

You can extend SAPUI5 applications residing remotely on the SAPUI5 ABAP repository or SAP Cloud Platform.

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 \[page 458\]](#)

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 Cloud Platform \[page 459\]](#)

You can extend an existing SAP Fiori application that resides on SAP Cloud Platform without importing it to SAP Web IDE.

[Create New Extensions \[page 460\]](#)

Extensions enable you to change the views or the logic of an extended project.

7.14.1 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 service in your back end. For more information, see [Requirements for Connecting to ABAP Systems \[page 161\]](#).

Context

You can also import an existing application to your workspace during the creation of the extension project.

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.
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 \[page 449\]](#).

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](#)

7.14.2 Extend Applications that Reside on SAP Cloud Platform

You can extend an existing SAP Fiori application that resides on SAP Cloud Platform 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 Cloud Platform*. The *Select Application from SAP Cloud Platform* dialog box is displayed.
3. Enter your SAP Cloud Platform 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.

7.14.3 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 \[page 461\]](#)

You can extend an existing SAP Fiori extension project from the extensibility pane, as well as by using the Extension wizard.

[Extend Controllers \[page 464\]](#)

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 \[page 465\]](#)

You can extend a view using an extension point.

[Hide Controls \[page 466\]](#)

You can hide a specific control in the original application.

[Edit Strings \[page 467\]](#)

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 \[page 468\]](#)

You can replace the extended application's OData service with a new OData service.

[Replace Views \[page 470\]](#)

You can replace a specific view in an original application with a new view.

7.14.3.1 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 and vice-versa: 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 Projects \[page 435\]](#).

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 \[page 460\]](#)

Related Information

[Extend Controllers \[page 464\]](#)
[Extend Views \[page 465\]](#)
[Hide Controls \[page 466\]](#)
[Edit Strings \[page 467\]](#)
[Replace OData Services \[page 468\]](#)
[Replace Views \[page 470\]](#)

7.14.3.2 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.

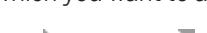
Task overview: [Create New Extensions \[page 460\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 461\]](#)
[Extend Views \[page 465\]](#)
[Hide Controls \[page 466\]](#)
[Edit Strings \[page 467\]](#)
[Replace OData Services \[page 468\]](#)
[Replace Views \[page 470\]](#)

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.

7.14.3.3 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.

i 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 \[page 460\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 461\]](#)

[Extend Controllers \[page 464\]](#)

[Hide Controls \[page 466\]](#)

[Edit Strings \[page 467\]](#)

[Replace OData Services \[page 468\]](#)

[Replace Views \[page 470\]](#)

7.14.3.4 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 \[page 460\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 461\]](#)

[Extend Controllers \[page 464\]](#)

[Extend Views \[page 465\]](#)

[Edit Strings \[page 467\]](#)

[Replace OData Services \[page 468\]](#)

[Replace Views \[page 470\]](#)

7.14.3.5 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* .
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 \[page 460\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 461\]](#)

[Extend Controllers \[page 464\]](#)

[Extend Views \[page 465\]](#)

[Hide Controls \[page 466\]](#)

[Replace OData Services \[page 468\]](#)

[Replace Views \[page 470\]](#)

7.14.3.6 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 \[page 432\]](#).

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 \[page 460\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 461\]](#)

[Extend Controllers \[page 464\]](#)

[Extend Views \[page 465\]](#)

[Hide Controls \[page 466\]](#)

[Edit Strings \[page 467\]](#)

[Replace Views \[page 470\]](#)

7.14.3.7 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 \[page 460\]](#)

Related Information

[Extend UI Elements Using the Extensibility Pane \[page 461\]](#)

[Extend Controllers \[page 464\]](#)

[Extend Views \[page 465\]](#)

[Hide Controls \[page 466\]](#)

[Edit Strings \[page 467\]](#)

[Replace OData Services \[page 468\]](#)

8 Plugins and Add-Ons

You can enable plugins in SAP Web IDE to extend the existing functionality.

i Note

This feature is not available in SAP Web IDE personal edition.

Plugins are the building blocks of SAP Web IDE. They are used to group functionality into small units.

