

SAP Cloud Platform Web IDE  
2018-03-01

## SAP Web IDE Full-Stack



# Content

|           |   |           |
|-----------|---|-----------|
| <b>1</b>  | <b>SAP Web IDE Full-Stack</b>                               | <b>5</b>  |
| <b>2</b>  | <b>What's New for SAP Web IDE</b>                           | <b>6</b>  |
| 2.1       | What's New in 2017  | 8         |
| 2.2       | Introducing SAP Web IDE Full-Stack                          | 16        |
| <b>3</b>  | <b>Overview</b>   | <b>17</b> |
| <b>4</b>  | <b>Features and Benefits</b>                                | <b>19</b> |
| <b>5</b>  | <b>User-Centric Customizability and Session Persistence</b> | <b>21</b> |
| <b>6</b>  | <b>Collaborative Development</b>                            | <b>22</b> |
| <b>7</b>  | <b>Assisted Development</b>                                 | <b>23</b> |
| <b>8</b>  | <b>Getting Started</b>                                      | <b>24</b> |
| 8.1       | Open SAP Web IDE  | 26        |
| 8.2       | Connect to Cloud Foundry Services                           | 27        |
|           | Troubleshooting   | 27        |
| 8.3       | Connect to ABAP Systems                                     | 28        |
|           | Requirements for Connecting to ABAP Systems                 | 31        |
| 8.4       | Neo and Cloud Foundry Regions                               | 32        |
| <b>9</b>  | <b>Setting User Preferences</b>                             | <b>34</b> |
| <b>10</b> | <b>SAP Web IDE Basics</b>                                   | <b>36</b> |
| 10.1      | Navigating SAP Web IDE                                      | 36        |
| 10.2      | Keyboard Shortcuts  | 38        |
|           | Customize Keyboard Shortcuts                                | 41        |
| 10.3      | Workspace Search Options                                    | 42        |
|           | Find and Replace in an Open File                            | 43        |
|           | Search for Files or Content in the Workspace                | 44        |
|           | Replace Strings Across Multiple Files                       | 45        |
|           | Find References   | 47        |
| 10.4      | Resizing Panes  | 48        |
| <b>11</b> | <b>Developing</b>   | <b>49</b> |
| 11.1      | Developing Web Applications                                 | 50        |
|           | Creating Projects   | 51        |
|           | Importing Projects  | 61        |

|   |            |
|---|------------|
| Customizing Your Project.....   | 66         |
| Modifying the Application Descriptor Configuration.....                     | 78         |
| Developing Applications.....  | 86         |
| Developing Application Tests.....   | 206        |
| Layout Editor.....  | 207        |
| Annotation Modeler.....   | 245        |
| Storyboard.....   | 271        |
| UI Adaptation Editor.....   | 276        |
| Using Source Control (Git).....   | 283        |
| Running Applications in Development Mode.....                               | 311        |
| Building Applications.....  | 325        |
| Deploying Applications.....   | 329        |
| Extending SAPUI5 Applications.....  | 338        |
| 11.2 Developing Multi-Target Applications.....                              | 352        |
| Inside an MTA Descriptor.....   | 355        |
| Setting Up Application Projects.....  | 359        |
| Developing SAP HANA Database (HDB) Modules.....                             | 362        |
| Developing Java Modules.....  | 366        |
| Developing HTML5 Modules.....   | 375        |
| Developing SAP S/4HANA Service Extensions.....                              | 380        |
| Packaging and Deploying Applications to Production Systems.....             | 385        |
| Developing Applications Using the SAP Cloud Platform Programming Model..... | 386        |
| <b>12 Additional Features.....</b>  | <b>390</b> |
| 12.1 Enable Additional Features.....  | 392        |
| <b>13 Extending SAP Web IDE.....</b>  | <b>393</b> |
| <b>14 SAP Web IDE Personal Edition.....</b>                                 | <b>394</b> |
| 14.1 Installation and Setup.....  | 395        |
| Install SAP Web IDE Personal Edition.....                                   | 395        |
| Start SAP Web IDE Personal Edition.....                                     | 399        |
| Upgrade SAP Web IDE Personal Edition.....                                   | 401        |
| Uninstall SAP Web IDE Personal Edition.....                                 | 402        |
| 14.2 Import the Git Server Certificate into the JVM.....                    | 402        |
| 14.3 Configure the Orion Application Server.....                            | 403        |
| 14.4 Connect Remote Systems in SAP Web IDE Personal Edition.....            | 405        |
| 14.5 Connect to an External Git Repository.....                             | 407        |
| 14.6 Run Applications with Multiple SAPUI5 Versions.....                    | 408        |
| <b>15 Security.....</b>   | <b>410</b> |
| 15.1 Architectural Overview.....  | 410        |
| 15.2 User Authentication and Authorization.....                             | 412        |

|   |            |
|---|------------|
| Authentication . . . . .  | 413        |
| Assign Users Permission for SAP Web IDE . . . . .                   | 413        |
| Browser Security . . . . .  | 415        |
| Transport Security . . . . .  | 416        |
| 15.3 Data Protection and Privacy . . . . .                          | 417        |
| Export Workspace . . . . .  | 418        |
| 15.4 Principal Propagation . . . . .                                | 419        |
| 15.5 Secure Programming Guide . . . . .                             | 420        |
| <b>16 Troubleshooting . . . . .</b>                                 | <b>421</b> |
| 16.1 Archive Import Troubleshooting . . . . .                       | 421        |
| 16.2 Deployment to SAPUI5 ABAP Repository Troubleshooting . . . . . | 422        |
| 16.3 Deployment to SAP Cloud Platform Troubleshooting . . . . .     | 424        |
| 16.4 Git Troubleshooting . . . . .                                  | 426        |
| <b>17 Known Issues . . . . .</b>                                    | <b>429</b> |
| <b>18 Third-Party Restrictions . . . . .</b>                        | <b>435</b> |

# 1 SAP Web IDE Full-Stack

You can download the English document in [PDF format](#).



## [What's New for SAP Web IDE \[page 6\]](#)

Learn about SAP Web IDE new features.



## [Overview \[page 17\]](#)

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



## [Getting Started \[page 24\]](#)

Learn how to set up and start SAP Web IDE.



## [Developing \[page 49\]](#)

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



## [Additional Features \[page 390\]](#)

Understand how to use additional SAP Web IDE features.



## [Extend SAP Web IDE \[page 393\]](#)

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



## [Security \[page 410\]](#)

Understand the security landscape.



## [SAP Web IDE Personal Edition \[page 394\]](#)

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

## 2 What's New for SAP Web IDE

### 1 March 2018 - SAP Web IDE

#### New

##### Static Code Checks in Build

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

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

#### Enhanced

##### Enhanced Integration with SAP Translation Hub

Destinations and trust management settings are now configured automatically.

For more information, see [Use SAP Translation Hub \[page 70\]](#).

### 15 February 2018 - SAP Web IDE

#### New

##### ESLint Plugin in SDK

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

For more information, see ► [Documentation](#) ► [Client-Side Plugin Programming Guidelines](#) ► [Static Code Checks](#) ▶ in the [SAP Web IDE SDK](#).

#### Enhanced

##### Stash Changes Operation in the Git Pane

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

For more information, see [Stash Changes \[page 298\]](#).

## 1 February 2018 - SAP Web IDE

### New

#### Support Assistant Setting

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

For more information, see [General Tab \[page 315\]](#) and [Support Assistant](#).

### New

#### Template

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

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

### New

#### HTML5 Modules for Multi-Target Projects

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

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

### i Note

These new modules work with SAPUI5 1.50 and later.

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

## 4 January 2018 - SAP Web IDE

### New

#### Debug Java Modules

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

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

For more information, see [Debug Java Modules \[page 372\]](#).

**New****Add or Delete View in Storyboard**

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

For more information, see [Add View \[page 275\]](#) or [Delete View \[page 276\]](#).

**New****Create a Navigation Between Views**

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

For more information, see [Create a Navigation Between Views \[page 215\]](#).

**New****Export User Workspace**

Administrators can now export the workspaces of a given user.

For more information, see [Export Workspace \[page 418\]](#).

## Archived Release Notes

- [What's New in 2017 \[page 8\]](#)

## 2.1 What's New in 2017

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

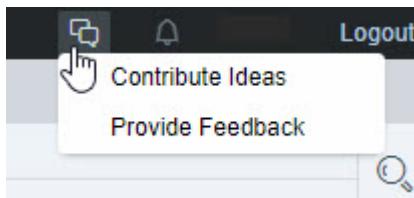
For more information about the latest release notes, see [What's New for SAP Web IDE \[page 6\]](#).

## 7 December 2017 - SAP Web IDE

### New

#### Contribute Ideas

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



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

### Enhanced

#### Provide Feedback

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

### Enhanced

#### Tagging Commits in the Git History pane

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

For more information, see [Git Commands from the Git History Pane \[page 306\]](#).

### Enhanced

#### ESLint v4.0.0 Support

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

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

## 23 November 2017 - SAP Web IDE

### New

#### Beautify Option in Code Editor Settings

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

For more information, see [Configure the Code Editor \[page 90\]](#).

## New

### New Functionality in Java Editor

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

- **Code Completion**

Code completion is now available in Java files.

- **Errors and Warnings in Problems View**

You can now see all error and warning messages in *Problems* view in Java files.

- **Beautify Code**

You can now beautify Java code from the context menu or the *Edit* menu.

 **Note**

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

## New

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

[Introduction and On-boarding](#) 

[Creating a Database Module](#) 

[Creating a Java Module](#) 

[Creating a UI Module](#) 

## New

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

See [Git Merge and Conflict Resolution](#) 

## Enhanced

### Annotation Modeler

You can now also use annotation modeler with mock data.

For more information, see [Configure Annotation Modeler to Use Mock Data \[page 66\]](#).

## Enhanced

### Cherry-Pick Operation in the Git History Pane

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

For more information, see [Git Commands from the Git History Pane \[page 306\]](#).

## 9 November 2017 - SAP Web IDE

### New

#### SAP Enterprise App Modeler

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

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

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

### New

#### Git History Pane

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

For more information, see [Git History \[page 305\]](#).

### Enhanced

#### Git Pane

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

For more information, see [Stage Files \[page 296\]](#).

## 26 October 2017 - SAP Web IDE

### Enhanced

#### Annotation Modeler

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

For more information, see [Architecture \[page 246\]](#).

## 14 September 2017 - SAP Web IDE

### New

#### Additional Features

- [SAP S/4HANA Service Extension Tools](#) enables you to develop OData V4 services that extend existing SAP S/4HANA services by exposing additional data from different sources. The tools include the dedicated project and Java module templates.
- For more information, see [Developing SAP S/4HANA Service Extensions \[page 380\]](#).

### Enhanced

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

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

For more information, see [Developing Java Modules \[page 366\]](#).

## 31 August 2017 - SAP Web IDE

### New

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

For more information, see [Use Eclipse to Develop Java Modules \[page 373\]](#).

### New

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

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

For more information, see [Develop Database Artifacts \[page 364\]](#).

### Enhanced

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

For more information, see [Edit Annotations \[page 255\]](#).

## 17 August 2017 - SAP Web IDE

### New

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

For more information, see [Packaging and Deploying Applications to Production Systems \[page 385\]](#).

## 3 August 2017 - SAP Web IDE

### New

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

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

### Enhanced

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

For more information, see [Search in Annotations \[page 262\]](#).

### Enhanced

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

### Enhanced

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

## 20 July 2017 - SAP Web IDE

### Enhanced

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

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

## 6 July 2017 - SAP Web IDE

### New

#### Compare in Git Status Table

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

For more information, see [Compare Code \[page 297\]](#).

### New

#### Workflow Editor Feature

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

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

## 22 June 2017 - SAP Web IDE

### New

#### SAP HANA Database Explorer

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

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

### Enhanced

#### Multitarget Application Development

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

For more information, see [Developing SAP HANA Database \(HDB\) Modules \[page 362\]](#) and [Developing Java Modules \[page 366\]](#).

## Enhanced

Using the Translation Hub

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

For more information, see [Use SAP Translation Hub \[page 70\]](#).

## 14 June 2017 - SAP Web IDE

### New

#### SAP Enterprise App Modeler Overview Feature

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

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

## 25 May 2017 - SAP Web IDE

### New

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

The screenshot shows the SAP Web IDE Data Connection wizard. The steps are: Template Selection, Basic Information, **Data Connection**, Template Customization, Confirmation. The Data Connection step is highlighted. Under Sources, the Service Catalog tab is selected. A red box highlights the text "If no destination exists for your Cloud Foundry service, configure a new destination".

For more information, see [Create Destinations for Cloud Foundry Services \[page 379\]](#)

## 17 May 2017 - SAP Web IDE

### Announcement

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

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

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

## 2.2 Introducing SAP Web IDE Full-Stack

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

To support SAP's full-stack development solution, this release includes changes to SAP Web IDE. In particular, SAP Web IDE now uses a powerful open-source server-side foundation called Eclipse Che, which will serve to introduce additional innovative development services as they become available. One new feature is already available in this release: the ability to create and run Grunt builds of your development project before final deployment.

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

You can continue to use the standard SAP Web IDE using the URL on the tile, or using an existing bookmark. There is no change, and no need to migrate your projects to the new release of SAP Web IDE. For more information, see [SAP Web IDE Developer Guide](#).

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 detailed onboarding instructions, see [Getting Started \[page 24\]](#).

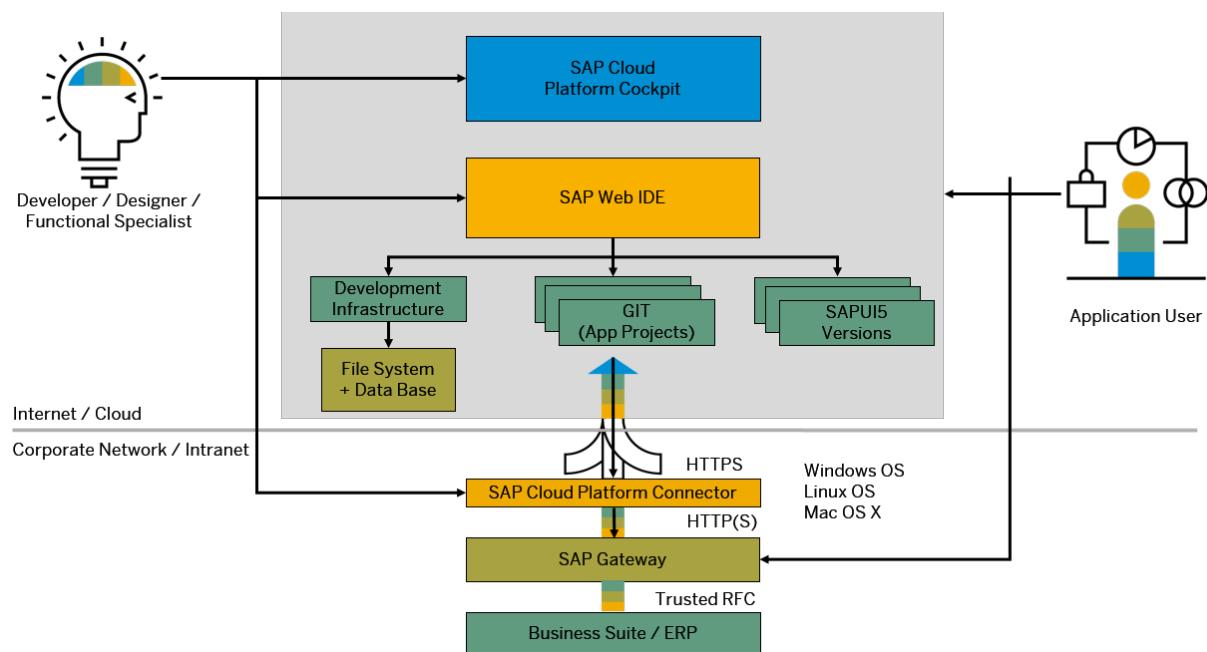
# 3 Overview

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

## Architecture

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



| Component                  | Description  |
|----------------------------|--|
| SAP 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.<br>For more information, see <a href="#">SAP Cloud Platform cockpit</a> . |

| Component                    | Description  |
|------------------------------|--|
| SAP Web IDE application      | Integrated development environment used to create or extend SAP UI5 or SAP Fiori applications.   |
| Git                          | Revision control and source code management system.  |
| SAPUI5                       | User interface technology that is used to build and adapt client applications.<br><br>For more information, see <a href="#">UI development toolkit for HTML5 - Demo Kit</a>                                  |
| SAP Cloud Platform connector | Allows SAP Web IDE and SAP Cloud Platform to connect to an on-premise system securely and with minimal configuration effort.<br><br>For more information, see <a href="#">SAP Cloud Platform connector</a> . |
| 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.

## 4 Features and Benefits

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

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

| Key Feature                       | Use  |
|-----------------------------------|--|
| Zero installation and maintenance | Available as a cloud service on SAP 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.  |

| Key Feature                | Use  |
|----------------------------|--|
| Visual UI Editors          | <ul style="list-style-type: none"> <li>• Layout Editor: The layout editor makes developing SAPUI5 applications faster and simpler. It has the following areas:           <ul style="list-style-type: none"> <li>◦ Palette: Contains the SAPUI5 controls. Expand or collapse the sections by clicking the arrows, and then you can drag and drop the controls to the canvas.</li> <li>◦ Canvas: Displays the content of the XML view in a way that closely corresponds to how it will appear in your finished application.</li> <li>◦ Properties/Data: Shows the properties' values and data binding of a selected control.</li> <li>◦ Outline pane: Displays the outline of your XML view.</li> </ul> </li> <li>• UI Adaptation: The UI Adaptation editor provides an intuitive user interface to make changes to SAP Fiori element applications.</li> </ul> <p>Storyboard</p> |
| 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.  |

# 5 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.

# 6 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.

## 6.1 Git Client Plugin

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

### Related Information

[Using Source Control \(Git\) \[page 283\]](#)

## 6.2 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.

# 7 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.

## 7.1 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.

## 7.2 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.

# 8 Getting Started

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

The checklist assumes that you already have:

- Decided to use SAP Web IDE Full-Stack. For more information on this version, see [Introducing SAP Web IDE Full-Stack \[page 16\]](#).
- Signed up for an SAP Cloud Platform global account, either trial or enterprise. For more information, see [Getting a Global Account](#).

## i Note

We recommend all developers working on the same project use the same SAP Web IDE version. If you choose different cloud versions or use the personal edition together with the full-stack development version, you can migrate the project between SAP Web IDE versions as described in [Import Projects from the Previous Version of SAP Web IDE \[page 62\]](#). However, all changes to the project settings must be copied manually.

| Step  | Description   | Links/Information  |
|---|---|--|
| <input type="checkbox"/> <b>Enable principal propagation</b><br>(production only) | In order to support the SSO solution of SAP Web IDE, you need to configure your account to allow principal propagation. | <a href="#">Principal Propagation [page 419]</a>                                   |
| <input type="checkbox"/> <b>Enable SAP Web IDE Full-Stack</b>                     | In the cockpit, go to ► <a href="#">Services</a> ► <a href="#">SAP Web IDE Full-Stack</a> ▶ and enable the service.     |  |
| <input type="checkbox"/> <b>Grant user permissions</b>                            | To enable working with SAP Web IDE, developers need to be assigned the <a href="#">DiDeveloper</a> role.                | <a href="#">Assign Users Permission for SAP Web IDE [page 413]</a>                 |
| <input type="checkbox"/> <b>Connect to backend systems</b><br>(optional)          | If you need to use ABAP backend services in your applications, create destinations to the services.                     | <a href="#">Connecting to ABAP Systems</a>   |
| <input type="checkbox"/> <b>Migrate projects</b>                                  | If you have existing projects in SAP Web IDE, you can migrate them to SAP Web IDE Full-Stack.                           | <a href="#">Import Projects from the Previous Version of SAP Web IDE [page 62]</a> |

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

| Step   | Description  | Links/Information  |
|--|--|--|
| <input type="checkbox"/> Make sure you have an extra 1GB of Application Runtime in your development space.   |  | <p>To add additional Application Runtime:</p> <ul style="list-style-type: none"> <li>• Customers - Please contact your account executive (AE).</li> <li>• Partners - Please contact your Partner Service Advisor (PSA).</li> </ul> |
| <input type="checkbox"/> <b>Create subaccounts in the SAP Cloud Platform Cloud Foundry environment.</b><br><br>Select a region according to the guidelines in the <a href="#">Neo and Cloud Foundry Regions [page 32]</a> topic. | <p>When you create a subaccount in the SAP Cloud Platform Cloud Foundry environment, a Cloud Foundry organization is automatically created for that subaccount.</p> <p>We recommend you create subaccounts for development, staging/test and production purposes.</p>  | <a href="#">Creating Subaccounts</a>   |
| <input type="checkbox"/> Create spaces   | <p>You can create and delete spaces in a Cloud Foundry organization using the SAP Cloud Platform cockpit or the console client (Cloud Foundry command line interface).</p> <p>We recommend at least one space for a development team working on the same project (that is, one space per project).</p> <p>For staging/test and production organizations, one space is sufficient.</p>                  | <a href="#">Managing Spaces</a>  |
| <input type="checkbox"/> Assign members to your Cloud Foundry organizations and spaces   | <p>Enable your developers to work with your SAP Cloud Platform Cloud Foundry environments.</p> <p>Your developers should be assigned to the space developer role to be able to use the space from SAP Web IDE.</p>   | <a href="#">Managing Members</a>   |
| <input type="checkbox"/> Connect to Cloud Foundry services (optional)  | <p>You can create destinations to Cloud Foundry services and consume them in the applications you create with SAP Web IDE.</p> <p>For each service, you need to create a destination for each space to which the service is deployed.</p> <p>The step is optional at this point. You can set up your destinations now as you get started, or you can create them when you create new applications.</p> | <a href="#">Connect to Cloud Foundry Services [page 27]</a><br><a href="#">Create Destinations for Cloud Foundry Services [page 379]</a>   |

## 8.1 Open SAP Web IDE

You open SAP Web IDE in a web browser.