Plugins can expose services to provide public APIs. Plugin 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 optional functionality provided by plugins can be enabled or disabled from the Preferences perspective. For more information, see [Enable Plugins \[page 474\]](#).

The list below shows a selection of our most used plugins.

SAP Web IDE Plugins and Add-Ons

Plugin	Description
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>
BUILD	<p>You can create a project in SAP Web IDE that will generate code from a BUILD prototype to build a real application.</p> <p>See Extend Prototypes with SAP Web IDE </p>
OData Model Editor	<p>You can develop your own OData model using the OData Model Editor in the SAP Web IDE framework.</p> <p>See Developing an OData Model.</p>
SAP Cloud Platform Portal service	<p>You can create a site template to be used by the SAP Cloud Platform Portal service administrator.</p> <p>See SAP Cloud Platform Portal service.</p>
Visualization Extension	<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>
SAP Event Management	<p>You can generate your own transactional SAP Fiori apps for SAP Event Management.</p> <p>See SAP Web IDE plugin for SAP Event Management .</p>
SAP Enterprise Portal	<p>You can deploy SAPUI5 applications developed in SAP Web IDE to SAP Enterprise Portal.</p> <p>See SAP Enterprise Portal Plugin for SAP Web IDE.</p>

Plugin	Description
IoT Application Enablement	<p>You can build new IoT-related applications and customize them by using predefined components and templates.</p> <p>i Note</p> <p>In order to use the IoT project templates, you must first subscribe to the IoT Application Enablement Services.</p>
	<p>See IoT Application Enablement Reuse Components and Templates.</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 Plugin for SAP Web IDE.</p>
Fact Sheet Editor	<p>You can create fact sheets from scratch and edit existing fact sheets using a drag-and-drop approach.</p> <p>i Note</p> <p>This plugin has been deprecated.</p>
	<p>See Fact Sheet Editor.</p>
SAP Global Track and Trace	<p>You can create, modify and delete metadata models for SAP Global Track and Trace, and save them either in a temporary workspace or in a GIT repository you define. After finishing the metadata definition, you can download the metadata model to a local environment and then upload it to the runtime repository. This is a microservice on the target runtime environment and provides data access APIs to other components or microservices.</p>
	<p>See SAP Global Track and Trace Plugin for SAP Web IDE.</p>
	<p>i Note</p> <p>If you want to use the same class for several plugins, you must:</p> <ol style="list-style-type: none"> 1. Define the desired class as an SAP Web IDE service for the first plugin. 2. Consume the class as a service in the other plugins.

8.1 Enable Plugins

You can use plugins in SAP Web IDE to create and develop applications.

Context

SAP Web IDE includes additional plugins that are not enabled by default. If you want to work on a project that was created using a plugin or create a new project with a plugin, the plugin must be enabled.

⚠ Caution

An SAP Web IDE feature extends the functionality of SAP Web IDE and provides new capabilities to your IDE. Such features and plugins have full privileges to access your browser, your computer, and any data stored in your SAP Web IDE workspace or on SAP Cloud Platform, including the ability to read and modify your private and organizational data.

Features and plugins that are not provided by SAP are under the responsibility of the feature author and may have different privacy policies, terms of use, or quality levels. You can enable features and plugins that are not provided by SAP and use them at your own risk. It is strongly recommended that you enable only features and extensions that you trust. At any time and without warning, SAP reserves the right to remove, disable, or uninstall features or plugins that are not provided by SAP from your environment.

To learn more about feature and plugin development, see the [SAP Web IDE SDK](#).

Procedure

1. To open the *Preferences* perspective, in the left sidebar, choose  (Preferences).
2. Choose *Plugins*.
3. Select the desired plugin repository from the dropdown list.
4. Select the checkboxes of the plugins that you want to enable.
5. Choose *Save*.
6. Refresh your browser.

Results

You can use the selected plugin to create and develop your project.

9 Extending SAP Web IDE

SAP Web IDE architecture allows developers to easily extend SAP Web IDE functionality by developing custom plugins and templates.

You can learn about plugin 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 Cloud Platform (CP) benefit from the full scope of SAP Web IDE and leverage SAP Cloud Platform capabilities with frequent updates of the cloud environment. In addition we offer a personal edition of SAP Web IDE.