### Prerequisites

To work with SAP Web IDE, you must be assigned the `DiDeveloper` role. For more information, see [Assign Users Permission for SAP Web IDE \[page 413\]](#).

### 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 URLs listed in [Regions and Hosts](#).
2. In the navigation pane, choose [Services](#).
3. On the [Services](#) pane, select [SAP Web IDE Full-Stack](#).
4. Click [Open SAP Web IDE Full-Stack](#), and then [Go to Service](#).

The following browsers are supported:

#### i Note

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

- Microsoft Internet Explorer (version 11)
- Microsoft Edge
- Mozilla Firefox
- Google Chrome
- Safari (on iOS platforms only)

#### i Note

Opening multiple instances of SAP Web IDE in parallel may cause issues. For more information, see [Known Issues \[page 429\]](#).

## 8.2 Connect to Cloud Foundry Services

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

### Prerequisites

You must be an administrator in your SAP Cloud Platform account.

### Procedure

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

Set the following parameters:

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

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

- **TrustAll:** `true`
- **WebIDEEEnabled:** `true`
- **WebIDESystem:** `API`
- **WebIDEUsage:** `odata_gen`

2. Establish trust from your SAP Cloud Platform account (where you are running SAP Web IDE) to your SAP Cloud Platform Cloud Foundry organization, as described in [Principal Propagation from the Neo to the Cloud Foundry Environment](#).

### Troubleshooting

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

- **500:**

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

## 8.3 Connect to ABAP Systems

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

### Prerequisites

- You must have an Administrator or Developer role to modify destinations.
- Make sure you are using OData Version 2.0 only.
- If your remote system resides behind a firewall (proxy type `OnPremise`), the following prerequisites must be met:
  - You have set up the Cloud Connector and defined a virtual host mapping for the system.  
See [Cloud Connector](#).
  - In the *Access Control* tab page of 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_uis`  
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-usage destination. You enter the required usages as a value for the property `WebIDEUsage`.

## Procedure

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

**i Note**

You can access the SAP Cloud Platform cockpit from SAP Web IDE by selecting ► [Tools](#) ► [SAP Cloud Platform Cockpit](#).

2. Open the [Destinations](#) editor.
3. Choose [New Destination](#).
4. Configure the destination as follows:

| Property              | Value  |
|-----------------------|--|
| <b>Name</b>           | Provide a name for the destination that includes the desired service.  |
| <b>Type</b>           | HTTP   |
| <b>Description</b>    | Optional - Provide a description for the destination   |
| <b>URL</b>            | <code>&lt;protocol&gt;://&lt;host&gt;:&lt;port&gt;</code><br><code>&lt;protocol&gt;://&lt;virtual host&gt;:&lt;virtual port&gt;</code> (if you are using SAP Cloud Platform connector)   |
|                       | <p><b>i Note</b></p> <ul style="list-style-type: none"><li>o Do not add the URL path for the usage (for example, /sap/opu/odata).</li><li>o 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.</li></ul>          |
| <b>Proxy Type</b>     | Select <a href="#">Internet</a> or <a href="#">OnPremise</a> , depending on the connection you need to provide for your application. <ul style="list-style-type: none"><li>o If your remote system is publicly accessible on the internet, select <a href="#">Internet</a>.</li><li>o If your remote system resides behind a firewall, select <a href="#">OnPremise</a>.</li></ul> |
| <b>Authentication</b> | Select <a href="#">No Authentication</a> or <a href="#">Basic Authentication</a> depending on the authentication you need for the connection. For more information, see <a href="#">Creating HTTP Destinations</a> .   |

5. In the [Additional Properties](#) section, choose [New Property](#) for each of the properties described below.

| Key                | Value   |
|--------------------|---|
| <b>WebIDEUsage</b> | Enter one or more of the following values: <ul style="list-style-type: none"><li>o odata_abap:<br/>for the OData functionality of Gateway (corresponds to URL path /sap/opu/odata)</li><li>o odata_gen:<br/>for generic OData functionality (service URL must be provided manually in the New Project wizard)</li><li>o ui5_execute_abap:</li></ul> |

| Key                               | Value  |
|-----------------------------------|--|
|                                   | <p>for executing SAPUI5 applications from the SAPUI5 ABAP Repository (corresponds to URL path /sap/bc/ui5_ui5)</p> <ul style="list-style-type: none"> <li>○ <code>dev_abap</code>:<br/>for extensibility scenarios and developing or deploying to SAPUI5 ABAP Repository (corresponds to URL path /sap/bc/adt)</li> <li>○ <code>bsp_execute_abap</code>:<br/>for working with fact sheets (corresponds to URL path /sap/bc/bsp)</li> <li>○ <code>plugin_repository</code>:<br/>for exposing optional plugin repositories (corresponds to URL path /plugins/pluginrepository)</li> <li>○ <code>odata_xs</code>:<br/>for SAP HANA XS OData services (corresponds to URL path /sap/hba)</li> <li>○ <code>api_mgmt_catalog</code>:<br/>for the API management system</li> <li>○ <code>api_mgmt_proxy</code>:<br/>for the API's endpoint</li> <li>○ <code>odata_smp</code>:<br/>for accessing OData Services in an SAP Mobile Platform system</li> <li>○ <code>odata_hcp_odp</code>:<br/>for accessing OData services in an OData Provisioning system on Cloud Platform</li> <li>○ <code>odata_hci</code>:<br/>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.</li> <li>○ <code>hci_ifl_node</code>:<br/>for setting up the integration flow node in the HANA Cloud Integration system on Cloud Platform</li> <li>○ <code>smart_business_odata</code>:<br/>for smart business OData services</li> <li>○ <code>smart_business_gen</code>:<br/>for smart business annotation generation services</li> </ul> |
|                                   | <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>○ In order to import an application from your ABAP system and execute it, the following WebIDEUsage property values are needed: <code>dev_abap</code>, <code>odata_abap</code>, and <code>ui5_execute_abap</code>.</li> <li>○ When you enter multiple usages for a destination, separate them by commas without spaces (for example, <code>odata_abap,ui5_execute_abap</code>).</li> </ul>   |
| <code>WebIDEEEnabled</code>       | true   |
| <code>WebIDESystem</code>         | <SAP system_ID>  |
| <code>sap-client</code>           | <SAP client from the ABAP system>  |
| <code>WebIDEAdditionalData</code> | <p><b>This property is optional.</b></p> <p>You can add one or more of the following values:</p> <ul style="list-style-type: none"> <li>○ <code>api_mgmt</code></li> </ul>   |

| Key | Value   |
|-----|---|
|     | <p>for specifying an API key to retrieve metadata</p> <ul style="list-style-type: none"> <li>○ <b>full_url</b><br/>for specifying the full URL of a service so that metadata is shown without having to choose the service</li> </ul> <div style="background-color: #fdf5e6; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></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> </div> |

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

## Related Information

[SAP Cloud Platform connector](#)

[Creating HTTP Destinations](#)

### 8.3.1 Requirements for Connecting to ABAP Systems

The following is prerequisite information for connecting to ABAP systems.

- You have installed *SAP\_BASIS* 7.31 SP 14 or 7.40 SP 8 or later.

**i Note**

If you are using earlier releases of *SAP\_BASIS* than those mentioned above, you must apply the following SAP Notes:

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

- If you are using *SAP\_BASIS* 7.31, make sure that the software components of the UI add-on for SAP NetWeaver (*UI\_INFRA* and *UI5\_731*) are installed.

## Related Information

[Connect to ABAP Systems \[page 28\]](#)

[Connect Remote Systems in SAP Web IDE Personal Edition \[page 405\]](#)

## 8.4 Neo and Cloud Foundry Regions

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

Region Mapping for Enterprise Accounts

| Neo Region<br>(Operated by SAP) | Host                                       | Recommended Cloud Foundry Region  |
|---------------------------------|--|---|
| Australia (Sydney 1)            | ap1.hana.ondemand.com                      | Europe (Frankfurt)<br><a href="https://api.cf.eu10.hana.ondemand.com">https://api.cf.eu10.hana.ondemand.com</a> |
| Australia (Sydney 2)            | ap2.hana.ondemand.com                      | Europe (Frankfurt)<br><a href="https://api.cf.eu10.hana.ondemand.com">https://api.cf.eu10.hana.ondemand.com</a> |
| Brazil (São Paulo)              | br1.hana.ondemand.com                      | US East (VA)<br><a href="https://api.cf.us10.hana.ondemand.com">https://api.cf.us10.hana.ondemand.com</a>       |
| Canada (Toronto)                | ca1.hana.ondemand.com                      | US East (VA)<br><a href="https://api.cf.us10.hana.ondemand.com">https://api.cf.us10.hana.ondemand.com</a>       |
| China (Shanghai)                | cn1.hana.ondemand.com                      | Europe (Frankfurt)<br><a href="https://api.cf.eu10.hana.ondemand.com">https://api.cf.eu10.hana.ondemand.com</a> |
| Europe (Rot/Germany)            | hana.ondemand.com<br>eu1.hana.ondemand.com | Europe (Frankfurt)<br><a href="https://api.cf.eu10.hana.ondemand.com">https://api.cf.eu10.hana.ondemand.com</a> |
| Europe (Amsterdam/Netherlands)  | eu2.hana.ondemand.com                      | Europe (Frankfurt)<br><a href="https://api.cf.eu10.hana.ondemand.com">https://api.cf.eu10.hana.ondemand.com</a> |
| Russia (Moscow)                 | ru1.hana.ondemand.com                      | Europe (Frankfurt)<br><a href="https://api.cf.eu10.hana.ondemand.com">https://api.cf.eu10.hana.ondemand.com</a> |
| Japan (Tokyo)                   | jp1.hana.ondemand.com                      | Europe (Frankfurt)<br><a href="https://api.cf.eu10.hana.ondemand.com">https://api.cf.eu10.hana.ondemand.com</a> |
| US East (Ashburn/VA)            | us1.hana.ondemand.com                      | US East (VA)<br><a href="https://api.cf.us10.hana.ondemand.com">https://api.cf.us10.hana.ondemand.com</a>       |
| US West (Chandler)              | us2.hana.ondemand.com                      | US East (VA)<br><a href="https://api.cf.us10.hana.ondemand.com">https://api.cf.us10.hana.ondemand.com</a>       |
| US East (Sterling/VA)           | us3.hana.ondemand.com                      | US East (VA)<br><a href="https://api.cf.us10.hana.ondemand.com">https://api.cf.us10.hana.ondemand.com</a>       |

#### Region Mapping for Trial Accounts

| Neo Region<br><b>(Operated by SAP)</b> | Host                   | Recommended Cloud Foundry Region  |
|--|------------------------|---|
| Europe (Rot/Germany) - Trial           | hanatrial.ondemand.com | Europe (Frankfurt)<br><a href="https://api.cf.eu10.hana.ondemand.com">https://api.cf.eu10.hana.ondemand.com</a> |

# 9 Setting User Preferences

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

## Procedure

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

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

| Preferences  | More Information  |
|--|---|
| Code check settings to enable and disable inline code validation. The project-level appearance settings are used to validate code and flag messages.   | <a href="#">Configure Code Checking [page 113]</a>  |
| Code completion settings to enable inline code completion and comment completion.  | <a href="#">Configure Code Completion [page 95]</a>   |
| 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. | <a href="#">Configure the Code Editor [page 90]</a>   |
| 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  <a href="#">Tools</a>  <a href="#">Preferences</a>  <a href="#">Default Editors</a>  . |
| Set up Git for source control.   | <a href="#">Set Up Git [page 287]</a>   |
| Customize SAP Web IDE keyboard shortcuts.  | <a href="#">Customize Keyboard Shortcuts [page 41]</a>  |
| Enable plugins to create and develop applications.   | <a href="#">Enable Additional Features [page 392]</a>   |

| Preferences                             | More Information  |
|---|---|
| Delete or ignore workspace persistence. | <p>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.</p> <p>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 <b>settings=ignore</b> 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).</p> <p>To delete the persistence information, add the parameter <b>settings=delete</b> to the URL.</p> |

# 10 SAP Web IDE Basics

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

## [Navigating SAP Web IDE \[page 36\]](#)

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

## [Keyboard Shortcuts \[page 38\]](#)

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

## [Workspace Search Options \[page 42\]](#)

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

## [Resizing Panes \[page 48\]](#)

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

## 10.1 Navigating SAP Web IDE

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

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

### Development Perspective

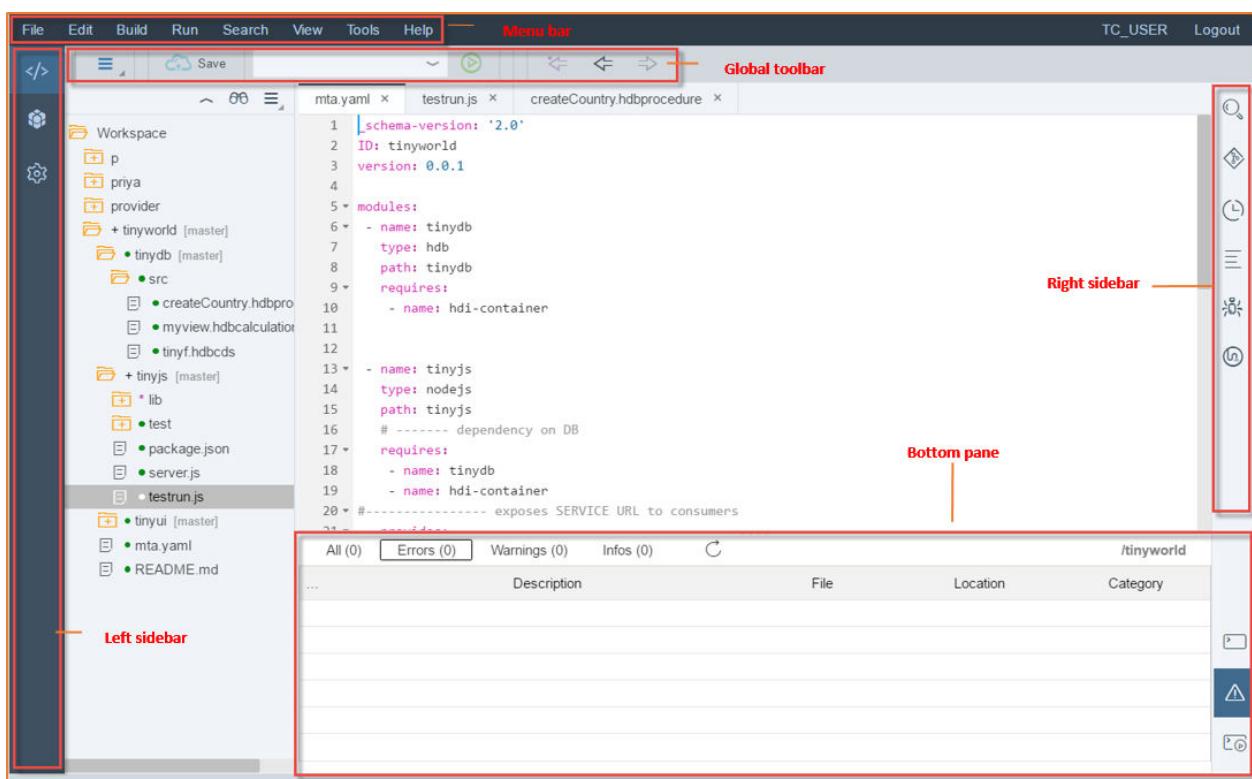
**Menu bar** - Provides access to all operations available in SAP Web IDE.

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

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

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

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



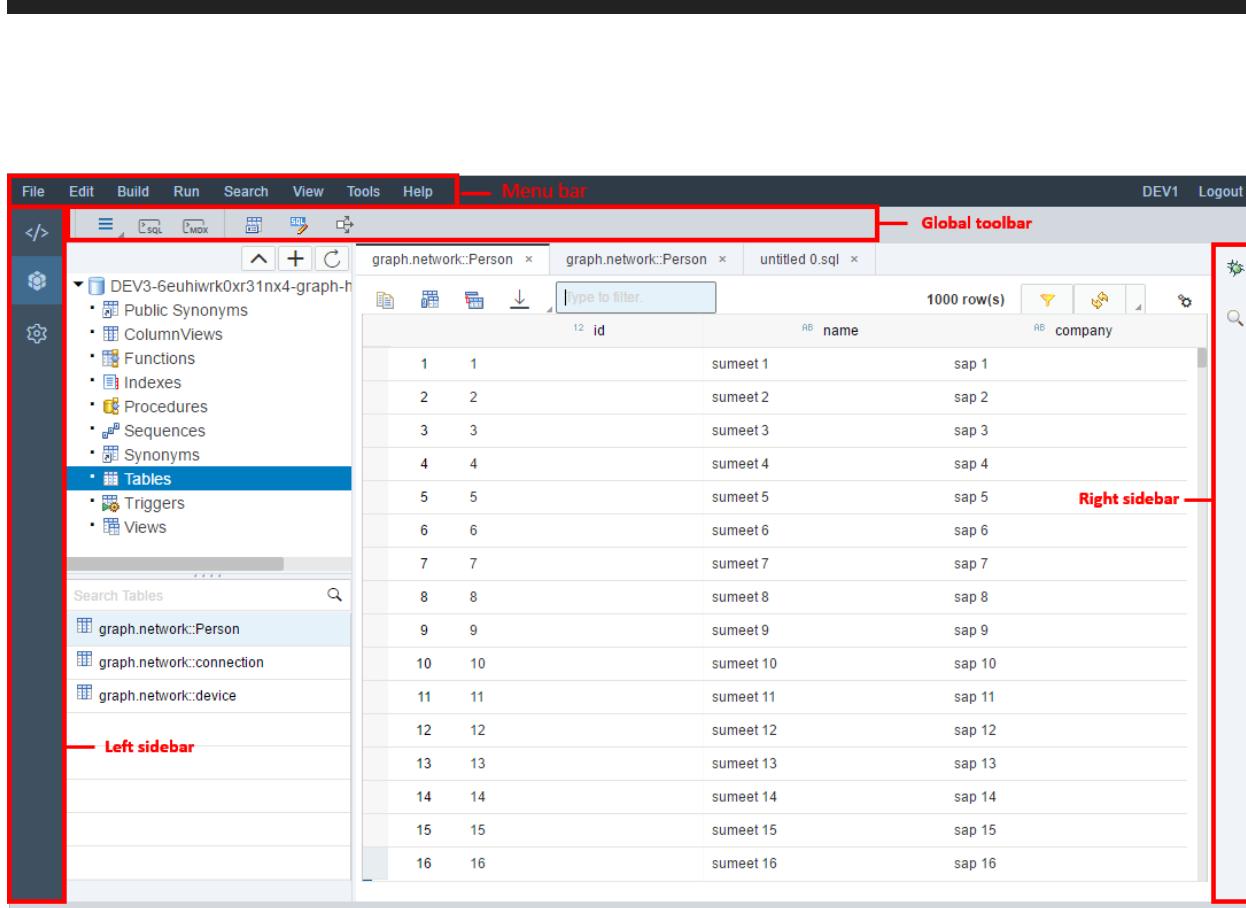
## Database Explorer Perspective

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

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

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

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



**Parent topic:** [SAP Web IDE Basics \[page 36\]](#)

## Related Information

[Keyboard Shortcuts \[page 38\]](#)

[Workspace Search Options \[page 42\]](#)

[Resizing Panes \[page 48\]](#)

[Working with the Database Explorer](#)

## 10.2 Keyboard Shortcuts

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

The following shortcuts are available in SAP Web IDE:

### 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 \[page 41\]](#).

| 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] |
| 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] + [I]                       |
| 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]                             |

| Action                              | Microsoft Windows Keyboard Shortcut | Mac OS Keyboard Shortcut          |
|-------------------------------------|-------------------------------------|-----------------------------------|
| Outdent line                        | [Shift] + [Tab]                     | [Shift] + [Tab]                   |
| Move lines up                       | [Alt] + [Up Arrow]                  | [Shift] + [Tab]                   |
| Move lines down                     | [Alt] + [Down Arrow]                | [Alt] + [Down Arrow]              |
| Copy lines up                       | [Alt] + [Shift] + [Up Arrow]        | [Alt] + [Shift] + [Up Arrow]      |
| Copy lines down                     | [Alt] + [Shift] + [Down Arrow]      | [Alt] + [Shift] + [Down Arrow]    |
| Beautify file format                | [Ctrl] + [Alt] + [B]                | [Command] + [Alt] + [B]           |
| Generate JSDoc Comment              | [Ctrl] + [Alt] + [J]                | [Command] + [Alt] + [J]           |
| Goto JavaScript definition          | [Ctrl] + [Alt] + [G]                | [Command] + [Alt] + [G]           |
| Run                                 | [Alt] + [Shift] + [R]               | [Alt] + [Shift] + [R]             |
| Run without frame                   | [Ctrl] + [Alt] + [Shift] + [R]      | [Command] + [Alt] + [Shift] + [R] |
| Find                                | [Ctrl] + [F]                        | [Command] + [F]                   |
| Find and replace                    | [Ctrl] + [H]                        | [Command] + [H]                   |
| Find references                     | [Ctrl] + [Alt] + [W]                | [Ctrl] + [Alt] + [W]              |
| Advanced repository search          | [Ctrl] + [Shift] + [F]              | [Command] + [Shift] + [F]         |
| Maximize/Restore Active Editor      | [Ctrl] + [M]                        | [Command] + [M]                   |
| View console                        | [Ctrl] + [Shift] + [M]              | [Command] + [Shift] + [M]         |
| View Git pane                       | [Ctrl] + [Shift] + [V]              | [Command] + [Shift] + [V]         |
| View outline                        | [Ctrl] + [Shift] + [U]              | [Command] + [Shift] + [U]         |
| View Problems                       | [Ctrl] + [Alt] + [P]                | [Command] + [Alt] + [P]           |
| Open Preferences perspective        | [Ctrl] + [.]                        | [Command] + [.]                   |
| Open Extensibility pane             | [Ctrl] + [Shift] + [E]              | [Command] + [Shift] + [E]         |
| Open Resource                       | [Ctrl] + [Shift] + [R]              | [Command] + [Shift] + [R]         |
| Open <i>Quick Access</i> dialog box | [Ctrl] + [3]                        | [Command] + [3]                   |
| Go to line                          | [Ctrl] + [L]                        | [Command] + [L]                   |
| Refactor                            | [Alt] + [J]                         | [Alt] + [J]                       |
| Open i18n                           | [Alt] + [I]                         | [Alt] + [I]                       |

Parent topic: [SAP Web IDE Basics \[page 36\]](#)

## Related Information

[Navigating SAP Web IDE \[page 36\]](#)

[Workspace Search Options \[page 42\]](#)

[Resizing Panes \[page 48\]](#)

### 10.2.1 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.

##### 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 \[page 38\]](#)

## 10.3 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 \[page 43\]](#)

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

[Search for Files or Content in the Workspace \[page 44\]](#)

Perform a file or string search within a folder or across all projects in your workspace.