SAP Cloud Platform Web IDE personal edition (SAP Web IDE personal edition) is intended as a complementary IDE, to be installed by a single developer on a personal workstation, for off-line 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 all customers who have a license to productively use SAP Web IDE on SAP Cloud Platform.

i Note

The personal edition is updated periodically and may not include features of SAP Web IDE on SAP Cloud Platform.

It includes the possibility to sync with Git on SAP Cloud Platform when needed or use a local Git repository.

When switching work modes from the cloud edition to the personal edition (and vice-versa), 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 Cloud Platform
 - Running applications on SAP Cloud Platform
 - Extending applications on SAP Cloud Platform
 - Importing applications from SAP Cloud Platform
 - Registering to SAP Fiori launchpad on SAP Cloud Platform
- Plugin development
- Template development
- Use of optional plugins

i Note

The SAP Fiori Overview Page plugin 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
- Gerrit pane

- 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 \[page 477\]](#)

[Start SAP Web IDE Personal Edition \[page 481\]](#)

[Upgrade SAP Web IDE Personal Edition \[page 483\]](#)

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 Cloud Platform 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 \[page 479\]](#).

5. Start SAP Web IDE personal edition. For more information, see [Start SAP Web IDE Personal Edition \[page 481\]](#).

[Post Installation Troubleshooting for Mac Sierra Users \[page 479\]](#)

Steps required to run SAP Web IDE personal edition on Mac Sierra systems.

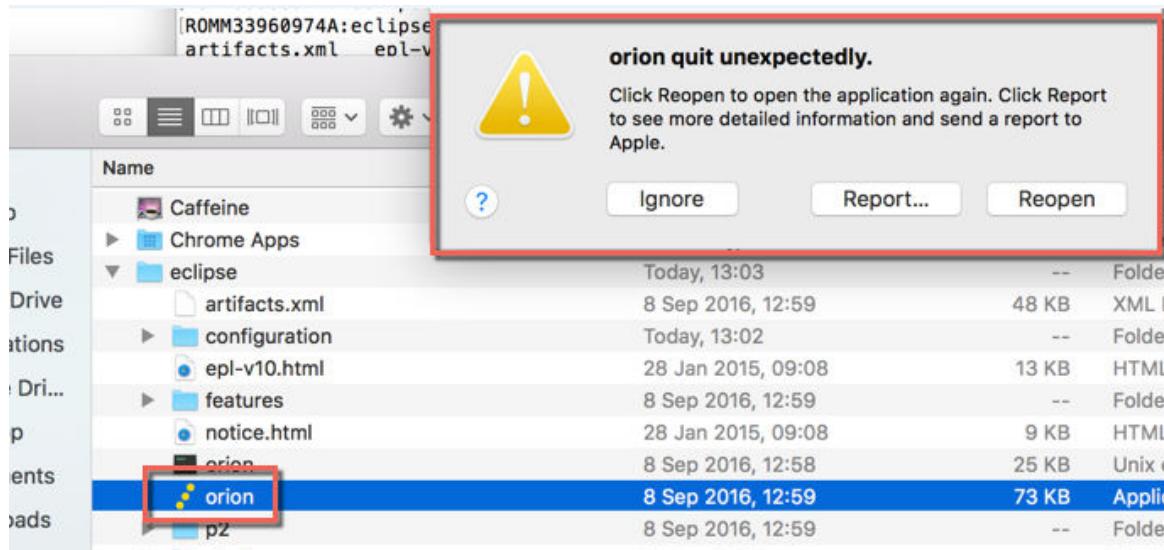
10.1.1.1 Post Installation Troubleshooting for Mac Sierra Users