[Replace Strings Across Multiple Files \[page 45\]](#)

Perform an advanced find and replace across multiple files with the Search pane.

[Find References \[page 47\]](#)

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 \[page 36\]](#)

## Related Information

[Navigating SAP Web IDE \[page 36\]](#)

[Keyboard Shortcuts \[page 38\]](#)

[Resizing Panes \[page 48\]](#)

## 10.3.1 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 \[page 45\]](#).

### 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 \[page 42\]](#)

### Related Information

[Search for Files or Content in the Workspace \[page 44\]](#)

[Replace Strings Across Multiple Files \[page 45\]](#)

[Find References \[page 47\]](#)

## 10.3.2 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 \[page 42\]](#)

## Related Information

[Find and Replace in an Open File \[page 43\]](#)

[Replace Strings Across Multiple Files \[page 45\]](#)

[Find References \[page 47\]](#)

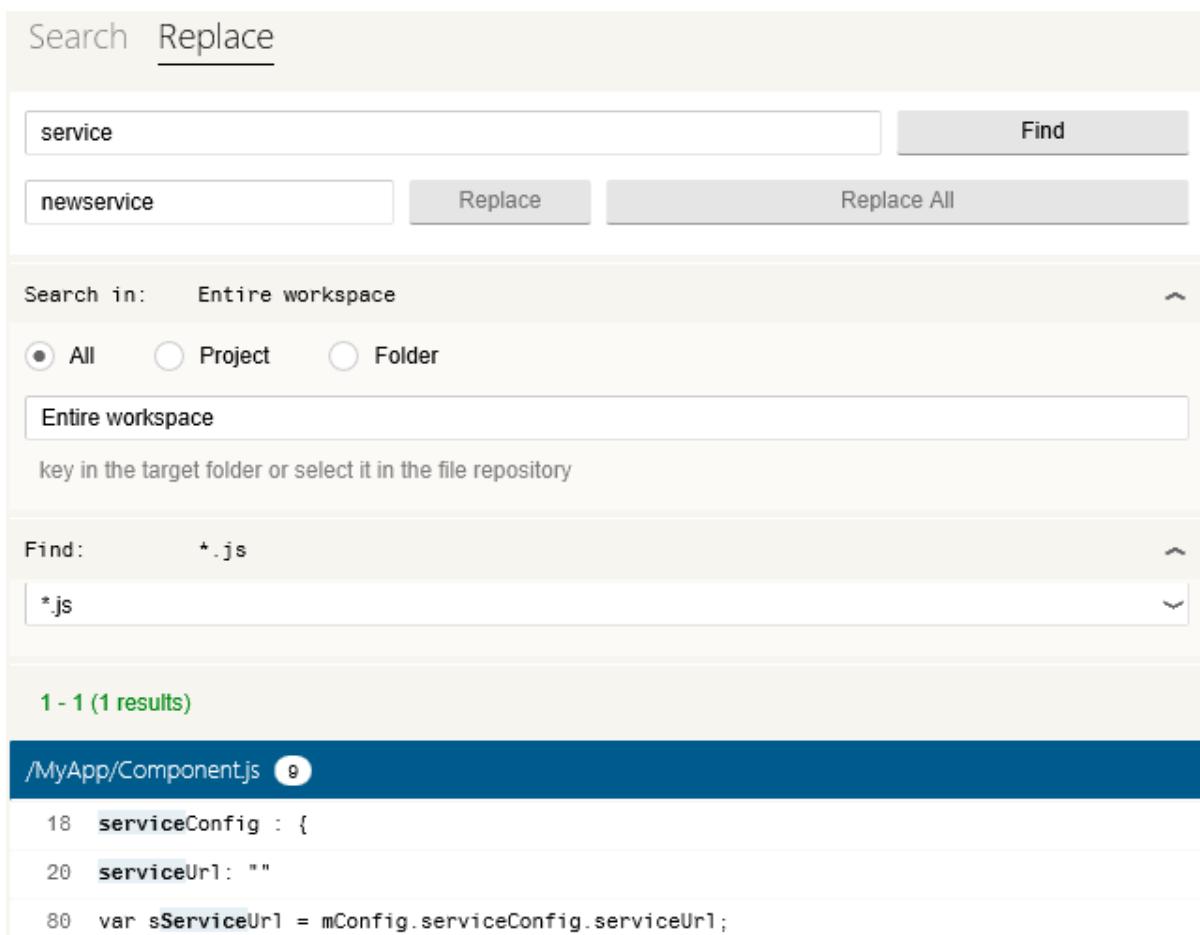
### 10.3.3 Replace Strings Across Multiple Files

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.

48 "attributes": ["Name", "sap:content-version"], 

○ A global replace of all results in the list. Choose *Replace All*.

 Note

You cannot undo the global replace operation.

**Task overview:** [Workspace Search Options \[page 42\]](#)

## Related Information

[Find and Replace in an Open File \[page 43\]](#)

---

[Search for Files or Content in the Workspace \[page 44\]](#)

[Find References \[page 47\]](#)

## 10.3.4 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 \[page 42\]](#)

### Related Information

[Find and Replace in an Open File \[page 43\]](#)

[Search for Files or Content in the Workspace \[page 44\]](#)

[Replace Strings Across Multiple Files \[page 45\]](#)

## 10.4 Resizing Panes

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

### Context

You can make the following adjustments:

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

#### Note

Double-click again to restore the editor window to its original size. This always 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 \[page 36\]](#)

### Related Information

[Navigating SAP Web IDE \[page 36\]](#)

[Keyboard Shortcuts \[page 38\]](#)

[Workspace Search Options \[page 42\]](#)

# 11 Developing

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



- [Developing Multi-Target Applications \[page 352\]](#)
- [Developing Web Applications \[page 50\]](#)

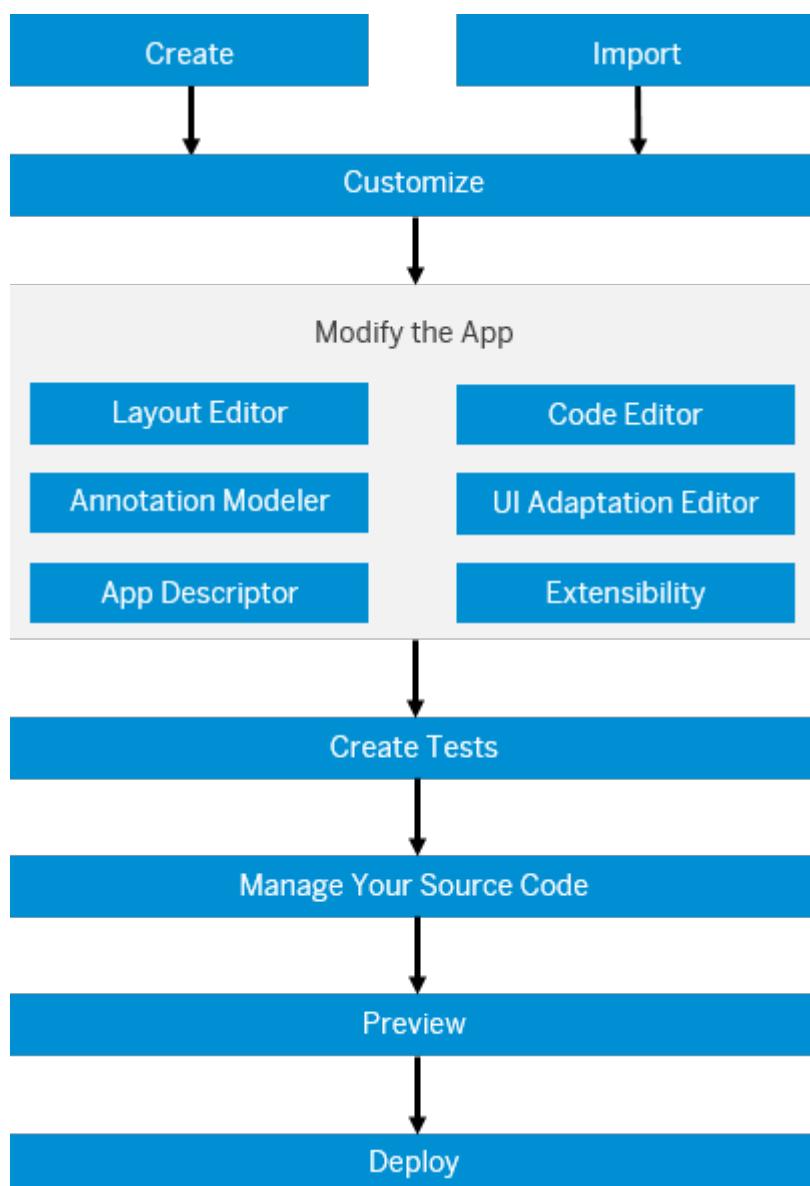
## Related Information

[Developing Web Applications \[page 50\]](#)

[Developing Multi-Target Applications \[page 352\]](#)

## 11.1 Developing Web Applications

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



- [Importing Projects \[page 61\]](#)
- [Creating Projects \[page 51\]](#)
- [Customizing Your Project \[page 66\]](#)
- [Layout Editor \[page 207\]](#)
- [Developing Applications \[page 86\]](#)
- [Annotation Modeler \[page 245\]](#)

- 
- [UI Adaptation Editor \[page 276\]](#)
  - [Extending SAPUI5 Applications \[page 338\]](#)
  - [Developing Application Tests \[page 206\]](#)
  - [Using Source Control \(Git\) \[page 283\]](#)
  - [Running Applications in Development Mode \[page 311\]](#)
  - [Modifying the Application Descriptor Configuration \[page 78\]](#)
  - [Deploying Applications \[page 329\]](#)

## 11.1.1 Creating Projects

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

### [Create Projects from a Template \[page 51\]](#)

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

### [Create Projects from a Sample Application \[page 60\]](#)

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

### [Add a New Component \[page 60\]](#)

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

## Related Information

[Creating a Quick Start Application with the Layout Editor \[page 224\]](#)

### 11.1.1.1 Create Projects from a Template

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

## Procedure

1. From the *File* menu, choose *New* *Project from Template*.
2. Select the relevant template based on the project that you want to generate 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*.

**i Note**

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

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

**i Note**

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

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

**i Note**

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

For more information, see [Select a Data Source \[page 54\]](#).

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

**i Note**

If you select an OData service, a `model` folder containing the `metadata.xml` file is 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 \[page 311\]](#).

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:

- Choose *From Service* to select one or more annotation files provided by the OData service.

**i Note**

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

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

**i Note**

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

7. Choose *Next*.
8. If relevant for your template, in the 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.

## 11.1.1.1 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](#).

## 11.1.1.2 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](#).

### 11.1.1.3 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 \[page 28\]](#).

#### 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*.

### **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 \[page 28\]](#).

2. Select a service.

### **Note**

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

3. To explore the selected service:

- a. Expand the service in the service table to view its elements, including entity sets and function imports.
- b. To get more information about the service, choose *Show Details*.

### **Restriction**

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

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

| Service Details | Description  |
|-----------------|--|
| Name            | Name of the OData service  |
| Description     | Description of the OData service and its purpose   |
| Status          | Indicates the status of the OData service. The typical ones that you can come across are: <ul style="list-style-type: none"><li>○ Running</li><li>○ Authentication failed</li><li>○ Service is unavailable</li></ul> |
| 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.  |

| Service Details   | Description                                      |
|-------------------|--|
| 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                |

### Note

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

- Name
- Status
- OData Version
- URL

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

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

- f. Choose [Annotations](#) in the right pane. If the selected service uses SAP Annotations, the operations that are supported by the entity set are displayed here. For example, **Creatable** and **Updatable**.
- g. Choose [Live Data](#) in the right pane to see live production data associated with the entity set. For more information, see [View Live Data \[page 57\]](#).

- h. To get more information about a specific property, select it in the service table and choose [Show Details](#). You can see detailed information about the property to the right of the service table.
- i. To get more information about a function import, select it in the service table and choose [Show Details](#). You can see the following detailed information about the function import to the right of the service table.

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

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

4. Once you have selected the required OData service, proceed to the next step. If you are creating a project from a template, see [Create Projects from a Template \[page 51\]](#) for more information.

### 11.1.1.3.1 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*).

 Note

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

4. To customize your view of the data in the entity set, choose  (*Settings*). The *View Settings* dialog box appears.
5. You can perform any of the following customizations:
- On the *Columns* tab, select only those properties that you wish to see as part of the live data. You can also search for a property by typing its name in the text box.  
Choose *Show Selected* to see only the selected properties in the tab. This proves handy, especially when you are dealing with many properties.
  - On the *Columns* tab, select a property and choose  (*Move to Top*),  (*Move Up*),  (*Move Down*), or  (*Move to Bottom*) to rearrange its position in the live data view.
  - On the *Sort* tab, select the property using which you want to sort the data. Also, select whether you want to sort it in the ascending or descending order.

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

- 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 \[page 54\]](#)

## 11.1.1.4 Consume APIs from SAP API Business Hub

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

### Procedure

1. Create your project using the required template.

**i Note**

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

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

**i Note**

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

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

## 11.1.1.2 Create Projects from a Sample Application

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

### Procedure

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

#### 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.

## 11.1.1.3 Add a New Component

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

### Procedure

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

### **i** Note

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

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

### **i** Note

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

### **i** Note

If the data source is selected from a local file (using *Browse*), the created application project can be run only using Mock Data, unless the service URL is manually added to the generated application code. For more information, see [Running Applications in Development Mode \[page 311\]](#).

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.

## 11.1.2 Importing Projects

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

### [Import Projects from an Archive \[page 62\]](#)

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

### [Import Projects from the Previous Version of SAP Web IDE \[page 62\]](#)

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

### [Import Applications from SAP Cloud Platform \[page 64\]](#)

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

### [Import Applications from the SAPUI5 ABAP Repository \[page 65\]](#)

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

## 11.1.2.1 Import Projects from an Archive

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

### Prerequisites

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

### Procedure

1. From the workspace, select the folder to which you want to import the project, for example, the top-level `Workspace` folder.
  2. Choose  `File`  `Import` .
- The `Import` dialog box is displayed, and the selected folder is displayed in the `Import to` field. You can change the destination folder by choosing `Select Folder` and browsing to the required folder, or typing in a new path to create a new folder.
3. In the `File` field, browse to the location of your archived project file, select it, and choose `Open`.
  4. Choose `OK`. The project is created in a new folder with the same name as the archive file.

#### Note

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

## 11.1.2.2 Import Projects from the Previous Version of SAP Web IDE

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

### Context

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

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

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

### Caution

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

### Note

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

- **Application Build:** If you used the SAPUI5 client-side build in the standard SAP Web IDE to minify and bundle files before deployment, in the Full-Stack Development version of SAP Web IDE, you must use the Grunt build. For more information, see [Application Build \[page 337\]](#) and [Building Applications \[page 325\]](#).

### Note

We recommend all developers working on the same project use the same SAP Web IDE version. If you choose different cloud versions or use the personal edition together with the full-stack development version, you can migrate the project between SAP Web IDE versions as described in [Import Projects from the Previous Version of SAP Web IDE \[page 62\]](#). However, all changes to the project settings must be copied manually.

## Procedure

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

## Next Steps

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

### Note

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

## Related Information

[Introducing SAP Web IDE Full-Stack \[page 16\]](#)

### 11.1.2.3 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.

## 11.1.2.4 Import Applications from the SAPUI5 ABAP Repository

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

### Prerequisites

Make sure you have complied with all the items described in [Requirements for Connecting to ABAP Systems \[page 31\]](#)

### 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](#)

## 11.1.3 Customizing Your Project

Customize the developer experience for your SAP Web IDE project.

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

### [Configure Annotation Modeler to Use Mock Data \[page 66\]](#)

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

### [Customize JavaScript Beautifier Properties \[page 68\]](#)

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

### [Customize Code Checking Rules \[page 69\]](#)

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

### [Customize Code Checking Triggers \[page 69\]](#)

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

### [Define Application Languages \[page 70\]](#)

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 SAP Translation Hub \[page 70\]](#)

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

### [Configure Mock Data Usage \[page 72\]](#)

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

### [Set Project Types \[page 73\]](#)

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

### [Set the SAPUI5 Version \[page 74\]](#)

Configure the SAPUI5 version for your project.

### [Configure Run Configurations for the UI Adaptation Editor \[page 75\]](#)

Select the Run Configuration to be used when running your project in the UI Adaptation editor.

### [Use the Smart Business Service \[page 75\]](#)

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

## 11.1.3.1 Configure Annotation Modeler to Use Mock Data

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

### Prerequisites

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

## Context

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

### 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.

### 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.

## 11.1.3.2 Customize JavaScript Beautifier Properties

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

### Context

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

### Procedure

- From the context menu of any file in your project, select **Project Settings** **Beautifier** **JavaScript**.
- 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 \ns between tokens, but you set this value to <i>1</i> , the second \n 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).                             |

- 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> .      |

- Determine additional code formatting options:

| Option                                  | Description   |
|---|---|
| <i>Space before conditional</i>         | Select this option to insert spaces before conditional if/then 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> . |

- Choose **OK**.

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

### 11.1.3.3 Customize Code Checking Rules

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

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

- Customize the SAP Web IDE JavaScript validator configuration with your own settings. For more information, see [Customize JavaScript Validator Configuration \[page 118\]](#).
- Configure your own JavaScript code checking rules and use them for your project. For more information, see [Create JavaScript Code Checking Rules \[page 117\]](#).
- Create your own code checking plugin, then choose the plugin in the corresponding code checking configuration pane.

### 11.1.3.4 Customize Code Checking Triggers

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

#### Context

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

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

#### Procedure

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

#### Example

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

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

## 11.1.3.5 Define Application Languages

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

### Procedure

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

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

### Related Information

[Create Run Configurations \[page 314\]](#)

## 11.1.3.6 Use SAP Translation Hub

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

### Context

### Procedure

Before you can translate your project's `i18n.properties` file, you must configure your system to use SAP Translation Hub.

1. Make sure your project is attached to a Git repository.
2. Enable SAP Translation Hub for your account.
  - a. In the SAP Cloud Platform cockpit, choose *Services* in the navigation pane.
  - b. Choose *SAP Translation Hub*.

- c. Choose *Enable*.
3. Assign the roles of the users who will access the services in SAP Web IDE.
  - a. Choose *Configure Service* from the *Take Action* section.
  - b. In the pane on the left, choose *Roles*.
  - c. Assign the users requiring access to SAP Translation Hub. For more information, see [User Authentication and Authorization](#).

**i Note**

The remaining steps are now completed automatically for new SAP Web IDE users of SAP Translation Hub. Destinations and trust management settings are automatically created and assigned to new accounts. The following steps provide details of the default destination and trust management settings.

4. Create a destination for your account. In the pane on the left, choose *Destinations*, and create a destination for your account with the following data.

| Field                 | Entry  |
|-----------------------|--|
| <i>Name</i>           | <b>translationHubBeta</b>  |
| <i>Type</i>           | <b>HTTP</b>  |
| <i>Description</i>    | <b>Translation</b>   |
| <i>URL</i>            | <p>Enter the URL. URLs adhere to the following naming convention: <b>&lt;base URL of your SAP Translation Hub account&gt;/ui</b></p> <p>You can copy the required URL from SAP Translation Hub as follows:</p> <ol style="list-style-type: none"> <li>1. In the SAP Cloud Platform cockpit, choose <i>Services</i>.</li> <li>2. Choose <i>SAP Translation Hub</i> from the list of services</li> <li>3. Choose <i>Go To UI for Translation Workflow</i> and copy the URL directly from the address bar.</li> </ol> <p><b>i Note</b></p> <p>For more information, see <a href="#">Building Base URL of SAP Translation Hub</a>.</p> |
| <i>Proxy Type</i>     | <b>Internet</b>  |
| <i>Authentication</i> | <b>AppToAppSSO</b>   |

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

| Field                | Entry       |
|----------------------|-------------|
| <i>TrustAll</i>      | <b>true</b> |
| <i>WebIDEEnabled</i> | <b>true</b> |

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

# Translating your Properties File

## Context

### Note

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

## Procedure

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

### Example

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

## 11.1.3.7 Configure Mock Data Usage

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

## Context

### 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.

### 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.

## 11.1.3.8 Set Project Types

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

## Context

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

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

## Procedure

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

### 11.1.3.9 Set the SAPUI5 Version

Configure the SAPUI5 version for your project.