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.

```

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.

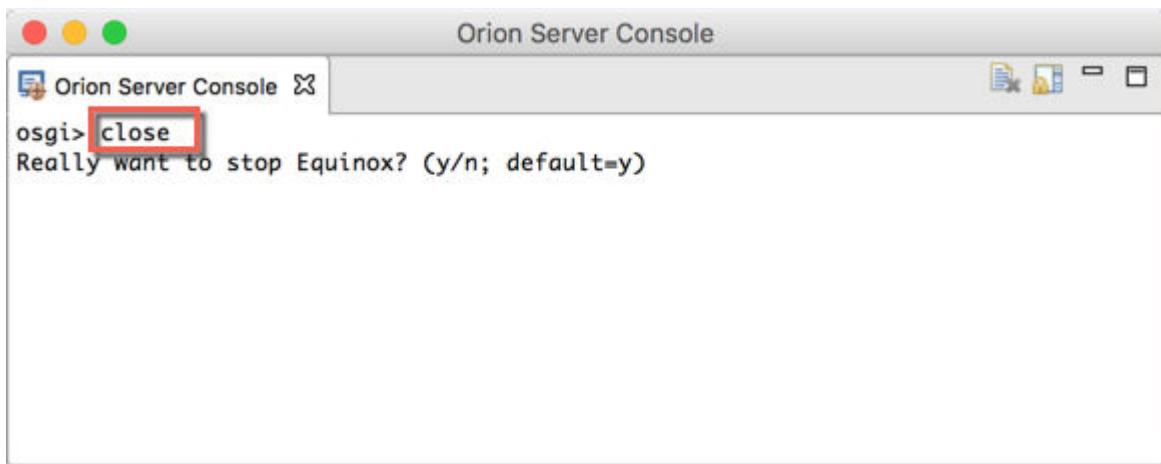
Sample Code

```

<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 \[page 477\]](#)

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 \[page 485\]](#).

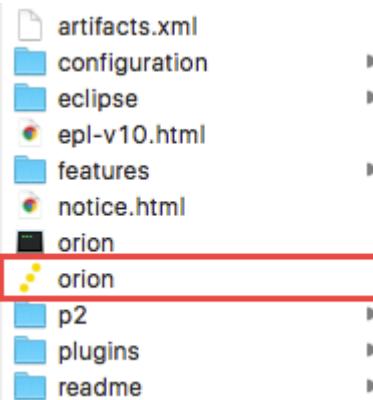
Microsoft Windows®

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 login 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 \[page 477\]](#).

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 \[page 479\]](#).

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 \[page 485\]](#)
- [Connect Remote Systems in SAP Web IDE Personal Edition \[page 486\]](#)
- [Import the Git Server Certificate into the JVM \[page 484\]](#)

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, by performing the steps.

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>`

i 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

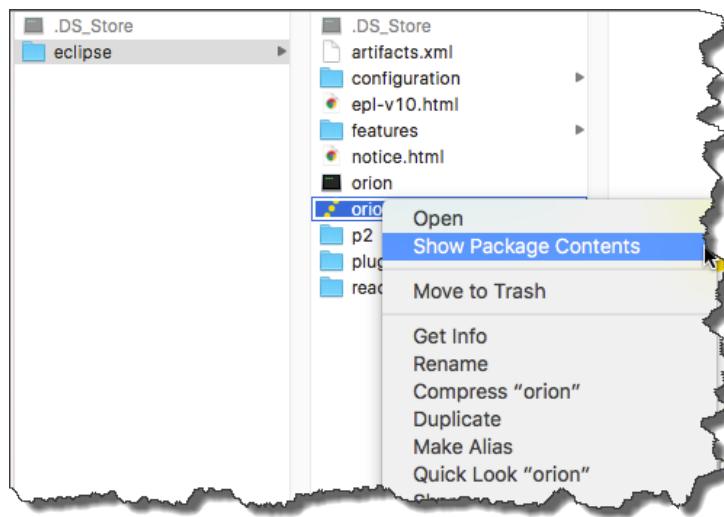
1. Open the `orion.ini` file:

Microsoft Windows®

Open the `orion.ini` file with a text editor.

Mac OS®

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 \[page 161\]](#).

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>
WebIDEEncoded=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"> ○ <code>odata_abap</code>: For the OData functionality of Gateway (corresponds to URL path <code>/sap/opu/odata</code>) ○ <code>odata_gen</code>: For generic OData functionality (service URL must be provided manually in the New Project wizard) ○ <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 the 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>odata_xs</code>: For SAP HANA XS OData services (corresponds to URL path <code>/sap/hba</code>)

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 \[page 485\]](#)

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 email 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 \[page 410\]](#)

10.6 Run Applications with Multiple SAPUI5 Versions

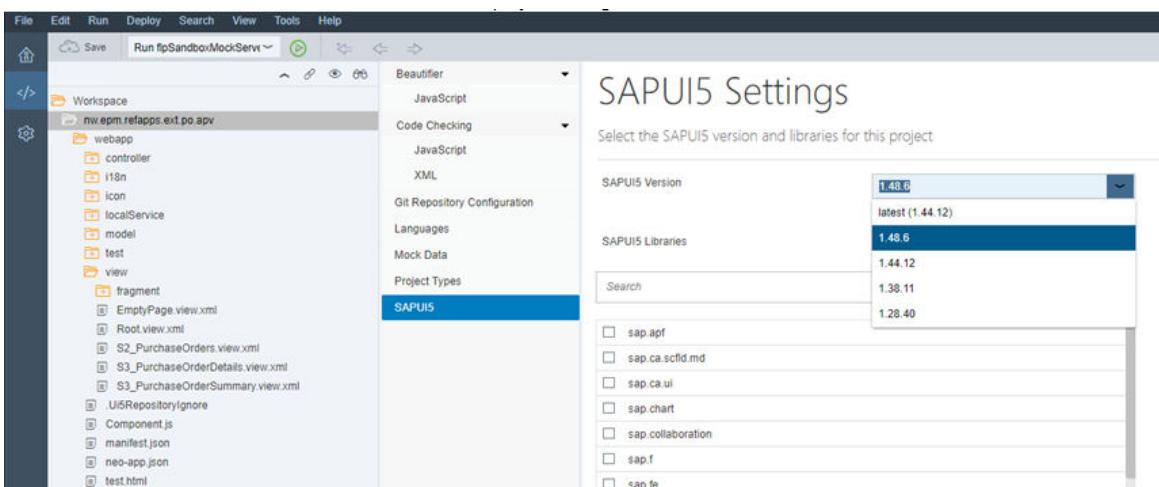
SAP Web IDE personal edition comes with three default SAPUI5 versions you can choose from. If you want to run your application using a different SAPUI5 version you must make the version available in SAP Web IDE personal edition.

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 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](#)
- [Destinations](#)
- [HTML5 Applications Development](#)

[Architectural Overview \[page 491\]](#)

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

[User Authentication and Authorization \[page 494\]](#)

User authentication and authorization processes in SAP Web IDE ensure that users can access only the resources to which they have the required permissions.

[Data Protection and Privacy \[page 499\]](#)

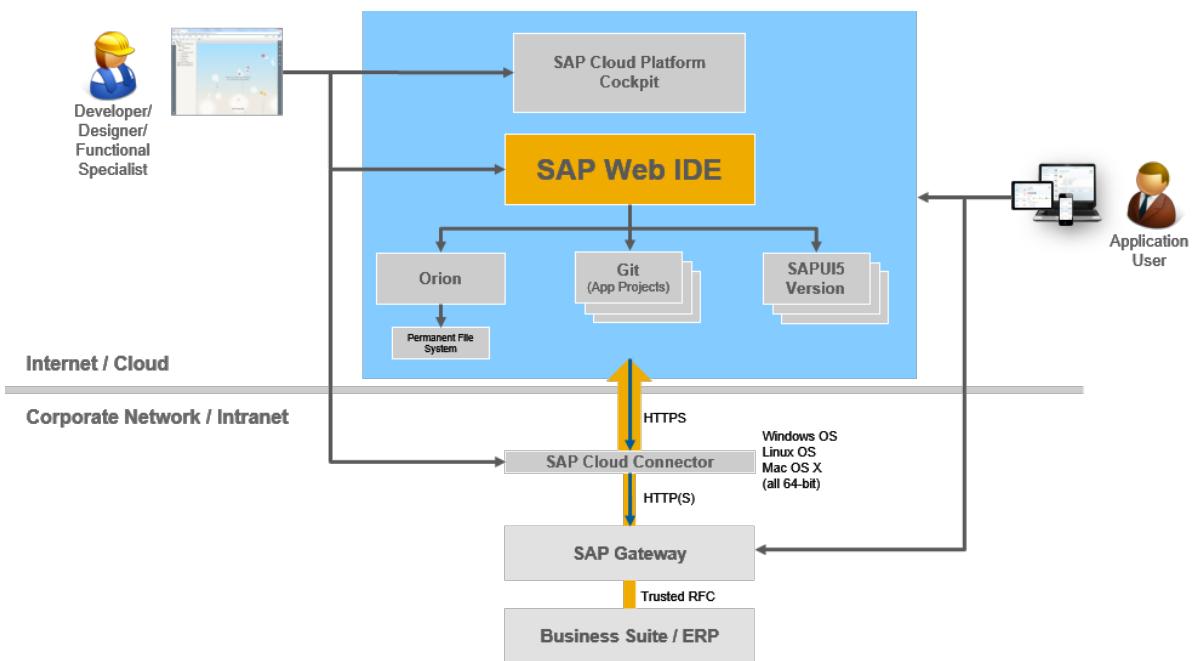
[Secure Programming Guide \[page 502\]](#)