## Context

The SAPUI5 version you choose affects the following:

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

## Procedure

1. From the project context menu, select *Project Settings*.
2. In the *SAPUI5 Settings* pane, select the design-time SAPUI5 version for your project.

You can also select *latest*, which uses the latest released version of SAPUI5.

## 11.1.3.10 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

1. From the project context menu, select *Project Settings*.
2. 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.

3. Select the configuration to be used when running the app in the UI Adaptation editor.
4. 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.

## 11.1.3.11 Use the Smart Business Service

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

### Prerequisites

You must have an administrator role to perform this procedure.

## Context

## Procedure

1. Subscribe to the Smart Business service in SAP Cloud 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:

- `WebIDEEnabled = true`
- `WebIDEUsage = smart_business_gen`

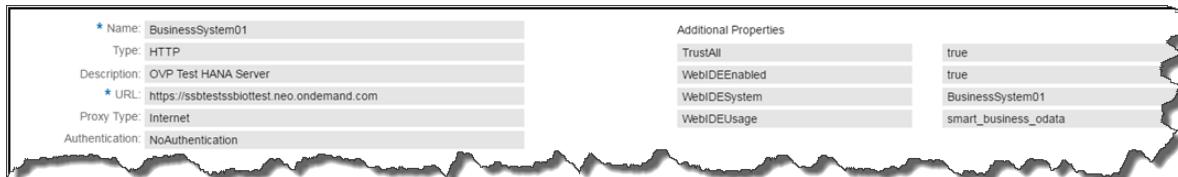
This destination is used to generate the smart business annotation.

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

The service must have the following additional properties:

- `WebIDEEnabled = 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 interface for selecting a service. On the left, a sidebar titled 'Sources' lists 'Service Catalog', 'Workspace', 'File System', and 'Service URL'. The 'Service URL' option is highlighted with a dark blue background. On the right, a main panel titled 'Choose a system to connect to the required service' shows a dropdown menu set to 'OVP Test HANA Server' and a text input field containing '/sap/smartsbusiness/core/odata/modeler/SMART\_BUSINESS.xsodata'. Below these are two buttons: 'Test' and 'Show Details'. A large central area titled 'Service' displays a tree view under 'SMART\_BUSINESS.xsodata' with nodes: Chips, Catalogs, INDICATORS, INDICATOR\_TEXTS, EVALUATIONS, EVALUATION\_TEXTS, EVALUATION\_FILTERS, EVALUATION\_VALUES, TAGS, PROPERTIES, and ASSOCIATIONS.

## 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 'Annotation Selection' step in the SAP Web IDE. At the top, a title bar says 'Annotation Selection'. Below it, a note reads: '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.' A button '+ Add Annotation Files' is available. A table lists the selected annotation file: Rank 1, Name 'Smart\_Business', and Source 'Remote'.

| Rank | Name           | Source |
|------|----------------|--------|
| 1    | Smart_Business | Remote |

## 11.1.4 Modifying the Application Descriptor Configuration

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

### Context

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

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

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

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

### Procedure

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

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

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

2. Change the settings in each tab as required.

[Settings Tab Options \[page 79\]](#)

[Data Sources Tab Options \[page 81\]](#)

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

[Models Tab Options \[page 81\]](#)

Define the data models for your application.

[Routing Tab Options \[page 82\]](#)

Define routes and targets for the application.

[Navigation Tab Options \[page 84\]](#)

## 11.1.4.1 Settings Tab Options

### General Section

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

| Option                | Description  |
|-----------------------|--|
| Version               | Required. The version of application descriptor schema.  |
| ID                    | Required. A unique application identifier in the format, <code>sap.fiori.&lt;appName&gt;</code> .  |
| Type                  | Required. The type of application: <i>application</i> , <i>component</i> , or <i>library</i> .   |
| Title                 | Required. Enter the title of the application inside double curly brackets ( <code>{ { } }</code> ).  |
| Description           | Enter the description of the application inside double curly brackets ( <code>{ { } }</code> ).  |
| Source Template       | The ID and version of the template from which the application was generated.   |
| 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.<br>To add a keyword, click <code>+</code> , and in the popup, enter the new keyword inside double curly brackets ( <code>{ { } }</code> ). |
| Application Component | The application component hierarchy.   |

### User Interface Section

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

| Option     | Description   |
|------------|---|
| Technology | The UI technology. The default is <i>UI5</i> .  |
| Devices    | Select the device types on which the application can run - desktop, tablet, and phone.  |
| Themes     | The supported SAP themes that the application can use, such as <code>sap_hcb</code> , <code>sap_bluecrystal</code> , and so on. |

| Option            | Description  |
|-------------------|--|
| Application Icons | <p>The icons used in the application:</p> <ul style="list-style-type: none"> <li>• <a href="#">Main</a> - Select the icon that you want to use as the main application icon.</li> <li>• <a href="#">Phone 57 px</a> - Select the 57x57 pixel application icon for non-Retina iPhones.</li> <li>• <a href="#">Phone 114 px</a> - Select the 114x114 pixel application icon for Retina iPhones.</li> <li>• <a href="#">Tablet 72 px</a> - Select the 72x72 pixel application icon for non-Retina iPads.</li> <li>• <a href="#">Tablet 144 px</a> - Select the 144x144 pixel application icon for Retina iPads.</li> <li>• <a href="#">Favorites</a> - Enter the ICO file to be used in the browser and for desktop shortcuts.</li> </ul> |

## 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 <b>+</b>, 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 <b>+</b>, and in the popup, choose the dependency type, <a href="#">Library</a> or <a href="#">Component</a>. 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"> <li>• <a href="#">Cozy</a>: 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.</li> <li>• <a href="#">Compact</a>: 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.</li> </ul> |

## 11.1.4.2 Data Sources Tab Options

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

### Services

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

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

| Property      | Description  |
|---------------|--|
| Name          | Required: Unique name for the data source.   |
| 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 <a href="#">2.0</a> (default) or version <a href="#">4.0</a> .   |

### 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.

## 11.1.4.3 Models Tab Options

Define the data models for your application.

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

## OData Models

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

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

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

## Other Models

Define models based on JSON, XML or other resource files, such 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 \[page 81\]](#)

### 11.1.4.4 Routing Tab Options

Define routes and targets for the application.

## Default Configuration

Define the default route and target properties.

| Option              | Description  |
|---------------------|--|
| View Path           | Prefix that precedes the view name. For example, if the view name is <code>myView</code> and the <code>View Path</code> is <code>myApp</code> , the created view is <code>myApp.myView</code> .                                      |
| View Type           | Type of view that is created: <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 <code>View Level</code> is used to define the transition direction when navigating to this view.   |
| Control Aggregation | Name of an aggregation of the <code>Control ID</code> (target control) that contains views. For example, a <code>NavContainer</code> has an aggregation called <code>Pages</code> and the shell container has <code>Content</code> . |
| Transition          | Type of transition when navigating from the previous view to this view: <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 <code>Control ID</code> parameter.   |
| Parent              | Name of another target that will be displayed once this target is displayed.   |
| Clear Aggregation   | Whether the aggregation should be cleared before adding the view.  |

## Routes

Add and configure routes for the application.

1. Click `+`.
2. Change the name, if required.
3. Define the route properties.
4. Use the up and down arrows to change the order of routes in the list.

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

## Manage Targets

Add and configure targets.

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

## 11.1.4.5 Navigation Tab Options

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

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

### i Note

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

## Inbound Table

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

1. To add an intent, click **+**.
2. In the new row:
  - Enter the semantic object as defined in an app launcher tile, for example, `SalesOrder`.
  - Enter the action to be performed on the object. The action can be a verb or a short phrase starting with lower case and without blanks. Examples are `display`, `create`, and `release`.
3. For each intent, select the row in the table, and then define its tile and inbound parameters.

## Inbound Tile

Define the tile to be used for navigation for the selected intent.

| Option   | Description   |
|----------|---|
| Title    | The title that you want to appear on the tile. The title must be in double curly brackets ( <code>{ { } }</code> ). |
| Subtitle | The subtitle that you want to appear on the tile.   |
| Icon     | Select the icon to appear on the tile.  |

| Option           | Description   |
|------------------|---|
| Data Source      | The data source defined in the <code>sap.app</code> section that returns the information for an SAP Fiori launchpad dynamic tile. |
| Path             | The relative path of the OData service that returns the information for the dynamic tile (relative to the data source path).      |
| Refresh Interval | The time interval between data refreshes in the dynamic tile.   |

## Inbound Parameters

Click **+** to add parameters, also known as signatures, for the selected intent.

For each parameter set the following options.

| Option        | Description  |
|---------------|--|
| Name          | A unique name for the parameter.   |
| Value         | A string value for the parameter. This value can be a verbatim default value or a reference to another parameter, according to the value format. For optional parameters, this is the default.   |
| Value Format  | The format of the parameter value: <ul style="list-style-type: none"> <li><i>plain</i> - the parameter value is taken as a literal string value.</li> <li><i>reference</i> - the parameter value is a reference to another parameter.</li> </ul>   |
| Required      | Whether this is a mandatory or optional parameter.   |
| Filter Value  | A string value for the filter. This value can be a verbatim filter value, a regular expression, or a reference, according to the filter format.  |
| Filter Format | The format of the filter value: <ul style="list-style-type: none"> <li><i>plain</i> - the filter value must match the actual parameter value.</li> <li><i>regexp</i> - the filter value represents a regexp, which must be present in the parameter value.</li> <li><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.</li> </ul> |

## 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.

## 11.1.5 Developing Applications

Once you have created a project, you can use interactive features to write code from scratch, add a new component to the application, or add an extension to it. Run the application at any time to see how it is evolving.

### [Working in the Code Editor \[page 86\]](#)

Use keyboard shortcuts and context menus to easily edit and navigate through your code and code comments.

### [Using Code Completion \[page 93\]](#)

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 \[page 112\]](#)

SAP Web IDE performs code checking, also known as validation, and displays errors as annotations.

### [Developing Apps Using SAP Fiori Elements \[page 198\]](#)

This section contains information about creating SAP Fiori apps using SAP Fiori elements in SAP Web IDE.

### [Using the Outline Pane for JavaScript Files \[page 203\]](#)

The Outline pane helps you to understand the structure of JavaScript files and to navigate through the code.

### [Creating an HTML5 Application Descriptor File \[page 205\]](#)

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.

## 11.1.5.1 Working in the Code Editor

Use keyboard shortcuts and context menus to easily edit and navigate through your code and code comments.

Feature support depends on file types.

All edit actions are also available from the *Edit* menu.

### **i Note**

ES6 is not supported in JavaScript files.

### **i Note**

The following keyboard shortcuts are for Microsoft Windows. For Mac OS, see [Keyboard Shortcuts \[page 38\]](#).

## 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                                  |  | <i>Close Tabs to the Right</i>           |
| Undo edit  | [Ctrl] + [Z]   | <i>Undo</i>                              |
| Redo edit  | [Ctrl] + [Y]   | <i>Redo</i>                              |
| 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]   | <i>Go to Line</i>                        |
| Copy a line to the line above or below                                       | [Alt] + [Shift] + [Up arrow]<br>[Alt] + [Shift] + [Down arrow] | -  |
| 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                |
|---|---|-----------------------------|
| <p>For JavaScript and XML files</p> <p>Comment out a line and restore to code.</p> <p>Comment syntax is appended to the code fragment automatically as a line. Each comment line is prefixed with //.</p> <p>For example:</p> <pre>//Use the comment style for short comments.</pre>  | <code>Ctrl</code> + <code>/</code>                      | <i>Toggle Line Comment</i>  |
| <p>Comment out a block and restore to code.</p> <p>Comment syntax is appended to the code fragment as a block using /* and */ to wrap the comment block. For example:</p> <pre>/* This is a comment block. Use this block comment style when comments span multiple lines.*/</pre> <pre>//Use the comment style for short comments.</pre> | <code>Ctrl</code> + <code>Shift</code> + <code>/</code> | <i>Toggle Block Comment</i> |
| <p>Flag JavaScript code with a TODO comment.</p> <p>A // TODO comment is added at the cursor location. If the line already contains a // TODO comment, the action is ignored.</p>   | <code>Ctrl</code> + <code>Alt</code> + <code>T</code>   | <i>Add TODO Comment</i>     |

## Beautify File Formatting

Beautify file formatting for JavaScript, JSON, XML, and CSS files using the context menu *Beautify* option or `Ctrl` + `Alt` + `B`.

- The XML beautifier formats code with line wrapping at 140 characters, and an indentation of one tab space. The beautifier is not configurable.
- The CSS beautifier formats indentation with one tab space and is not configurable.
- You can customize JavaScript beautifier settings for your project. For more information, see [Customize JavaScript Beautifier Properties \[page 68\]](#).

## 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.

### 11.1.5.1.1 Configure the Code Editor

Define the appearance and behavior of the code editor, and whether to autosave all changes in SAP Web IDE.

#### Procedure

1. In the left sidebar, choose  (Preferences), then choose *Code Editor*.
2. Customize the appearance, input behavior, or save options.

##### Appearance

| Option                           | Description   |
|----------------------------------|---|
| <b>Code Editor Theme</b>         | The theme of the code editor determines the background color and color of text.   |
| <b>Font</b>                      | The monospace font for text in the code editor  |
| <b>Font Size</b>                 | The font size of text in the code editor.   |
| <b>Code Folding</b>              | 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.<br>Click in the gutter to mark begin and end points. |
| <b>Full line selection</b>       | Line selection extends to the end of the line instead of to the end of the text in the line.  |
| <b>Highlight selected word</b>   | Highlights all occurrences of the word at the cursor position.  |
| <b>Show invisible characters</b> | Shows white-space characters such as spaces, tabs, and new lines.   |

##### Input Behavior

| Option                                | Description   |
|---------------------------------------|---|
| <b>Auto-pair characters</b>           | Auto-pairs characters, such as quotation marks, parentheses, brackets, and so on.   |
| <b>Use spaces for tab indentation</b> | Uses spaces for tab indentation, also known as soft tabs.   |
| <b>New Line Mode</b>                  | Lets you choose one of the following: <ul style="list-style-type: none"><li>○ auto</li><li>○ windows</li><li>○ unix</li></ul> |

##### Save Options

| Option  | Description  |
|---|--|
| <b>Automatically save changes in all open documents at pre-set intervals.</b> | For more information, see <a href="#">Setting User Preferences [page 34]</a> .   |
| <b>Automatically beautify the code of an active document on manual save.</b>  | For more information about beautifying JavaScript code, see <a href="#">Customize JavaScript Beautifier Properties [page 68]</a> . |

Test your settings in the preview area on the right side of the screen..

3. Change the *Autosave* setting, if required.
4. Click *Save*.

## 11.1.5.1.2 Generate JSDoc Comment Snippets

You can generate a snippet for JavaScript function declaration that creates a template for documenting the function.

### Context

The JSDoc comment snippet provides a template for documenting the function that you create. You can use code completion for variables within the snippet.

### Procedure

1. Select a function, and press `Ctrl` + `Alt` + `J`, or choose *Generate JSDoc Comment* from the context menu.  
The snippet is displayed above the function code.
2. Enter information about the function variables in the corresponding placeholders in the comment.

## 11.1.5.1.3 Locate Objects in Code

The code editor allows you to locate objects or definitions of objects in code.

### Context

Methods can be defined in other modules on which the source file depends. You can use *Goto Definition* to navigate to methods declared explicitly in the dependency file and generated methods for metadata properties, associations, aggregations, and events of SAPUI5 controls.

The dependencies can be defined using a dependency declaration in either of the following ways:

- Use `jQuery.sap.require(<moduleId1>, <moduleId2>, ...)`. The dependency (target) file should contain the module declaration `jQuery.sap.declare(<moduleId>)`.
- Use `sap.ui.define([<dependencyFile1>, <dependencyFile2>, ...], function(d1, d2, ...) { })`; where `dependencyFile` is the relative file path in the current project or the logical path to the library module. The dependency (target) files should contain a module declaration using `sap.ui.define` or Asynchronous Module Definition (AMD).

Suggestions can be provided only if the dependency file is visible in the current project.

You can:

- Use goto services to locate the definition action of a user-defined object (for example, a variable, function, object, or property) in JavaScript files for in a project. Goto searches the active file as well as all files in the same project.

#### i Note

Goto is not supported for JavaScript native keywords (for example, `var`, `window`, `JSON`, or SAPUI5 library objects (for example, `sap`, `sap.ui`, `sap.ui.core`), as they are not user-defined objects. If you attempt to use goto with a restricted object, a message appears in the console pane to indicate that the definition was not found.

- Highlight all instances of the selected object in the active file.

## Procedure

1. To locate the definition using goto services:
  - a. Validate whether the object is user-defined by placing the cursor over the user-defined object and pressing `CTRL` + `ALT`.  
If the object has a user-defined definition, it appears underlined in blue. Otherwise, the console reports that the definition cannot be found.
  - b. To go to the location, press `CTRL` + `ALT` and click the underlined object.  
If the definition is in the current file, the definition is highlighted. Otherwise, the correct file is first opened and the definition is highlighted.

#### 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>
```

## 11.1.5.2 Using Code Completion

The code completion feature assists you when you are writing your code by preventing typos and other common mistakes, and providing API reference information for SAPUI5 objects.

### Context

Code completion is triggered in one of the following ways:

- Auto hint code completion for JavaScript and XML files  
Based on your cursor location, SAP Web IDE establishes the context and displays a constrained list of suggestions. Use the corresponding icon to visually identify the code type being completed (that is, XML or JavaScript). This option is configurable; see [Enabling Code Completion \[page 95\]](#).
- Manual code completion  
Supported for various file types including: i18 and messagebundle property files.
  - i18 and messagebundle
  - JavaScript
  - XMLNo configuration is required.

### Cross-file completion for JavaScript

The dependencies can be defined using a dependency declaration in either of the following ways: SAP Web IDE can provide suggestions for functions (methods) that are defined in other modules on which the source file depends. Suggestions include methods declared explicitly in the dependency file and generated methods for metadata properties, associations, aggregations, and events of SAPUI5 controls.

- Use SAP Web IDE can provide suggestions for functions (methods) that are defined in `jQuery.sap.require(<moduleId1>, <moduleId2>, ...)`. The dependency (target) file should contain the module declaration `jQuery.sap.declare(<moduleId>)`.
- Use `sap.ui.define([<dependencyFile1>, <dependencyFile2>, ...], function(d1, d2, ...) { })`; where dependencyFile is the relative file path in the current project or the logical path to the library module. The dependency (target) files should contain a module declaration using `sap.ui.define` or Asynchronous Module Definition (AMD).

### 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.

### Example

This example of embedded type code completion shows how you can attach a type to your variable definition to enable presentation of appropriate code completion suggestions for the variable.

Define the variable type in a JSDoc comment before the variable definition, and then press `Ctrl` + `Space` for code assistance. When you hover over a suggestion (e.g., `setIconDensityAware`), API reference information is shown in a tooltip.

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