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.

11.1 Architectural Overview

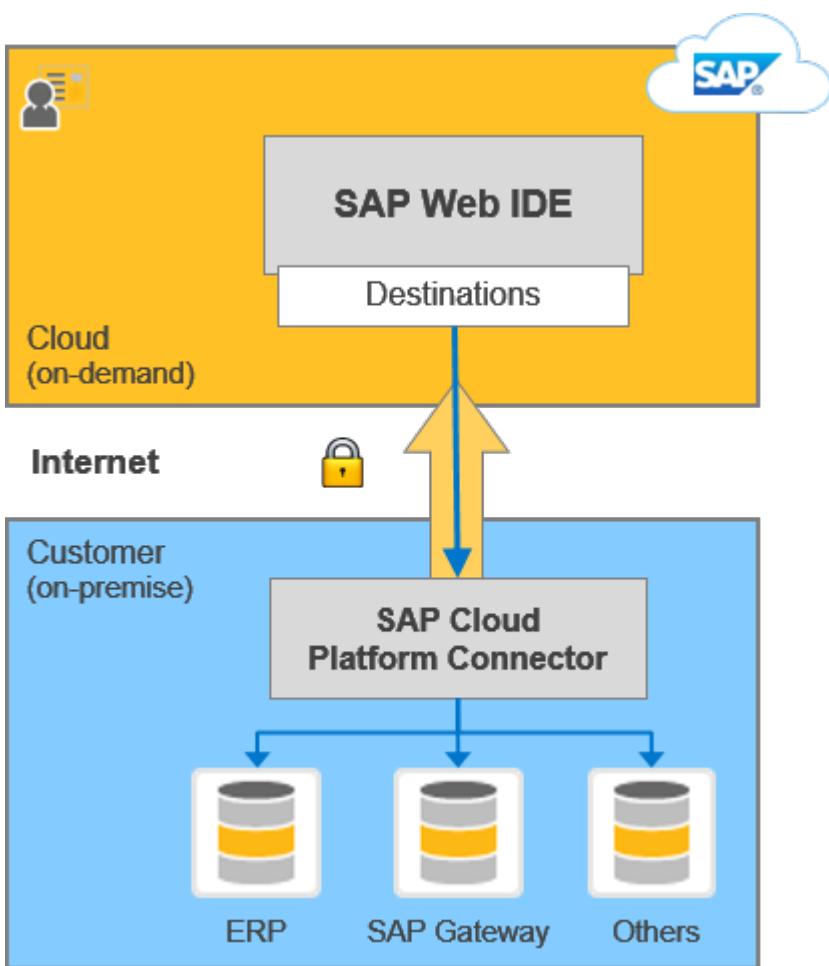
SAP Web IDE is a browser-based IDE consisting of integrated parts that interact with each other and with an SAP system.

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 image below illustrates the SAP Web IDE architecture. For more information about the dispatcher, see [HTML5: Development](#).



SAP Web IDE implements the following features of SAP Cloud Connectivity Service to provide a secure connection to on-premise systems:

- Establishes a secure SSL tunnel between SAP Cloud Platform 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.



Parent topic: Security [page 491]

Related Information

[User Authentication and Authorization \[page 494\]](#)

[Data Protection and Privacy \[page 499\]](#)

[Secure Programming Guide \[page 502\]](#)

[Connectivity Service](#)

[SAP Cloud Platform connector](#)

[HTML5: Development](#)

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.
- Orion: Used for file management. Orion leverages SSO as part of the user login to SAP Web IDE.
- Dispatcher (on SAP Cloud Platform): 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 Cloud Platform. However, to access the Git server on the cloud, you must provide your credentials (user and password) as configured in the SAP ID service.

Parent topic: [Security \[page 491\]](#)

Related Information

[Architectural Overview \[page 491\]](#)

[Data Protection and Privacy \[page 499\]](#)

[Secure Programming Guide \[page 502\]](#)

11.2.1 Authentication

Authentication in SAP Web IDE and user propagation 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 (SCI) 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

To push changes to the Git server, the SAP ID service requests the user's SCN credentials.

Related Information

[Identity and Access Management \(SAP Cloud Platform documentation\)](#)

[Principal Propagation \(SAP Cloud Platform documentation\)](#)

[Register, Create and Manage Your Profile \(SCN document\)](#)

[SAP Cloud Platform Connector \(SAP Cloud Platform documentation\)](#)

11.2.2 Assign Users Permission in SAP Web IDE

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 Cloud Platform 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 Cloud Platform cockpit.

For more information, see [Managing Roles](#).

Assigning Developer Permissions

1. In the SAP Cloud Platform 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 ([WebIDEPPermission](#)) 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 [WebIDEPPermission](#) row, in the [Assigned Role](#) column, select the role to which you want to assign the SAP Web IDE permission ([WebIDEPPermission](#)), 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 \[page 500\]](#) or [Delete User Data \[page 500\]](#).

You can assign the SAP Web IDE administrator role (SAP Web IDE Admin) as follows:

1. In the SAP Cloud Platform 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.

Related Information

[Managing Roles](#)

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.

11.2.4.1 Connection to External Systems

Access to SAP systems, Orion, and other external systems is maintained in the Dispatcher. Every system must have a defined destination.

By default, the SAP ID service is configured to connect your SAP system to SAP Web IDE, enabling an SAP Community Network (SCN) user account to call a configured destination in order to access the SAP system.

SAP Cloud Platform uses destinations as connection properties for accessing target systems. For more information on destinations, see [HTTP Destinations](#).

Destination Security Types

Type	Description	Comments
NoAuthentication	Any challenge request by the target system is returned to an end user and results in a credentials dialog box popup.	
PrincipalPropagation	End user identity is forwarded to the target system.	If the cloud connector is in use, it can use the identity and generate a new token.
BasicAuthentication	Credentials used for target system access are embedded into the destination configuration. This results in all requests using the same hard-coded credentials.	This option is not applicable for production environments. ⚠ Caution 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.

Related Information

[Managing Destinations](#)

11.3 Data Protection and Privacy

Overview

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 \[page 500\]](#) and [Delete User Data \[page 500\]](#).

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 Cloud Platform 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 need to provide their user name and email 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 email, 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.

Parent topic: [Security \[page 491\]](#)

Related Information

[Architectural Overview \[page 491\]](#)

[User Authentication and Authorization \[page 494\]](#)

[Secure Programming Guide \[page 502\]](#)

[Using Source Control \(Git\) \[page 402\]](#)

[Setting User Preferences \[page 163\]](#)

11.3.1 Export Workspace

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 Users Permission in SAP Web IDE \[page 495\]](#).
- 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 Cloud Platform 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.

11.3.2 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 SAP Web IDE Admin role. For more information, see [Assign Users Permission in SAP Web IDE \[page 495\]](#).
- 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 Cloud Platform, 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 Cloud Platform 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.

11.4 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](#).

Parent topic: [Security \[page 491\]](#)

Related Information

[Architectural Overview \[page 491\]](#)

[User Authentication and Authorization \[page 494\]](#)

Data Protection and Privacy [page 499]

12 Troubleshooting

Here are some common troubleshooting issues in SAP Web IDE.