```

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(blIconDensityAware) : sap.m.Button
    M setIconFirst(blIconFirst) : sap.m.Button
    M setText(sText) : sap.m.Button
    M setTextDirection(sTextDirection) : sap.m.Button
}

```

A tooltip for the `setIconDensityAware(blIconDensityAware)` method is displayed on the right, containing the following information:

**setIconDensityAware(blIconDensityAware) : 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 an XML 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 on the right, containing the following information:

**List**

Page background color when a List is set as the Page content.

The tooltip lists the following options: List, Solid, Standard, Transparent.

## 11.1.5.2.1 Configure Code Completion

You can enable code completion as you type (auto hint) for JavaScript and XML files.

### Procedure

- To open the *Preferences* perspective, in the left sidebar, choose (Preferences).
- Click *Code Completion* and select the *Enable* checkbox.
- Click *Save*.

## 11.1.5.2.2 Code Completion for Property Files

You can insert code snippets into your `i18n` and `messagebundle` property files enabling you to easily insert UI elements (for example, button names, table titles) into your code. The elements are highlighted in different colors as you type.

This feature enables you to::

- Press `Ctrl` + `Space` to choose the text type from a dropdown list.
- Type in the comments, keys, and values - as you type each element is highlighted in a different color.
- Use the `Tab` key to quickly navigate between the highlighted elements.

## 11.1.5.2.3 Code Completion for OPA and QUnit Tests

Use code completion for snippets when developing OPA and QUnit tests in SAP Web IDE.

### OPA Tests

When you develop OPA tests in an OPA page, SAP Web IDE can provide snippets for OPA actions and OPA assertions. You can also use code completion to add the OPA tests that you create to an OPA journey.

At the relevant location in the file, enter `opa``Ctrl` + `Space` and choose the relevant snippet from the list of suggestions. The code is inserted and you can then add the required definitions in the placeholders.

For more information about OPA testing, see [One Page Acceptance Tests \(OPA5\)](#)

### QUnit Tests

When you develop QUnit tests, you can use code completion for QUnit modules and QUnit tests.

At the relevant location in the file, enter `qunit``Ctrl` + `Space` and choose the relevant snippet from the list of suggestions. The code is inserted and you can then add the required definitions in the placeholders.

For more information about QUnit testing, see [QUnit Testing Fundamentals](#)

## 11.1.5.2.4 Try It: JavaScript Code Completion Basics

A self-guided example that demonstrates how to use JavaScript code completion services for SAPUI5 applications.

### Prerequisites

Open a new JavaScript file in a new or existing project.

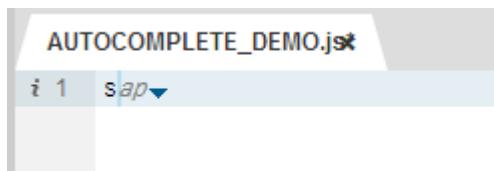
### Context

This example demonstrates the following code completion features:

- JavaScript code completion
- Element hints
- Snippet insertion

### Procedure

1. In the JavaScript editor, type `s`, and notice that the autocomplete suggestions automatically appear adjacent to the letter in context.



2. When you see `sap`, press `Enter`.
3. Continue the string and type `.`, then press `Ctrl` + `Space` to list all the namespaces for `sap`.
4. In the popup, use the arrow keys to select `ui : sap.ui`, and press `Enter`.

A screenshot of the SAP Web IDE interface. The title bar says "CodeCompletion.js". In the code editor, the first line starts with "sap.". A dropdown menu shows several suggestions: "Logger : sap.Logger", "Logon : sap.Logon", "m : sap.m", "me : sap.me", "Push : sap.Push", "Settings : sap.Settings", "ui : sap.ui" (which is highlighted in blue), "ushell : sap.ushell", and "viz : sap.viz".

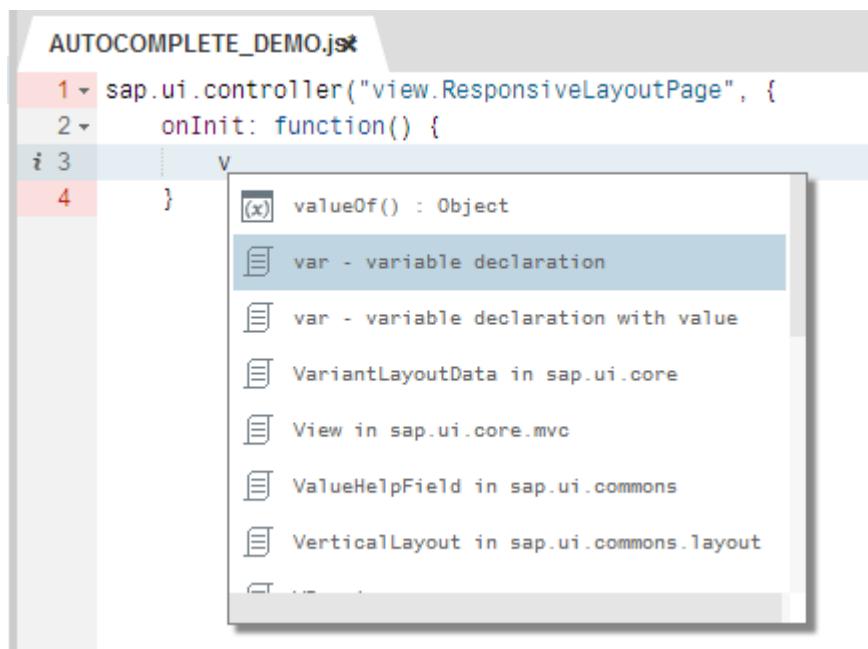
5. To append the controller function, type **cont**. The suggestion reduces to the correct one. Press **Enter** to append it in this location.

A screenshot of the SAP Web IDE interface. The title bar says "AUTOCOMPLETE\_DEMO.js". In the code editor, the first line is "sap.ui.controller(sName, [oControllerImpl])".

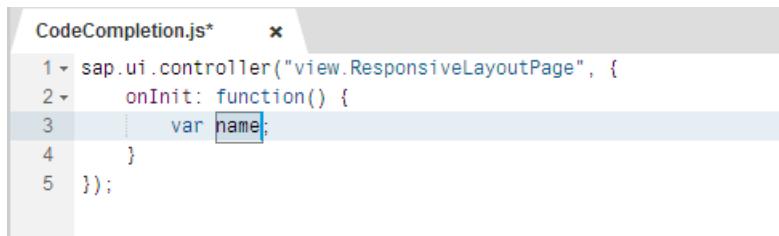
6. Add the `onInit` function in row 2, using the techniques demonstrated in previous steps.

A screenshot of the SAP Web IDE interface. The title bar says "AUTOCOMPLETE\_DEMO.js". The code editor contains:  
1 sap.ui.controller("view.ResponsiveLayoutPage", {  
2 onInit: function() {  
3 ...  
4 }  
5 }  
6 }  
7 }  
8 }  
9 }  
10 }  
11 }  
12 }  
13 }  
14 }  
15 }  
16 }  
17 }  
18 }  
19 }  
20 }  
21 }  
22 }  
23 }  
24 }  
25 }  
26 }  
27 }  
28 }  
29 }  
30 }  
31 }  
32 }  
33 }  
34 }  
35 }  
36 }  
37 }  
38 }  
39 }  
40 }  
41 }  
42 }  
43 }  
44 }  
45 }  
46 }  
47 }  
48 }  
49 }  
50 }  
51 }  
52 }  
53 }  
54 }  
55 }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 }  
63 }  
64 }  
65 }  
66 }  
67 }  
68 }  
69 }  
70 }  
71 }  
72 }  
73 }  
74 }  
75 }  
76 }  
77 }  
78 }  
79 }  
80 }  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }  
101 }  
102 }  
103 }  
104 }  
105 }  
106 }  
107 }  
108 }  
109 }  
110 }  
111 }  
112 }  
113 }  
114 }  
115 }  
116 }  
117 }  
118 }  
119 }  
120 }  
121 }  
122 }  
123 }  
124 }  
125 }  
126 }  
127 }  
128 }  
129 }  
130 }  
131 }  
132 }  
133 }  
134 }  
135 }  
136 }  
137 }  
138 }  
139 }  
140 }  
141 }  
142 }  
143 }  
144 }  
145 }  
146 }  
147 }  
148 }  
149 }  
150 }  
151 }  
152 }  
153 }  
154 }  
155 }  
156 }  
157 }  
158 }  
159 }  
160 }  
161 }  
162 }  
163 }  
164 }  
165 }  
166 }  
167 }  
168 }  
169 }  
170 }  
171 }  
172 }  
173 }  
174 }  
175 }  
176 }  
177 }  
178 }  
179 }  
180 }  
181 }  
182 }  
183 }  
184 }  
185 }  
186 }  
187 }  
188 }  
189 }  
190 }  
191 }  
192 }  
193 }  
194 }  
195 }  
196 }  
197 }  
198 }  
199 }  
200 }  
201 }  
202 }  
203 }  
204 }  
205 }  
206 }  
207 }  
208 }  
209 }  
210 }  
211 }  
212 }  
213 }  
214 }  
215 }  
216 }  
217 }  
218 }  
219 }  
220 }  
221 }  
222 }  
223 }  
224 }  
225 }  
226 }  
227 }  
228 }  
229 }  
230 }  
231 }  
232 }  
233 }  
234 }  
235 }  
236 }  
237 }  
238 }  
239 }  
240 }  
241 }  
242 }  
243 }  
244 }  
245 }  
246 }  
247 }  
248 }  
249 }  
250 }  
251 }  
252 }  
253 }  
254 }  
255 }  
256 }  
257 }  
258 }  
259 }  
260 }  
261 }  
262 }  
263 }  
264 }  
265 }  
266 }  
267 }  
268 }  
269 }  
270 }  
271 }  
272 }  
273 }  
274 }  
275 }  
276 }  
277 }  
278 }  
279 }  
280 }  
281 }  
282 }  
283 }  
284 }  
285 }  
286 }  
287 }  
288 }  
289 }  
290 }  
291 }  
292 }  
293 }  
294 }  
295 }  
296 }  
297 }  
298 }  
299 }  
300 }  
301 }  
302 }  
303 }  
304 }  
305 }  
306 }  
307 }  
308 }  
309 }  
310 }  
311 }  
312 }  
313 }  
314 }  
315 }  
316 }  
317 }  
318 }  
319 }  
320 }  
321 }  
322 }  
323 }  
324 }  
325 }  
326 }  
327 }  
328 }  
329 }  
330 }  
331 }  
332 }  
333 }  
334 }  
335 }  
336 }  
337 }  
338 }  
339 }  
340 }  
341 }  
342 }  
343 }  
344 }  
345 }  
346 }  
347 }  
348 }  
349 }  
350 }  
351 }  
352 }  
353 }  
354 }  
355 }  
356 }  
357 }  
358 }  
359 }  
360 }  
361 }  
362 }  
363 }  
364 }  
365 }  
366 }  
367 }  
368 }  
369 }  
370 }  
371 }  
372 }  
373 }  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }

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.



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.



9. Name the var as **fc1 = form**, then press **Ctrl** + **Space** to retrieve a list of form templates for it.

A screenshot of the SAP Web IDE interface. The main editor window shows a file named "AUTOCOMPLETE\_DEMO.js\*". The code contains a snippet of JavaScript using the sap.ui.controller function. A cursor is positioned at the end of the line "var fc1 = form". A dropdown code completion menu is open, listing several options under the category "FormContainer in sap.ui.layout.form". The option "FormContainer in sap.ui.layout.form" is highlighted with a blue selection bar.

```
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2     ...onInit: function() {
3         ...var fc1 = form
4     }
}
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
```

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 main editor window shows a file named "CodeCompletion.js\*". The code defines a controller with an `onInit` function. Inside it, a variable `fc1` is assigned a new instance of `sap.ui.layout.form.FormContainer`. The cursor is placed on the `id` property of this object. A code completion dropdown is open, showing various properties of the `FormContainer` class, such as `expanded`, `expandable`, `visible`, `tooltip`, `customData`, `formElements`, and `title`.

```
1 sap.ui.controller("view.ResponsiveLayoutPage", {
2     ...onInit: function() {
3         ...var fc1 = new sap.ui.layout.form.FormContainer({
4             id: "id", // sap.ui.core.ID
5             expanded: true, // boolean
6             expandable: false, // boolean
7             visible: true, // boolean
8             tooltip: undefined, // sap.ui.core.TooltipBase
9             customData: [], // sap.ui.core.CustomData
10            formElements: [], // sap.ui.layout.form.FormElement
11            title: undefined // sap.ui.core.Title
12        })
13    ;
14  }
15});
```

## 11.1.5.2.5 Try It: JavaScript Code Completion with a User-Defined Object

A self-guided example that demonstrates how to use code completion with an object that you define. Create one file to define the object. Create a second file in which to use code completion with the object that you defined in the first file.

### Context

SAP Web IDE code completion automatically displays user-defined objects in the same project.

### Procedure

1. In a new or existing project, create a new JavaScript file.
2. Add `jQuery.sap.declare(FileName)` to the start of all source files; otherwise, objects will not appear as a hint.
3. Create an object called `obj1`, by replicating these lines in this empty file and saving the changes:

```
cross-file1.js
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:
  - o Either type `var n = obj1`, then press `Ctrl` + `Space` to get this hint from the first file:

A screenshot of the SAP Web IDE interface. Two tabs are open: 'cross-file1.js' and 'cross-file2.js'. The code in 'cross-file1.js' is as follows:

```
1 jQuery.sap.require("cross-file1");
2
3 var n = obj1.
```

The cursor is at the end of 'obj1.', and a tooltip-like code completion dropdown is displayed. It shows the following suggestions:

- (x) a : Number
- (x) b : String
- 
- (x) hasOwnProperty(property) : Boolean
- (x) isPrototypeOf(object) : Boolean
- (x) propertyIsEnumerable(property) : Boolean
- (x) toLocaleString() : String
- (x) toString() : String
- (x) valueOf() : Object
- (x) prototype : Object

- Or type `var f = fun`, then press `Ctrl` + `Space` to get this hint from the first file:

A screenshot of the SAP Web IDE interface. Two tabs are open: 'cross-file1.js' and '\*cross-file2.js'. The code in 'cross-file1.js' is as follows:

```
1 jQuery.sap.require("cross-file1");
2
3 var f = fun.
```

The cursor is at the end of 'fun.', and a tooltip-like code completion dropdown is displayed. It shows the following suggestions:

- (x) fun2(x) : Number
- (x) Function() : Function
- function - function declaration
- function

7. Select the first result and press `Enter`.

## Results

The code updates accordingly in the second file.

### 11.1.5.2.6 Try It: XML Code Completion

A self-guided example that demonstrates how to use XML code completion services.

#### Prerequisites

Open a new XML file in a new or existing project.

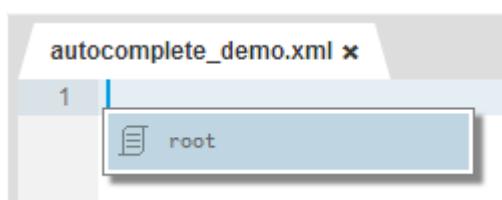
## Context

This example demonstrates the following code completion features:

- XML node completion
- Element hints

## Procedure

1. Press **Ctrl** + **Space** and select the `root` node suggestion.



The `root` node is automatically populated with required elements and values.

```
autocomplete_demo.xml*  
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 interface showing the completed XML code. 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>  
The code is color-coded, and the entire code block is highlighted with a light gray background, indicating it is selected or part of a larger block.

2. In row 8, type **<p**.

A screenshot of the SAP Web IDE interface. The title bar says "autocomplete\_demo.xml\*x". The code editor shows the following XML snippet:

```
1 <core:View
2   xmlns:core="sap.ui.core"
3   xmlns="sap.m"
4   xmlns:commons="sap.ui.commons"
5   xmlns:layout="sap.ui.layout"
6   controllerName=""
7   viewName="">
8 <panel>
9 </core:View>
```

The cursor is at the end of the word "panel" in line 8. A dropdown menu is open, showing suggestions: "<> />" (selected), "<> >", "backgroundDesign", "busy", "busyIndicatorDelay", "enableScrolling", "formatError", and "icon".

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.

A screenshot of the SAP Web IDE interface. The title bar says "autocomplete\_demo.xml\*x". The code editor shows the following XML snippet:

```
1 <core:View
2   xmlns:core="sap.ui.core"
3   xmlns="sap.m"
4   xmlns:commons="sap.ui.commons"
5   xmlns:layout="sap.ui.layout"
6   controllerName=""
7   viewName="">
8 <Page |>
9 </core:View>
10
```

The cursor is at the end of the word "Page" in line 8. A dropdown menu is open, showing suggestions: "<> />" (selected), "<> >", "backgroundDesign", "busy", "busyIndicatorDelay", "enableScrolling", "formatError", and "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
```

### 11.1.5.2.7 Try It: XML Code Snippet Insertion

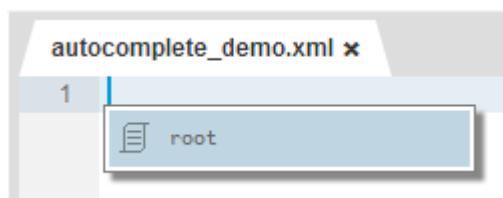
A self-guided example that demonstrates how to insert an XML code snippet.

#### Prerequisites

Open a new XML file in a new or existing project.

#### Procedure

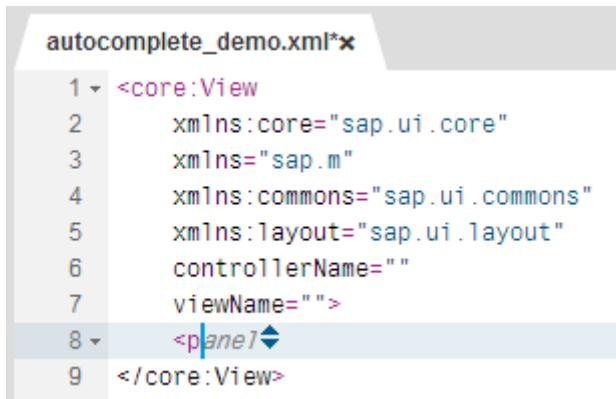
1. Press **Ctrl** + **Space** and select the `root` node suggestion.



The `root` node is automatically populated with required elements and values.

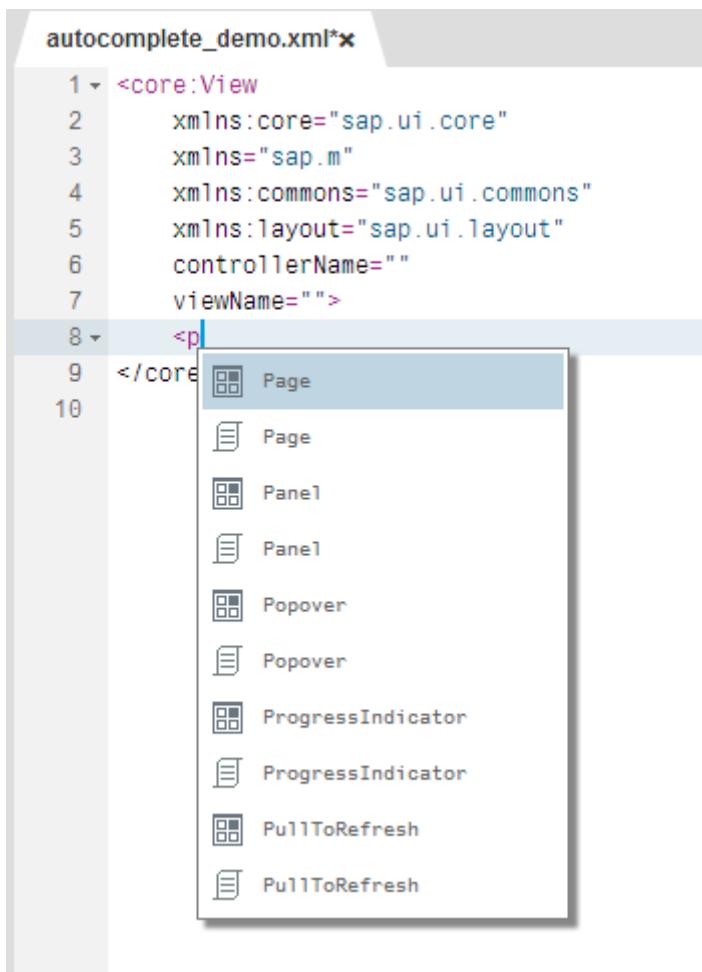
```
1 <core:View
2   xmlns:core="sap.ui.core"
3   xmlns="sap.m"
4   xmlns:commons="sap.ui.commons"
5   xmlns:layout="sap.ui.layout"
6   controllerName=""
7   viewName="">
8
9 </core:View>
10
```

2. In the root node, type `<p`.



```
autocomplete_demo.xml*x
1 <core:View
2   xmlns:core="sap.ui.core"
3   xmlns="sap.m"
4   xmlns:commons="sap.ui.commons"
5   xmlns:layout="sap.ui.layout"
6   controllerName=""
7   viewName="">
8 <p|ane1◆
9 </core:View>
```

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>
```

### 11.1.5.2.8 Try It: XML Metadata Completion from Schema Files

Display autocomplete metadata information from schema files. Metadata suggestions are context-aware. You can get suggestions for elements, attributes, attribute type, properties, and property values.

#### Prerequisites

Metadata and schema files must exist. For the purposes of this guided example, create two schema files:

**Schema demo2a.xsd** Includes these metadata definitions:

```
<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <xs:any minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute/>
  </xs:complexType>
</xs:element>
<xs:element name="employees">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="employee" maxOccurs="unbounded">
        </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Schema demo2b.xsd Includes these definitions:

```

<xs:attributeGroup name="customattrgroup">
    <xs:attribute name="inheritable" type="xs:boolean"/>
    <xs:attribute name="birthday" type="xs:date"/>
</xs:attributeGroup>
<xs:group name="infogroup">
    <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="phone" type="xs:string"/>
    </xs:sequence>
</xs:group>
<xs:element name="customer">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="customername" type="xs:string"
maxOccurs="unbounded"/>
            <xs:group ref="infogroup"/>
        </xs:sequence>
        <xs:attribute ref="gender"/>
        <xs:attribute name="mode" default="interleave" use="optional">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="none"/>
                    <xs:enumeration value="interleave"/>
                    <xs:enumeration value="suffix"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attributeGroup ref="customattrgroup"/>
    </xs:complexType>
</xs:element>

```

## Procedure

1. In a new or existing project, create a new XML file, and specify `demo2a.xsd` and `demo2b.xsd` as the metadata schema instances.

For example, if the XSD file uses a namespace, define the same namespace in the XML file as shown; otherwise, autocompletion for metadata fails.