Related Information

[Deployment to SAPUI5 ABAP Repository Troubleshooting \[page 504\]](#)

[Deployment to SAP Cloud Platform Troubleshooting \[page 506\]](#)

[Git Troubleshooting \[page 507\]](#)

12.1 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.

i Note

The technical messages are displayed in the language selected for your backend 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), so 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 the instructions under *Destination Properties* in [Accessing REST Services](#) and increase the timeout to the maximum of 300s.

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 use 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, but 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.

12.2 Deployment to SAP Cloud Platform Troubleshooting

Steps you can take if you have trouble deploying your application to SAP Cloud Platform.

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 Cloud Platform, 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 Cloud Platform).

Then, in SAP Cloud Platform, 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 Cloud Platform Neo 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 Cloud Platform account to the SAP Web IDE workspace via [File > Import > Application from SAP Cloud Platform](#).
 - 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.

12.3 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.

Invalid committer

The email listed in the Git repository is not the same as the email assigned to you in Gerrit.

1. Right-click on your project, select [Project Settings > Git Repository Configuration](#) and change the email 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 Changes](#) 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

When you set up your local repository (either by cloning, or by initializing your repository and then setting the remote repository), you did not indicate that you want to work with Gerrit, by selecting [Add configuration for Gerrit](#), when in fact your remote repository does work with Gerrit.

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. Checkout 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.

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. For more information on setting up a configuration for an internal Git system, see [Connect to your Corporate Git System \[page 408\]](#).
- **502 Bad Gateway:** You are using a Git system within your corporate network and the channel to your Git system that you opened in the cloud connector is either disconnected or contains the wrong URL to your Git system. For more information on creating a cloud connector channel to your internal Git system, see [.Connect to your Corporate Git System \[page 408\]](#).

Authentication not supported

This error occurs when using Git in SAP Cloud Platform and configuring a custom identity provider, and may be caused by not setting required properties for your custom identity provider. The needed properties must be set up in SAP Cloud Platform at [Security](#) [Trust](#) [Trusted Identity Provider](#) [Your identity provider](#) [Attributes](#) [Assertion-based Attributes](#) .

For more information, see [Initial Configuration](#) and [Configuring a CA Certificate for Principal Propagation](#) in the SAP Cloud Platform connector documentation.

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.

13 Known Issues

Locate the symptom and follow the recommended analysis and resolution steps for it.

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.

Cannot Clone the Corporate Git System

When trying to clone the corporate Git system, I get the following error message:



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.

Display of Buttons in the Frame of the Preview Window

Some of the buttons in the frame of the Preview window might not display when the width of the frame size is less than 1050 pixels. As the Preview frame size decreases, the buttons are hidden starting from the right side with the QR button.

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 in SAP Web IDE \[page 495\]](#).

Error Message When Logging On

You may receive an error message *Unhandled Error: Unexpected token* when logging onto 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 \[page 158\]](#)

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.

- **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 website to the list of trusted sites, enter the webide 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 and 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 one 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.

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.

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.

Error Deploying an App to the SAPUI5 ABAP Repository

- If there is a mismatch in the security protocol between the ABAP server and the destination defined in the cockpit, you might get the following message when you try to deploy an app to the SAPUI5 ABAP Repository:

HTTP Status 403 FORBIDDEN

Solution:

Change the destination definition so that it uses HTTPS or change the security configuration of the ABAP server so that it expects HTTP.

- If your destination configuration in the cockpit is set to `NoAuthentication`, and your ICF service on the ABAP server is defined as a public service (via an alias with fixed user credentials), you will get the following message:

HTTP Status 403 FORBIDDEN

Solution:

From your SAP Logon, open transaction `sicf` go to `sap > bc > adt` and select the *Logon Data tab*. If the *User* field is populated, clear it. When a user is defined, the credentials popup message does not appear and the call to get the csrf token fails.

- If you still receive the *HTTP Status 403 FORBIDDEN* message after trying the solutions above, implement SAP Note [1977537](#) which introduces a new concept for XHR (`XMLHttpRequest`) client authentication.

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.

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 Cloud Platform only as a platform and connect only to official SAP Cloud Platform APIs.
- Make no direct usage of third-party components within SAP Cloud Platform.

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.

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.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

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.