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:SchemaLocation="http://<domain> demo2a.xsd
http://<domain> demo2b.xsd"

```



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.



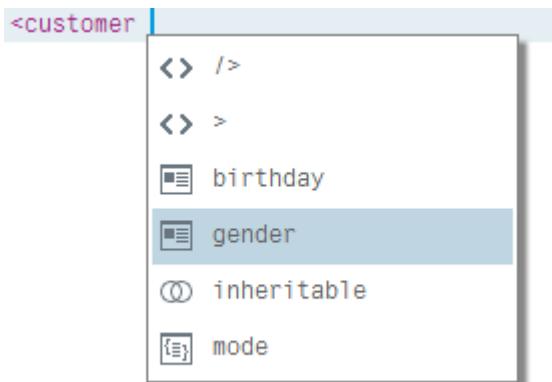
Because `<xs:anyAttribute/>` is defined for the `employee` element, the `gender` attribute from `demo2b.xsd` appears in this list.

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.

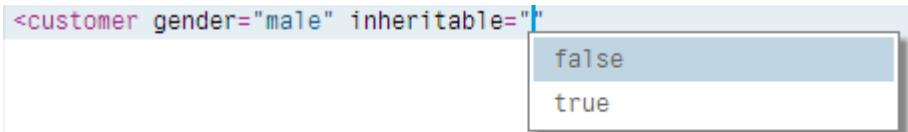


### i Note

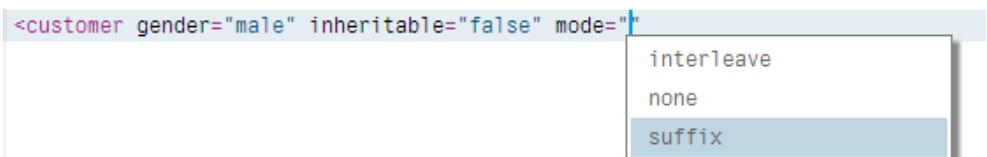
Note the icons in this example: the `mode` attribute's icon shows an enum type, and the `inheritable` attribute's icon shows a boolean type.

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



## 11.1.5.2.9 Supported SAPUI5 Libraries

Code completion is supported for many SAPUI5 libraries.

Code completion is supported for the following SAPUI5 libraries:

- `sap.ca.ui`
- `sap.m`
- `sap.me`
- `sap.ui.commons`
- `sap.ui.comp`
- `sap.ui.core`

- sap.ui.layout
- sap.ui.richtexteditor
- sap.ui.table
- sap.ui.unified
- sap.ui.ux3
- sap.ushell
- sap.uxapp
- sap.viz

### 11.1.5.3 Checking Code

SAP Web IDE performs code checking, also known as validation, and displays errors as annotations.

By default, when you open a JavaScript, JSON, or XML file, code checking is triggered and the detected code issues are displayed as annotations within the editor. Understand how to configure and use code checking in your project.

#### [Configure Code Checking \[page 113\]](#)

Configure when to trigger code checking, also known as code validation, and the level of messages to display.

#### [Code Checking Annotations \[page 114\]](#)

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 \[page 116\]](#)

Review the default JavaScript validator configuration. Customizations always override these defaults.

#### [XML Validation \[page 170\]](#)

You can configure which XML validator to use in your project.

#### [Validation of neo-app.json Files \[page 195\]](#)

A project's neo-app.json file is validated on opening in the code editor.

#### [Using the Problems View \[page 196\]](#)

View information about problems in the projects in your workspace.

#### [Validation of manifest.json Files \[page 197\]](#)

Application descriptor files (manifest.json) for SAP Fiori projects are validated on opening in the code editor.

## 11.1.5.3.1 Configure Code Checking

Configure when to trigger code checking, also known as code validation, and the level of messages to display.

### Context

By default, code checking is enabled when you make changes to your code, and all messages are displayed. You can change these defaults.

The code checking level is also applied to the Problems view.

#### Note

You can customize code checking rules for each project. For more information about customizing and using code checking, see [Checking Code \[page 112\]](#).

### 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 \[page 112\]](#)

### Related Information

- [Code Checking Annotations \[page 114\]](#)
- [JavaScript Validation \[page 116\]](#)
- [XML Validation \[page 170\]](#)
- [Validation of neo-app.json Files \[page 195\]](#)

[Using the Problems View \[page 196\]](#)  
[Validation of manifest.json Files \[page 197\]](#)

## 11.1.5.3.2 Code Checking Annotations

If a syntax error is found during editing, the relevant lines are annotated with flags indicating the error severity. Understand the severity level of these flags so that when you open a file, you know how to interpret these annotations.

- All syntax errors are annotated on code line and tab levels and each annotation is colored according to its severity.

| Color  | Severity    |
|--------|-------------|
| Red    | Error       |
| Yellow | Warning     |
| Blue   | Information |

- When you hover over an annotation, a tooltip displays one or more detected issues and possible resolutions for the annotated line.

The detected issues of an annotated line are categorized (for easy identification), identified, and described, so that you can determine how best to resolve an issue.

**Categories** Used to classify the issue. For example, possible error, best practice, stylistic issue, and others.

**Rule IDs** Used to define the logic for detection or list known issue exceptions. For example, *semi* is a rule ID for ESLint.

**Messages** Detail the issue or suggest a possible resolution.

You can resolve the issue and continue development with iterative fixes.

**Parent topic:** [Checking Code \[page 112\]](#)

## Related Information

[Configure Code Checking \[page 113\]](#)  
[JavaScript Validation \[page 116\]](#)  
[XML Validation \[page 170\]](#)  
[Validation of neo-app.json Files \[page 195\]](#)  
[Using the Problems View \[page 196\]](#)  
[Validation of manifest.json Files \[page 197\]](#)

## 11.1.5.3.2.1 Try It: Code Correction Using the Default JavaScript Validator

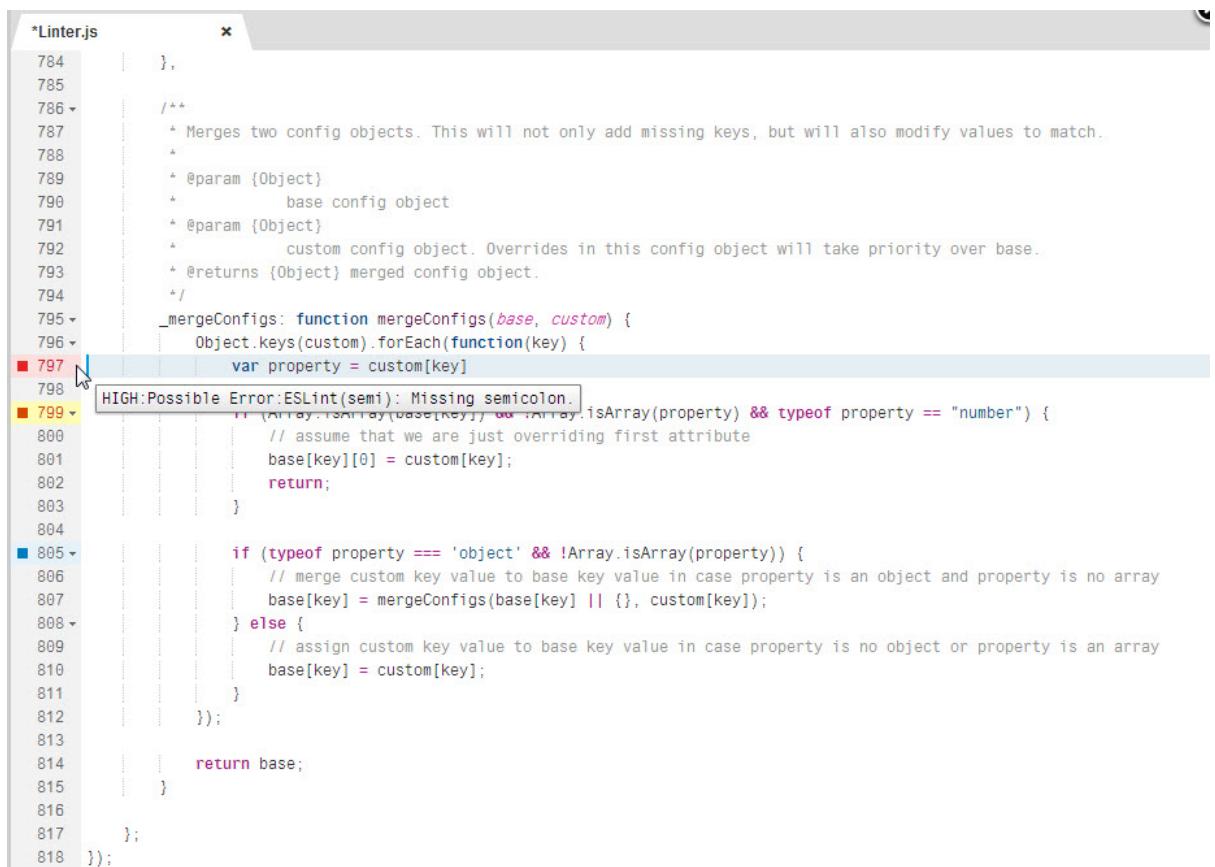
A self-guided example that demonstrates how to use the default JavaScript validator on an open JavaScript file.

### Prerequisites

Configure validation to run on change as described in [Configure Code Checking \[page 113\]](#). Then follow the steps in [Opening and Reading Files with Annotations \[page 114\]](#) to open a file named `Linter.js`.

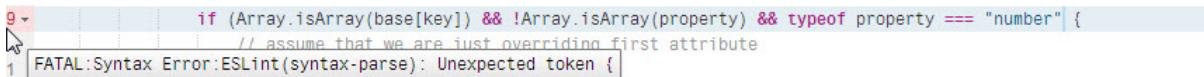
### Procedure

- Once the `Linter.js` file is open, use the colored square flags to determine which potential issues are the most serious.



The screenshot shows the SAP Web IDE interface with the file `*Linter.js` open. The code is a merge function for configuration objects. A tooltip is displayed over line 799, indicating a possible ESLint error: `HIGH:Possible Error:ESLint(semi): Missing semicolon.` The tooltip details that if an array's first element is not an array and its property type is not a number, it assumes the first attribute is being overridden. The code uses the `base[key][0] = custom[key];` assignment. The tooltip also notes that if the property type is an object or not an array, it merges the custom key value into the base key value. The tooltip is positioned over the line of code where `base[key] = custom[key];` is present. The IDE interface includes line numbers (784-818) and a status bar at the bottom.

- 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.

### 11.1.5.3.3 JavaScript Validation

Review the default JavaScript validator configuration. Customizations always override these defaults.

The SAP Web IDE default JavaScript validator uses ESLint code checking.

|                           |   |
|---------------------------|---|
| <b>ESLint Rule</b>        | See these resources for additional support:   |
| <b>Execution Defaults</b> | <ul style="list-style-type: none"><li>• For ESLint configuration details, see <a href="#">Configuring ESLint</a>.</li><li>• For ESLint rule information, see <a href="#">Rules</a>.</li></ul>   |
| <b>ESLint Rule</b>        | Attributes can have multiple supported values. Use the following:   |
| <b>Metadata Defaults</b>  | <ul style="list-style-type: none"><li>• <code>severity</code> attribute to define whether an issue renders as <code>error</code>, <code>warning</code> (default), or <code>information</code>.</li><li>• <code>category</code> attribute for a better semantic classification: <code>possible_error</code>, <code>best_practice</code>, <code>stylistic_issue</code>, and others.</li><li>• <code>help</code> attribute (optional) to override the default help links listed.</li></ul> |

**Parent topic:** [Checking Code \[page 112\]](#)

## Related Information

[Configure Code Checking \[page 113\]](#)

[Code Checking Annotations \[page 114\]](#)

- 
- [XML Validation \[page 170\]](#)
  - [Validation of neo-app.json Files \[page 195\]](#)
  - [Using the Problems View \[page 196\]](#)
  - [Validation of manifest.json Files \[page 197\]](#)

### 11.1.5.3.3.1 Create JavaScript Code Checking Rules

You can replace the SAP Web IDE default rule configurations in the JavaScript validator by configuring your own code checking rule configurations.

#### Context

The SAP Web IDE default JavaScript validator uses ESLint code checking. You can override the default rule configuration by creating a folder containing your custom rules.

#### Procedure

1. Choose .
2. Name the new folder and add your customized code checking rules to this folder.

**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.

## 11.1.5.3.3.2 Customize JavaScript Validator Configuration

You can customize the configuration of SAP Web IDE JavaScript validators.

### Context

You can customize the configuration and rules of SAP Web IDE JavaScript validators directly in the SAP Web IDE UI.

If you change the default settings and turn on or off specific rules, this information is stored in a `.eslintrc` file in the root folder of the project. If you have already configured ESLint rules in another development environment, you can import your `.eslintrc` file into the root folder, instead of using the UI. Additional SAP Web IDE-specific information about each rule – such as category, severity and help URL – is stored in an automatically generated `.eslintrc.ext` file, also in the root folder.

### Procedure

1. From the context menu of any file in your project, choose *Project Settings*.
2. From the *Project Settings* options, choose  *Code Checking*  *JavaScript* .
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.

#### Note

You can restore the default validator configuration and rules by clicking the *Reset* button next to the *Validator* field.

## 11.1.5.3.3.3 Fiori JavaScript Validator Rules

The Fiori JavaScript validator rules are used when building Fiori projects.

### [sap-cross-application-navigation \[page 120\]](#)

No static cross-application navigation targets are allowed.

---

## [sap-forbidden-window-property \[page 122\]](#)

Detects the usage of forbidden window properties.

## [sap-no-navigator \[page 123\]](#)

Detects `window.navigator` usage.

## [sap-no-override-rendering \[page 125\]](#)

Override of control methods is not allowed.

## [sap-no-override-storage-prototype \[page 126\]](#)

Override of `csap-no-override-storage-prototype` is not allowed.

## [sap-no-proprietary-browser-api \[page 127\]](#)

Discourage usage of proprietary browser APIs.

## [sap-no-sessionstorage \[page 129\]](#)

Usage of session storage is not allowed.

## [sap-no-ui5-prop-warning \[page 130\]](#)

Usage of private members of SAPUI5 objects is not allowed.

## [sap-no-ui5base-prop \[page 131\]](#)

Usage of private members of SAPUI5 objects is not allowed

## [sap-timeout-usage \[page 133\]](#)

Discourage usage of `setTimeout`.

## [sap-ui5-no-private-prop \[page 135\]](#)

Detects usage of private properties of SAPUI5 objects.

## [sap-usage-basemastercontroller \[page 137\]](#)

Detects usage of `BaseMasterController`.

## [sap-message-toast \[page 139\]](#)

Wrong usage of `sap.m.MessageToast` is not allowed.

## [sap-no-absolute-component-path \[page 140\]](#)

Absolute paths to component `includes` are not allowed.

## [sap-no-br-on-return \[page 142\]](#)

Detects the usage of `document.queryCommandSupported`.

## [sap-no-dom-access \[page 143\]](#)

Usage of certain methods of `document` is discouraged.

## [sap-no-dom-insertion \[page 145\]](#)

Usage of DOM insertion methods is not allowed.

## [sap-no-dynamic-style-insertion \[page 146\]](#)

Detects dynamic style insertion.

## [sap-no-element-creation \[page 148\]](#)

Direct DOM insertion is not allowed.

## [sap-no-encode-file-service \[page 149\]](#)

Detects the usage of `encode_file` service

## [sap-no-exec-command \[page 151\]](#)

Detects direct DOM manipulation.

## [sap-no-global-define \[page 152\]](#)

Detects definition of globals via window object.

[sap-no-global-event \[page 153\]](#)

Detects global event handling overrides.

[sap-no-global-selection \[page 155\]](#)

Usage of global selection is discouraged.

[sap-no-global-variable \[page 156\]](#)

Global variables should not be used in SAP Fiori applications.

[sap-no-hardcoded-color \[page 157\]](#)

Usage of hard coded colors is not allowed.

[sap-no-hardcoded-url \[page 159\]](#)

Use of hardcoded URLs is not allowed.

[sap-no-history-manipulation \[page 160\]](#)

Direct history manipulation is discouraged.

[sap-no-jquery-device-api \[page 162\]](#)

Usage of the jQuery device APIs is not allowed.

[sap-no-localhost \[page 163\]](#)

Usage of localhost is not allowed.

[sap-no-localstorage \[page 165\]](#)

Usage of local storage is not allowed.

[sap-no-location-reload \[page 166\]](#)

Detects location reload.

[sap-no-location-usage \[page 168\]](#)

Override of location properties and methods is not allowed.

### 11.1.5.3.3.1 sap-cross-application-navigation

No static cross-application navigation targets are allowed.

SAP Fiori-as-a-Service Enablement guideline prohibits the use of a static list of cross-application navigation targets.

#### Rule Details

This check prevents the usage of static cross-application navigation targets.

Instead, use the `IntentSupported` function of the `CrossApplicationNavigation` service. See the corresponding Cross Application Navigation and JSDOC API documentation. Note that the function is mass-enabled, so you can check an array of all relevant navigation targets in one call.

The following patterns are considered warnings:

```
sap.ushell.Container.getService("CrossApplicationNavigation").toExternal({});
```

The following patterns are not warnings:

```
checkPromoFactSheetAvailable : function() {
    // By default: promo factsheet not available
    this._bPromoFactSheetAvailable = false;
    if (this._oCrossAppNav) {
        // Check if the intent for the promotion factsheet is supported
        var sIntent = "#Promotion-displayFactSheet";
        var oDeferred = this._oCrossAppNav.isIntentSupported([sIntent]);
        oDeferred.done(jQuery.proxy(function(oIntentSupported) {
            if (oIntentSupported && oIntentSupported[sIntent] &&
                oIntentSupported[sIntent].supported === true) {
                // Remember that the navigation to the promotion factsheet is
                // possible
                this._bPromoFactSheetAvailable = true;
                // Activate the promotion links if they were already added to the
                // view
                this.activatePromotionLinks();
            }
        }, this));
    }
}
```

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

## Related Information

[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)

---

[sap-no-global-variable](#) [page 156]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

### 11.1.5.3.3.2 sap-forbidden-window-property

Detects the usage of forbidden window properties.

#### Warning Message

Usage of a forbidden window property.

#### Rule Details

The following patterns are considered warnings:

```
var top = window.top;
window.addEventListener(listener);
```

**Parent topic:** [Fiori JavaScript Validator Rules](#) [page 118]

#### Related Information

[sap-cross-application-navigation](#) [page 120]  
[sap-no-navigator](#) [page 123]  
[sap-no-override-rendering](#) [page 125]  
[sap-no-override-storage-prototype](#) [page 126]  
[sap-no-proprietary-browser-api](#) [page 127]  
[sap-no-sessionstorage](#) [page 129]  
[sap-no-ui5-prop-warning](#) [page 130]  
[sap-no-ui5base-prop](#) [page 131]  
[sap-timeout-usage](#) [page 133]

---

sap-ui5-no-private-prop [page 135]  
sap-usage-basemastercontroller [page 137]  
sap-message-toast [page 139]  
sap-no-absolute-component-path [page 140]  
sap-no-br-on-return [page 142]  
sap-no-dom-access [page 143]  
sap-no-dom-insertion [page 145]  
sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-define [page 152]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-hardcoded-url [page 159]  
sap-no-history-manipulation [page 160]  
sap-no-jquery-device-api [page 162]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.3 sap-no-navigator

Detects `window.navigator` usage.

#### Warning Message

`navigator` usage is forbidden, use `sap.ui.Device` API instead.

#### Rule Details

The `window.navigator` object should not be used at all, instead the `sap.ui.Device` API should be used.

The following patterns are considered warnings:

```
var language = navigator.language;
var name = navigator.appCodeName;
```

---

**Parent topic:** Fiori JavaScript Validator Rules [page 118]

## Related Information

[sap-cross-application-navigation](#) [page 120]  
[sap-forbidden-window-property](#) [page 122]  
[sap-no-override-rendering](#) [page 125]  
[sap-no-override-storage-prototype](#) [page 126]  
[sap-no-proprietary-browser-api](#) [page 127]  
[sap-no-sessionstorage](#) [page 129]  
[sap-no-ui5-prop-warning](#) [page 130]  
[sap-no-ui5base-prop](#) [page 131]  
[sap-timeout-usage](#) [page 133]  
[sap-ui5-no-private-prop](#) [page 135]  
[sap-usage-basemastercontroller](#) [page 137]  
[sap-message-toast](#) [page 139]  
[sap-no-absolute-component-path](#) [page 140]  
[sap-no-br-on-return](#) [page 142]  
[sap-no-dom-access](#) [page 143]  
[sap-no-dom-insertion](#) [page 145]  
[sap-no-dynamic-style-insertion](#) [page 146]  
[sap-no-element-creation](#) [page 148]  
[sap-no-encode-file-service](#) [page 149]  
[sap-no-exec-command](#) [page 151]  
[sap-no-global-define](#) [page 152]  
[sap-no-global-event](#) [page 153]  
[sap-no-global-selection](#) [page 155]  
[sap-no-global-variable](#) [page 156]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

## 11.1.5.3.3.4 sap-no-override-rendering

Override of control methods is not allowed.

### Rule Details

The check detects override of getters, setters, and the functions `onBeforeRendering` and `onAfterRendering` for SAPUI5 controls.

The following patterns are considered warnings:

```
var myButton = new sap.m.Button();
myButton.onAfterRendering = function render(){foo.bar = 1; };
myButton.getWidth = function width(){return 3; };
```

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)

---

[sap-no-global-variable](#) [page 156]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

### 11.1.5.3.3.5 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 118]

#### Related Information

[sap-cross-application-navigation](#) [page 120]  
[sap-forbidden-window-property](#) [page 122]  
[sap-no-navigator](#) [page 123]  
[sap-no-override-rendering](#) [page 125]  
[sap-no-proprietary-browser-api](#) [page 127]  
[sap-no-sessionstorage](#) [page 129]  
[sap-no-ui5-prop-warning](#) [page 130]  
[sap-no-ui5base-prop](#) [page 131]  
[sap-timeout-usage](#) [page 133]  
[sap-ui5-no-private-prop](#) [page 135]  
[sap-usage-basemastercontroller](#) [page 137]  
[sap-message-toast](#) [page 139]  
[sap-no-absolute-component-path](#) [page 140]

---

sap-no-br-on-return [page 142]  
sap-no-dom-access [page 143]  
sap-no-dom-insertion [page 145]  
sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-define [page 152]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-hardcoded-url [page 159]  
sap-no-history-manipulation [page 160]  
sap-no-jquery-device-api [page 162]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.6 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 \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)

[sap-forbidden-window-property \[page 122\]](#)

[sap-no-navigator \[page 123\]](#)

[sap-no-override-rendering \[page 125\]](#)

[sap-no-override-storage-prototype \[page 126\]](#)

[sap-no-sessionstorage \[page 129\]](#)

[sap-no-ui5-prop-warning \[page 130\]](#)

[sap-no-ui5base-prop \[page 131\]](#)

[sap-timeout-usage \[page 133\]](#)

[sap-ui5-no-private-prop \[page 135\]](#)

[sap-usage-basemastercontroller \[page 137\]](#)

[sap-message-toast \[page 139\]](#)

[sap-no-absolute-component-path \[page 140\]](#)

[sap-no-br-on-return \[page 142\]](#)

[sap-no-dom-access \[page 143\]](#)

[sap-no-dom-insertion \[page 145\]](#)

[sap-no-dynamic-style-insertion \[page 146\]](#)

[sap-no-element-creation \[page 148\]](#)

[sap-no-encode-file-service \[page 149\]](#)

[sap-no-exec-command \[page 151\]](#)

[sap-no-global-define \[page 152\]](#)

[sap-no-global-event \[page 153\]](#)

[sap-no-global-selection \[page 155\]](#)

[sap-no-global-variable \[page 156\]](#)

[sap-no-hardcoded-color \[page 157\]](#)

[sap-no-hardcoded-url \[page 159\]](#)

[sap-no-history-manipulation \[page 160\]](#)

[sap-no-jquery-device-api \[page 162\]](#)

[sap-no-localhost \[page 163\]](#)

[sap-no-localstorage \[page 165\]](#)

[sap-no-location-reload \[page 166\]](#)

[sap-no-location-usage \[page 168\]](#)

## 11.1.5.3.3.7 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 \[page 118\]](#)

### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)

---

[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-reload \[page 166\]](#)  
[sap-no-location-usage \[page 168\]](#)

### 11.1.5.3.3.3.8 sap-no-ui5-prop-warning

Usage of private members of SAPUI5 objects is not allowed.

Private members of SAPUI5 objects must never be used in SAP Fiori applications. They can be changed by SAPUI5 at anytime and the application might not work anymore.

#### Rule Details

The rule checks usage of a member that has the same name as the following SAPUI5 members:

`sap.ui.model.odata.ODataModel, sap.ui.model.odata.v2.ODataModel` oData

#### False Positives

As the check cannot determine whether the property used is from a SAPUI5 object, there might be false positives if you defined a property with the same name in your own object. In this case, you can disable the check in your coding as follows:

```
/* eslint-disable sap-no-ui5base-prop */  
...some code false positives  
/* eslint-enable sap-no-ui5base-prop */
```

Parent topic: [Fiori JavaScript Validator Rules \[page 118\]](#)

#### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)

---

sap-no-override-storage-prototype [page 126]  
sap-no-proprietary-browser-api [page 127]  
sap-no-sessionstorage [page 129]  
sap-no-ui5base-prop [page 131]  
sap-timeout-usage [page 133]  
sap-ui5-no-private-prop [page 135]  
sap-usage-basemastercontroller [page 137]  
sap-message-toast [page 139]  
sap-no-absolute-component-path [page 140]  
sap-no-br-on-return [page 142]  
sap-no-dom-access [page 143]  
sap-no-dom-insertion [page 145]  
sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-define [page 152]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-hardcoded-url [page 159]  
sap-no-history-manipulation [page 160]  
sap-no-jquery-device-api [page 162]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.9 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:

|  |  |
|--|--|
| <b>sap.ui.base.ManagedObject</b>   | mProperties, mAggregations, mAssociations, mMethods,<br>oParent, aDelegates, aBeforeDelegates,<br>iSuppressInvalidate, oPropagatedProperties, oModels,<br>oBindingContexts, mBindingInfos, sBindingPath,<br>mBindingParameters, mBoundObjects  |
| <b>sap.ui.base.EventProvider</b>   | mEventRegistry, oEventPool   |
| <b>sap.ui.base.Event</b>   | oSource, mParameters, sId  |
| <b>sap.ui.model.odata.ODataModel,</b><br><b>sap.ui.model.odata.v2.ODataModel</b> | oServiceData, bCountSupported, bCache, oRequestQueue,<br>aBatchOperations, oHandler, mSupportedBindingModes,<br>sDefaultBindingMode, bJSON, aPendingRequestHandles,<br>aCallAfterUpdate, mRequests, mDeferredRequests,<br>mChangedEntities, mChangeHandles,<br>mDeferredBatchGroups, mChangeBatchGroups,<br>bTokenHandling, bWithCredentials, bUseBatch,<br>bRefreshAfterChange, sMaxDataServiceVersion,<br>bLoadMetadataAsync, bLoadAnnotationsJoined,<br>sAnnotationURI, sDefaultCountMode,<br>sDefaultOperationMode, oMetadataLoadEvent,<br>oMetadataFailedEvent, sRefreshBatchGroupId,<br>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 \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)

---

sap-no-ui5-prop-warning [page 130]  
sap-timeout-usage [page 133]  
sap-ui5-no-private-prop [page 135]  
sap-usage-basemastercontroller [page 137]  
sap-message-toast [page 139]  
sap-no-absolute-component-path [page 140]  
sap-no-br-on-return [page 142]  
sap-no-dom-access [page 143]  
sap-no-dom-insertion [page 145]  
sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-define [page 152]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-hardcoded-url [page 159]  
sap-no-history-manipulation [page 160]  
sap-no-jquery-device-api [page 162]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.10 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 \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-reload \[page 166\]](#)  
[sap-no-location-usage \[page 168\]](#)

## 11.1.5.3.3.11 sap-ui5-no-private-prop

Detects usage of private properties of SAPUI5 objects.

The SAP Fiori guideline prohibits the use of SAPUI5 properties or private functions.

### Rule Details

The rule detects SAPUI5 objects of the following namespaces:

- sap.ui.core
- sap.apf
- sap.ca.scfld.md
- sap.ca.ui
- sap.chart
- sap.collaboration
- sap.fiori
- sap.landvisz
- sap.m
- sap.makit
- sap.me
- sap.ndc
- sap.ovp
- sap.portal.ui5
- sap.suite.ui.common
- 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 \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)

[sap-forbidden-window-property \[page 122\]](#)

[sap-no-navigator \[page 123\]](#)

[sap-no-override-rendering \[page 125\]](#)

[sap-no-override-storage-prototype \[page 126\]](#)

[sap-no-proprietary-browser-api \[page 127\]](#)

[sap-no-sessionstorage \[page 129\]](#)

---

```
sap-no-ui5-prop-warning [page 130]
sap-no-ui5base-prop [page 131]
sap-timeout-usage [page 133]
sap-usage-basemastercontroller [page 137]
sap-message-toast [page 139]
sap-no-absolute-component-path [page 140]
sap-no-br-on-return [page 142]
sap-no-dom-access [page 143]
sap-no-dom-insertion [page 145]
sap-no-dynamic-style-insertion [page 146]
sap-no-element-creation [page 148]
sap-no-encode-file-service [page 149]
sap-no-exec-command [page 151]
sap-no-global-define [page 152]
sap-no-global-event [page 153]
sap-no-global-selection [page 155]
sap-no-global-variable [page 156]
sap-no-hardcoded-color [page 157]
sap-no-hardcoded-url [page 159]
sap-no-history-manipulation [page 160]
sap-no-jquery-device-api [page 162]
sap-no-localhost [page 163]
sap-no-localstorage [page 165]
sap-no-location-reload [page 166]
sap-no-location-usage [page 168]
```

### 11.1.5.3.3.12 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 \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-reload \[page 166\]](#)  
[sap-no-location-usage \[page 168\]](#)

## 11.1.5.3.3.13 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 \[page 118\]](#)

### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)

[sap-no-dom-access](#) [page 143]  
[sap-no-dom-insertion](#) [page 145]  
[sap-no-dynamic-style-insertion](#) [page 146]  
[sap-no-element-creation](#) [page 148]  
[sap-no-encode-file-service](#) [page 149]  
[sap-no-exec-command](#) [page 151]  
[sap-no-global-define](#) [page 152]  
[sap-no-global-event](#) [page 153]  
[sap-no-global-selection](#) [page 155]  
[sap-no-global-variable](#) [page 156]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

### 11.1.5.3.3.3.14 sap-no-absolute-component-path

Absolute paths to component `includes` are not allowed.

#### Rule Details

The rule checks if `includes` inside a component have a leading `/`.

The following patterns are considered warnings:

```
code:  
"sap.ui.core.UIComponent.extend('sap.ui.demoKit.explored.Component', {  
    + "metadata : { "  
    + "includes : [ "  
    + "'css/style2.css', "  
    + "'/css/style2.css', "  
    + "'/css/titles.css'"  
    + "], "  
    + "routing : { "  
    + "config : { "  
    + "routerClass : MyRouter, "  
    + "viewType : 'XML', "  
    + "viewPath :  
        'sap.ui.demoKit.explored.view', "  
        + "targetControl : 'splitApp', "  
        + "clearTarget : false "  
        + "}, "  
        + "routes : [ { "  
        + "pattern : 'entity/{id}/{part}', "  
        + "name : 'entityRoute'"  
    + "}"  
}  
});
```

```
+ "name : 'entity', "
+ "view : 'entity', "
+ "viewLevel : 3, "
+ "targetAggregation : 'detailPages' "
+ "}" ] + "}" + ")" + "});"
```

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-reload \[page 166\]](#)  
[sap-no-location-usage \[page 168\]](#)

## 11.1.5.3.3.15 sap-no-br-on-return

Detects the usage of `document.queryCommandSupported`.

This rule checks any call of `queryCommandSupported` on `document`. Calls with argument `sap-no-br-on-return` are not allowed because this is a browser specific command.

### Warning Message

`insertBrOnReturn` is not allowed since it is a Mozilla specific method, other browsers don't support that.

### Rule Details

The rule checks each `IconTabBar`. All tags with just one `IconTabFilter` will be found.

The following patterns are considered warnings:

```
var abc = document.queryCommandSupported('insertBrOnReturn');
```

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)

---

sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-define [page 152]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-hardcoded-url [page 159]  
sap-no-history-manipulation [page 160]  
sap-no-jquery-device-api [page 162]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.16 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 \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-reload \[page 166\]](#)  
[sap-no-location-usage \[page 168\]](#)

## 11.1.5.3.3.17 sap-no-dom-insertion

Usage of DOM insertion methods is not allowed.

The SAPUI5 guidelines do not allow insertion of elements into the DOM. Instead usage of a custom control should be considered.

### Rule Details

The rule detects all method calls of `insertBefore` , `appendChild` , `replaceChild` , `after` , `before` , `insertAfter` , `insertBefore` , `append` , `prepend` , `appendTo` , `prependTo` .

The following patterns are considered warnings:

```
$("#container").append("Test");
var list = document.getElementById("myList1");
list.insertBefore(node, list.childNodes[0]);
myObject.after(document.body);
```

### False Positives

There might be cases where the check produces a false positive, i.e. when you have a method containing one of the strings given above. In this case, you can change the method name or deactivate the rule by placing the following pseudo-comment block around your code. It is recommended to have your code reviewed before you enter such a pseudo-comment.

```
/*eslint-disable sap-no-dom-insertion*/
<your code>
/*eslint-enable sap-no-dom-insertion*/
```

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)

---

sap-no-ui5base-prop [page 131]  
sap-timeout-usage [page 133]  
sap-ui5-no-private-prop [page 135]  
sap-usage-basemastercontroller [page 137]  
sap-message-toast [page 139]  
sap-no-absolute-component-path [page 140]  
sap-no-br-on-return [page 142]  
sap-no-dom-access [page 143]  
sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-define [page 152]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-hardcoded-url [page 159]  
sap-no-history-manipulation [page 160]  
sap-no-jquery-device-api [page 162]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.18 sap-no-dynamic-style-insertion

Detects dynamic style insertion.

#### Warning Message

Dynamic style insertion, use library CSS or lessifier instead

#### Rule Details

The check detects any usage of `document.styleSheets`.

The following patterns are considered warnings:

```
var sheet = document.styleSheets[i];
```

```
var abc = document.styleSheets.length;
```

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-reload \[page 166\]](#)  
[sap-no-location-usage \[page 168\]](#)

## 11.1.5.3.3.3.19 sap-no-element-creation

Direct DOM insertion is not allowed.

The SAPUI5 guidelines do not allow creation of elements in the DOM. Instead usage of a custom control should be considered.

### Warning Message

Direct DOM insertion, create a custom control instead

### Rule Details

The rule detects all method calls of `createElement`, `createTextNode`, `createElementNS`, `createDocumentFragment`, `createComment`, `createAttribute`, `createEvent`.

The following patterns are considered warnings:

```
document.createElement('foo');
```

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)

---

[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-reload \[page 166\]](#)  
[sap-no-location-usage \[page 168\]](#)

### 11.1.5.3.3.20 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 \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-reload \[page 166\]](#)  
[sap-no-location-usage \[page 168\]](#)

## 11.1.5.3.3.21 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 \[page 118\]](#)

### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)

[sap-no-dynamic-style-insertion \[page 146\]](#)

[sap-no-element-creation \[page 148\]](#)

[sap-no-encode-file-service \[page 149\]](#)

[sap-no-global-define \[page 152\]](#)

[sap-no-global-event \[page 153\]](#)

[sap-no-global-selection \[page 155\]](#)

[sap-no-global-variable \[page 156\]](#)

[sap-no-hardcoded-color \[page 157\]](#)

[sap-no-hardcoded-url \[page 159\]](#)

[sap-no-history-manipulation \[page 160\]](#)

[sap-no-jquery-device-api \[page 162\]](#)

[sap-no-localhost \[page 163\]](#)

[sap-no-localstorage \[page 165\]](#)

[sap-no-location-reload \[page 166\]](#)

[sap-no-location-usage \[page 168\]](#)

### 11.1.5.3.3.22 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 \[page 118\]](#)

#### Related Information

[sap-cross-application-navigation \[page 120\]](#)

---

sap-forbidden-window-property [page 122]  
sap-no-navigator [page 123]  
sap-no-override-rendering [page 125]  
sap-no-override-storage-prototype [page 126]  
sap-no-proprietary-browser-api [page 127]  
sap-no-sessionstorage [page 129]  
sap-no-ui5-prop-warning [page 130]  
sap-no-ui5base-prop [page 131]  
sap-timeout-usage [page 133]  
sap-ui5-no-private-prop [page 135]  
sap-usage-basemastercontroller [page 137]  
sap-message-toast [page 139]  
sap-no-absolute-component-path [page 140]  
sap-no-br-on-return [page 142]  
sap-no-dom-access [page 143]  
sap-no-dom-insertion [page 145]  
sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-hardcoded-url [page 159]  
sap-no-history-manipulation [page 160]  
sap-no-jquery-device-api [page 162]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.23 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 \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)

---

[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

## 11.1.5.3.3.3.24 sap-no-global-selection

Usage of global selection is discouraged.

According to SAP Fiori design guidelines, it is not allowed to have an IconTabBar with just a single IconTabFilter.

### Warning Message

Global selection modification, only modify local selections

### Rule Details

The following patterns are considered warnings:

```
window.getSelection().rangeCount = 9;
```

**Parent topic:** [Fiori JavaScript Validator Rules](#) [page 118]

### Related Information

[sap-cross-application-navigation](#) [page 120]  
[sap-forbidden-window-property](#) [page 122]  
[sap-no-navigator](#) [page 123]  
[sap-no-override-rendering](#) [page 125]  
[sap-no-override-storage-prototype](#) [page 126]  
[sap-no-proprietary-browser-api](#) [page 127]  
[sap-no-sessionstorage](#) [page 129]  
[sap-no-ui5-prop-warning](#) [page 130]  
[sap-no-ui5base-prop](#) [page 131]  
[sap-timeout-usage](#) [page 133]  
[sap-ui5-no-private-prop](#) [page 135]  
[sap-usage-basemastercontroller](#) [page 137]

---

[sap-message-toast](#) [page 139]  
[sap-no-absolute-component-path](#) [page 140]  
[sap-no-br-on-return](#) [page 142]  
[sap-no-dom-access](#) [page 143]  
[sap-no-dom-insertion](#) [page 145]  
[sap-no-dynamic-style-insertion](#) [page 146]  
[sap-no-element-creation](#) [page 148]  
[sap-no-encode-file-service](#) [page 149]  
[sap-no-exec-command](#) [page 151]  
[sap-no-global-define](#) [page 152]  
[sap-no-global-event](#) [page 153]  
[sap-no-global-variable](#) [page 156]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

### 11.1.5.3.3.25 sap-no-global-variable

Global variables should not be used in SAP Fiori applications.

#### Rule Details

The rule checks if a variable is declared as global (defined outside of any function scope) and returns an error message in this case.

Allowed variables are = [ "undefined", "NaN", "arguments", "PDFJS", "console", "Infinity" ]

**Parent topic:** [Fiori JavaScript Validator Rules](#) [page 118]

#### Related Information

[sap-cross-application-navigation](#) [page 120]  
[sap-forbidden-window-property](#) [page 122]  
[sap-no-navigator](#) [page 123]

---

[sap-no-override-rendering](#) [page 125]  
[sap-no-override-storage-prototype](#) [page 126]  
[sap-no-proprietary-browser-api](#) [page 127]  
[sap-no-sessionstorage](#) [page 129]  
[sap-no-ui5-prop-warning](#) [page 130]  
[sap-no-ui5base-prop](#) [page 131]  
[sap-timeout-usage](#) [page 133]  
[sap-ui5-no-private-prop](#) [page 135]  
[sap-usage-basemastercontroller](#) [page 137]  
[sap-message-toast](#) [page 139]  
[sap-no-absolute-component-path](#) [page 140]  
[sap-no-br-on-return](#) [page 142]  
[sap-no-dom-access](#) [page 143]  
[sap-no-dom-insertion](#) [page 145]  
[sap-no-dynamic-style-insertion](#) [page 146]  
[sap-no-element-creation](#) [page 148]  
[sap-no-encode-file-service](#) [page 149]  
[sap-no-exec-command](#) [page 151]  
[sap-no-global-define](#) [page 152]  
[sap-no-global-event](#) [page 153]  
[sap-no-global-selection](#) [page 155]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

### 11.1.5.3.3.26 sap-no-hardcoded-color

Usage of hard coded colors is not allowed.

It is not allowed to style SAP Fiori applications with colors in JavaScript code as they will break the SAP Fiori themes.

## Rule Details

The following patterns are considered warnings:

```
$(\"<div id='lasso-selection-help' style='position: absolute; pointer-  
events: none; background: #cccccc; '></div>\")
```

## How to Fix

Do not specify colors in custom CSS but use the standard theme-dependent classes instead.

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)

---

[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

### 11.1.5.3.3.27 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 118]

#### Related Information

[sap-cross-application-navigation](#) [page 120]  
[sap-forbidden-window-property](#) [page 122]  
[sap-no-navigator](#) [page 123]  
[sap-no-override-rendering](#) [page 125]  
[sap-no-override-storage-prototype](#) [page 126]  
[sap-no-proprietary-browser-api](#) [page 127]

---

sap-no-sessionstorage [page 129]  
sap-no-ui5-prop-warning [page 130]  
sap-no-ui5base-prop [page 131]  
sap-timeout-usage [page 133]  
sap-ui5-no-private-prop [page 135]  
sap-usage-basemastercontroller [page 137]  
sap-message-toast [page 139]  
sap-no-absolute-component-path [page 140]  
sap-no-br-on-return [page 142]  
sap-no-dom-access [page 143]  
sap-no-dom-insertion [page 145]  
sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-define [page 152]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-history-manipulation [page 160]  
sap-no-jquery-device-api [page 162]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.28 sap-no-history-manipulation

Direct history manipulation is discouraged.

#### Warning Message

Direct history manipulation, does not work with deep links, use router and navigation events instead

## Rule Details

The following patterns are considered warnings:

```
window.history.back();  
  
history.go(-3);  
  
var personalHistory = window.history;  
personalHistory.back();
```

The following patterns are not considered warnings:

```
myNavBack : function(sRoute, mData) {  
    var oHistory = sap.ui.core.routing.History.getInstance();  
    var sPreviousHash = oHistory.getPreviousHash();  
    //The history contains a previous entry  
    if (sPreviousHash !== undefined) {  
        window.history.go(-1);  
    } else {  
        var bReplace = true; // otherwise we go backwards with a forward history  
        this.navTo(sRoute, mData, bReplace)  
    }  
,
```

**Parent topic:** [Fiori JavaScript Validator Rules \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)

---

[sap-no-element-creation](#) [page 148]  
[sap-no-encode-file-service](#) [page 149]  
[sap-no-exec-command](#) [page 151]  
[sap-no-global-define](#) [page 152]  
[sap-no-global-event](#) [page 153]  
[sap-no-global-selection](#) [page 155]  
[sap-no-global-variable](#) [page 156]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

### 11.1.5.3.3.29 sap-no-jquery-device-api

Usage of the jQuery device APIs is not allowed.

The jQuery device API is deprecated since SAPUI5 1.20. The respective functions of `sap.ui.Device` should be used instead.

#### Rule Details

The check looks for any call of `jQuery.device`.

The following patterns are considered warnings:

```
if (jQuery.device.is.android_phone === false) {}  
if ($.device.is.android_phone === false) {}
```

The following patterns are not considered warnings:

```
if (!sap.ui.Device.system.desktop) {  
    this.getView().byId("factSheetButton").setVisible(false);  
}
```

**Parent topic:** [Fiori JavaScript Validator Rules](#) [page 118]

#### Related Information

[sap-cross-application-navigation](#) [page 120]

---

sap-forbidden-window-property [page 122]  
sap-no-navigator [page 123]  
sap-no-override-rendering [page 125]  
sap-no-override-storage-prototype [page 126]  
sap-no-proprietary-browser-api [page 127]  
sap-no-sessionstorage [page 129]  
sap-no-ui5-prop-warning [page 130]  
sap-no-ui5base-prop [page 131]  
sap-timeout-usage [page 133]  
sap-ui5-no-private-prop [page 135]  
sap-usage-basemastercontroller [page 137]  
sap-message-toast [page 139]  
sap-no-absolute-component-path [page 140]  
sap-no-br-on-return [page 142]  
sap-no-dom-access [page 143]  
sap-no-dom-insertion [page 145]  
sap-no-dynamic-style-insertion [page 146]  
sap-no-element-creation [page 148]  
sap-no-encode-file-service [page 149]  
sap-no-exec-command [page 151]  
sap-no-global-define [page 152]  
sap-no-global-event [page 153]  
sap-no-global-selection [page 155]  
sap-no-global-variable [page 156]  
sap-no-hardcoded-color [page 157]  
sap-no-hardcoded-url [page 159]  
sap-no-history-manipulation [page 160]  
sap-no-localhost [page 163]  
sap-no-localstorage [page 165]  
sap-no-location-reload [page 166]  
sap-no-location-usage [page 168]

### 11.1.5.3.3.30 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 \[page 118\]](#)

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)

---

[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

### 11.1.5.3.3.3.31 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 118]

#### Related Information

[sap-cross-application-navigation](#) [page 120]  
[sap-forbidden-window-property](#) [page 122]  
[sap-no-navigator](#) [page 123]  
[sap-no-override-rendering](#) [page 125]  
[sap-no-override-storage-prototype](#) [page 126]  
[sap-no-proprietary-browser-api](#) [page 127]  
[sap-no-sessionstorage](#) [page 129]  
[sap-no-ui5-prop-warning](#) [page 130]  
[sap-no-ui5base-prop](#) [page 131]  
[sap-timeout-usage](#) [page 133]  
[sap-ui5-no-private-prop](#) [page 135]  
[sap-usage-basemastercontroller](#) [page 137]  
[sap-message-toast](#) [page 139]  
[sap-no-absolute-component-path](#) [page 140]  
[sap-no-br-on-return](#) [page 142]  
[sap-no-dom-access](#) [page 143]

[sap-no-dom-insertion](#) [page 145]  
[sap-no-dynamic-style-insertion](#) [page 146]  
[sap-no-element-creation](#) [page 148]  
[sap-no-encode-file-service](#) [page 149]  
[sap-no-exec-command](#) [page 151]  
[sap-no-global-define](#) [page 152]  
[sap-no-global-event](#) [page 153]  
[sap-no-global-selection](#) [page 155]  
[sap-no-global-variable](#) [page 156]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-location-reload](#) [page 166]  
[sap-no-location-usage](#) [page 168]

## 11.1.5.3.3.3.32 sap-no-location-reload

Detects location reload.

SAP Fiori guidelines do not allow `location.reload()`.

### Warning Message

`location.reload()` is not permitted.

### Rule Details

This check detects usage of `location.reload()`.

The following patterns are considered warnings:

```
location.reload();  
var mylocation = location; mylocation.reload();
```

**Parent topic:** [Fiori JavaScript Validator Rules](#) [page 118]

---

## Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)  
[sap-no-br-on-return \[page 142\]](#)  
[sap-no-dom-access \[page 143\]](#)  
[sap-no-dom-insertion \[page 145\]](#)  
[sap-no-dynamic-style-insertion \[page 146\]](#)  
[sap-no-element-creation \[page 148\]](#)  
[sap-no-encode-file-service \[page 149\]](#)  
[sap-no-exec-command \[page 151\]](#)  
[sap-no-global-define \[page 152\]](#)  
[sap-no-global-event \[page 153\]](#)  
[sap-no-global-selection \[page 155\]](#)  
[sap-no-global-variable \[page 156\]](#)  
[sap-no-hardcoded-color \[page 157\]](#)  
[sap-no-hardcoded-url \[page 159\]](#)  
[sap-no-history-manipulation \[page 160\]](#)  
[sap-no-jquery-device-api \[page 162\]](#)  
[sap-no-localhost \[page 163\]](#)  
[sap-no-localstorage \[page 165\]](#)  
[sap-no-location-usage \[page 168\]](#)

## 11.1.5.3.3.3.33 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 \[page 118\]](#)

### Related Information

[sap-cross-application-navigation \[page 120\]](#)  
[sap-forbidden-window-property \[page 122\]](#)  
[sap-no-navigator \[page 123\]](#)  
[sap-no-override-rendering \[page 125\]](#)  
[sap-no-override-storage-prototype \[page 126\]](#)  
[sap-no-proprietary-browser-api \[page 127\]](#)  
[sap-no-sessionstorage \[page 129\]](#)  
[sap-no-ui5-prop-warning \[page 130\]](#)  
[sap-no-ui5base-prop \[page 131\]](#)  
[sap-timeout-usage \[page 133\]](#)  
[sap-ui5-no-private-prop \[page 135\]](#)  
[sap-usage-basemastercontroller \[page 137\]](#)  
[sap-message-toast \[page 139\]](#)  
[sap-no-absolute-component-path \[page 140\]](#)

[sap-no-br-on-return](#) [page 142]  
[sap-no-dom-access](#) [page 143]  
[sap-no-dom-insertion](#) [page 145]  
[sap-no-dynamic-style-insertion](#) [page 146]  
[sap-no-element-creation](#) [page 148]  
[sap-no-encode-file-service](#) [page 149]  
[sap-no-exec-command](#) [page 151]  
[sap-no-global-define](#) [page 152]  
[sap-no-global-event](#) [page 153]  
[sap-no-global-selection](#) [page 155]  
[sap-no-global-variable](#) [page 156]  
[sap-no-hardcoded-color](#) [page 157]  
[sap-no-hardcoded-url](#) [page 159]  
[sap-no-history-manipulation](#) [page 160]  
[sap-no-jquery-device-api](#) [page 162]  
[sap-no-localhost](#) [page 163]  
[sap-no-localstorage](#) [page 165]  
[sap-no-location-reload](#) [page 166]

#### 11.1.5.3.3.4 Set JavaScript Rules for All Users

You can centrally define ESLint validation rules for JavaScript code checking, which then become the default rules for all projects for all users in the SAP 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  [Tools](#)  [SAP Cloud Platform Cockpit](#) .

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.

## 11.1.5.3.4 XML Validation

You can configure which XML validator to use in your project.

### Context

SAP Web IDE uses a default XML validator to check the syntax of your XML files.

#### [XML Semantic Validation \[page 170\]](#)

SAP Web IDE performs semantic validation on XML files.

#### [Fiori XML Validator Rules \[page 172\]](#)

The Fiori XML validator rules are used when building Fiori projects.

**Task overview:** [Checking Code \[page 112\]](#)

### Related Information

[Configure Code Checking \[page 113\]](#)

[Code Checking Annotations \[page 114\]](#)

[JavaScript Validation \[page 116\]](#)

[Validation of neo-app.json Files \[page 195\]](#)

[Using the Problems View \[page 196\]](#)

[Validation of manifest.json Files \[page 197\]](#)

## 11.1.5.3.4.1 XML Semantic Validation

SAP Web IDE performs semantic validation on XML files.

SAP Web IDE performs XML semantic validation including deprecation checks using the following rules.

| Rule                            | Description  |
|---------------------------------|--|
| XML_INVALID_AGGR_NODE           | The aggregation node for the specified control is incorrect. Enter the correct node name.                                    |
| XML_INVALID_CANDIDATE           | The candidate for the specified aggregation node is incorrect. Enter a candidate of the correct type.                        |
| XML_INVALID_MULTIPLICITY        | The specified aggregation is allowed only one child node but it has more than one child. You need to remove the extra nodes. |
| XML_INVALID_ID                  | The control ID is incorrect.   |
| XML_DUPLICATE_ID                | The control ID must be unique but it is the same as another control ID in the view. Change the ID so that it is unique       |
| XML_INVALID_CSS                 | The css class is incorrect.  |
| XML_INVALID_ASSOCIATION_VALUE   | The Association property value is incorrect.   |
| XML_INVALID_FORMATTER_FUNC      | The formatter function name is incorrect.  |
| XML_INVALID_EVENT_FUNC          | The event function name is incorrect.  |
| XML_INVALID_PROPERTY_NAME       | The property or event name does not exist for the control. Enter the correct property or event name.                         |
| XML_INVALID_PROPERTY_VALUE      | The value of the specified property is incorrect for the specified mode type.  |
| XML_INVALID_PROPERTY_ENUM_VALUE | The specified property value is incorrect for the specified enum type of the specified property.                             |
| XML_DEPRECATED_CONTROL          | The specified control is deprecated since the specified version.   |
| XML_DEPRECATED_PROPERTY         | The specified property is deprecated since the specified version.  |
| XML_DEPRECATED_EVENT            | The specified event is deprecated since the specified version.   |
| XML_DEPRECATED_AGGREGATION      | The specified aggregation is deprecated since the specified version.   |

**Parent topic:** [XML Validation \[page 170\]](#)

## Related Information

[Fiori XML Validator Rules \[page 172\]](#)

## 11.1.5.3.4.2 Fiori XML Validator Rules

The Fiori XML validator rules are used when building Fiori projects.

### [DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)

Detects buttons with text and icons.

### [DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)

Detects lists and tables with showNoData attributes that are not set to true.

### [DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)

Detects duplicate icons in IconTabBars.

### [DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)

Detects single IconTabFilter in IconTabBars.

### [XML\\_COMMONS\\_USAGE \[page 179\]](#)

Detects the usage of sap.ui.commons objects

### [XML\\_DEPRECATED \[page 180\]](#)

Checks for deprecated controls.

### [XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)

Checks Dialog, Popover, ResponsivePopover, and ActionSheet in views.

### [XML\\_FORM\\_USAGE \[page 182\]](#)

Checks usage of sap.ui.commons.form.

### [XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)

Checks accessibility for icons

### [XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)

Checks accessibility for icons.

### [XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)

Checks accessibility for images

### [XML\\_LAYOUT\\_USAGE \[page 186\]](#)

Detects the usage of sap.ui.commons.layout tags

### [XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)

Checks usage of absolute media\_src URLs.

### [XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)

Checks for stable IDs for controls.

### [XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)

Detects missing title attributes.

### [XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)

Detects missing title attributes.

### [XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)

Detects missing title attributes.

### [XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)

Checks usage of FileUpload and AddPicture tags.

### [XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

Checks your setting of serviceRefreshInterval

---

**Parent topic:** [XML Validation \[page 170\]](#)

## Related Information

[XML Semantic Validation \[page 170\]](#)

### 11.1.5.3.4.2.1 DG\_XML\_FOOTER\_BUTTON\_TEXT\_ICON

Detects buttons with text and icons.

According to 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:** [Fiori XML Validator Rules](#) [page 172]

## Related Information

[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA](#) [page 174]  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS](#) [page 176]  
[DG\\_XML\\_NO\\_SINGLE\\_TAB](#) [page 177]  
[XML\\_COMMONS\\_USAGE](#) [page 179]  
[XML\\_DEPRECATED](#) [page 180]  
[XML\\_DIALOG\\_IN\\_VIEW](#) [page 181]  
[XML\\_FORM\\_USAGE](#) [page 182]  
[XML\\_ICON\\_ACCESSIBILITY](#) [page 183]  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY](#) [page 184]  
[XML\\_IMAGE\\_ACCESSIBILITY](#) [page 185]  
[XML\\_LAYOUT\\_USAGE](#) [page 186]  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER](#) [page 187]  
[XML\\_MISSING\\_STABLE\\_ID](#) [page 188]  
[XML\\_PAGE\\_ACCESSIBILITY](#) [page 190]  
[XML\\_TABLE\\_ACCESSIBILITY](#) [page 191]  
[XML\\_TITLE\\_ACCESSIBILITY](#) [page 192]  
[XML\\_UPLOAD\\_IN\\_VIEW](#) [page 193]  
[XML\\_BOOKMARK\\_PERFORMANCE](#) [page 194]

## 11.1.5.3.4.2.2 DG\_XML\_LIST\_BASE\_SHOW\_NO\_DATA

Detects lists and tables with showNoData attributes that are not set to true.

According to 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:** [Fiori XML Validator Rules \[page 172\]](#)

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)

---

[XML\\_MISSING\\_STABLE\\_ID](#) [page 188]  
[XML\\_PAGE\\_ACCESSIBILITY](#) [page 190]  
[XML\\_TABLE\\_ACCESSIBILITY](#) [page 191]  
[XML\\_TITLE\\_ACCESSIBILITY](#) [page 192]  
[XML\\_UPLOAD\\_IN\\_VIEW](#) [page 193]  
[XML\\_BOOKMARK\\_PERFORMANCE](#) [page 194]

### 11.1.5.3.4.2.3 DG\_XML\_NO\_DUPLICATE\_ICONS

Detects duplicate icons in IconTabBars.

According to Fiori Design Guidelines an IconTabFilter cannot contain the same icon twice.

#### Warning Message

A specific icon must not occur twice in a tab bar

#### Rule Details

The rule checks each `IconTabFilter` tag in all `IconTabBar` tags. All tags with the same icon value are found.

The following patterns are considered warnings:

```
<IconTabBar id="idIconTabBar">
  <items>
    <IconTabFilter
      icon="sap-icon://begin"
      text="Heavy"
      key="Heavy" />
    <IconTabSeparator />
    <IconTabFilter
      icon="sap-icon://begin"
      text="Overweight"
      key="Overweight" />
  </items>
</IconTabBar>
```

#### False Positives

The rule can't detect the visibility of an element. So there might be cases when a duplicate icon is not visible. In such a cases, please ignore the warning.

**Parent topic:** [Fiori XML Validator Rules](#) [page 172]

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

### 11.1.5.3.4.2.4 DG\_XML\_NO\_SINGLE\_TAB

Detects single IconTabFilter in IconTabBars.

According to Fiori Design Guidelines, an IconTabBar cannot contain just a single IconTabFilter.

#### Warning Message

An IconTabBar should have more than one IconTab

#### Rule Details

The rule checks each IconTabBar and finds all tags with just one IconTabFilter.

The following patterns are considered warnings:

```
<IconTabBar
    id="idIconTabBar"
    select="handleIconTabBarSelect"
    class="sapUiResponsiveContentPadding">
    <items>
```

```
<IconTabFilter  
    showAll="true"  
    count="{!!ProductCollectionStats/Counts/Total}"  
    text="Products"  
    keys="All" />  
<IconTabSeparator />  
</items>  
</IconTabBar>
```

## False Positives

If there are multiple `IconTabFilter` tags, and some of them are invisible, the rule does not detect a violation if just one `IconTabFilter` is visible.

**Parent topic:** [Fiori XML Validator Rules \[page 172\]](#)

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.5 XML\_COMMONS\_USAGE

Detects the usage of sap.ui.commons objects

According to Fiori Architectural Guidelines controls from sap.ui.commons 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:** [Fiori XML Validator Rules \[page 172\]](#)

### Related Information

- [DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)
- [DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)
- [DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)
- [DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)
- [XML\\_DEPRECATED \[page 180\]](#)
- [XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)
- [XML\\_FORM\\_USAGE \[page 182\]](#)
- [XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)
- [XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)
- [XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)
- [XML\\_LAYOUT\\_USAGE \[page 186\]](#)
- [XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)
- [XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)
- [XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)
- [XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)
- [XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)

---

[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.6 XML\_DEPRECATION

Checks for deprecated controls.

Controls marked as deprecated in the API Reference documentation are no longer intended to be used. They will not get feature updates in the future.

### Warning Message

A deprecated control is used in the XML view

### Rule Details

For each control, the check scans for deprecation in the API documentation in the corresponding version loaded from Nexus.

**Parent topic:** [Fiori XML Validator Rules \[page 172\]](#)

### Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)

---

[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.7 XML\_DIALOG\_IN\_VIEW

Checks Dialog, Popover, ResponsivePopover, and ActionSheet 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 xml fragment instantiation.

**Parent topic:** [Fiori XML Validator Rules \[page 172\]](#)

---

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

### 11.1.5.3.4.2.8 XML\_FORM\_USAGE

Checks usage of sap.ui.commons.form.

The `sap.ui.commons.form` object must not be used anymore according to the 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:** [Fiori XML Validator Rules \[page 172\]](#)

---

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

### 11.1.5.3.4.2.9 XML\_ICON\_ACCESSIBILITY

Checks accessibility for icons

According to 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:** [Fiori XML Validator Rules \[page 172\]](#)

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.10 XML\_ICON\_BUTTON\_ACCESSIBILITY

Checks accessibility for icons.

According to 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:** Fiori XML Validator Rules [page 172]

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.11 XML\_IMAGE\_ACCESSIBILITY

Checks accessibility for images

According to 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:** [Fiori XML Validator Rules \[page 172\]](#)

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.12 XML\_LAYOUT\_USAGE

Detects the usage of sap.ui.commons.layout tags

According to Fiori Architectural Guidelines, usage of sap.ui.commons.layout is not allowed. Instead, sap.ui.layout 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:** [Fiori XML Validator Rules \[page 172\]](#)

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)

[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)

[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)

[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)

[XML\\_COMMONS\\_USAGE \[page 179\]](#)

[XML\\_DEPRECATED \[page 180\]](#)

[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)

[XML\\_FORM\\_USAGE \[page 182\]](#)

[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)

[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)

[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)

[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)

[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)

[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)

[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)

[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)

[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)

[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.13 XML\_METADATA\_MEDIA\_SRC\_WITHOUT\_FORMATTER

Checks usage of absolute media\_src URLs.

If your 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:** [Fiori XML Validator Rules \[page 172\]](#)

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.14 XML\_MISSING\_STABLE\_ID

Checks for stable IDs for controls.

If a control has no static ID defined in the XML view, during runtime no stable ID can be guaranteed. This can be a problem, for example, for in-application help or UI testing tools. See 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:** [Fiori XML Validator Rules \[page 172\]](#)

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)

[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)

[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)

[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)

[XML\\_COMMONS\\_USAGE \[page 179\]](#)

[XML\\_DEPRECATED \[page 180\]](#)

[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)

[XML\\_FORM\\_USAGE \[page 182\]](#)

[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)

[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)

[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)

[XML\\_LAYOUT\\_USAGE \[page 186\]](#)

[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)

---

[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.15 XML\_PAGE\_ACCESSIBILITY

Detects missing title attributes.

According to 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:** [Fiori XML Validator Rules \[page 172\]](#)

### Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)

---

[XML\\_MISSING\\_STABLE\\_ID](#) [page 188]  
[XML\\_TABLE\\_ACCESSIBILITY](#) [page 191]  
[XML\\_TITLE\\_ACCESSIBILITY](#) [page 192]  
[XML\\_UPLOAD\\_IN\\_VIEW](#) [page 193]  
[XML\\_BOOKMARK\\_PERFORMANCE](#) [page 194]

## 11.1.5.3.4.2.16 XML\_TABLE\_ACCESSIBILITY

Detects missing title attributes.

According to 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:** [Fiori XML Validator Rules](#) [page 172]

### Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON](#) [page 173]  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA](#) [page 174]  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS](#) [page 176]  
[DG\\_XML\\_NO\\_SINGLE\\_TAB](#) [page 177]  
[XML\\_COMMONS\\_USAGE](#) [page 179]  
[XML\\_DEPRECATED](#) [page 180]  
[XML\\_DIALOG\\_IN\\_VIEW](#) [page 181]  
[XML\\_FORM\\_USAGE](#) [page 182]  
[XML\\_ICON\\_ACCESSIBILITY](#) [page 183]  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY](#) [page 184]  
[XML\\_IMAGE\\_ACCESSIBILITY](#) [page 185]  
[XML\\_LAYOUT\\_USAGE](#) [page 186]  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER](#) [page 187]

---

[XML\\_MISSING\\_STABLE\\_ID](#) [page 188]  
[XML\\_PAGE\\_ACCESSIBILITY](#) [page 190]  
[XML\\_TITLE\\_ACCESSIBILITY](#) [page 192]  
[XML\\_UPLOAD\\_IN\\_VIEW](#) [page 193]  
[XML\\_BOOKMARK\\_PERFORMANCE](#) [page 194]

## 11.1.5.3.4.2.17 XML\_TITLE\_ACCESSIBILITY

Detects missing title attributes.

According to 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:** [Fiori XML Validator Rules](#) [page 172]

### Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON](#) [page 173]  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA](#) [page 174]  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS](#) [page 176]  
[DG\\_XML\\_NO\\_SINGLE\\_TAB](#) [page 177]  
[XML\\_COMMONS\\_USAGE](#) [page 179]  
[XML\\_DEPRECATED](#) [page 180]  
[XML\\_DIALOG\\_IN\\_VIEW](#) [page 181]  
[XML\\_FORM\\_USAGE](#) [page 182]  
[XML\\_ICON\\_ACCESSIBILITY](#) [page 183]  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY](#) [page 184]  
[XML\\_IMAGE\\_ACCESSIBILITY](#) [page 185]  
[XML\\_LAYOUT\\_USAGE](#) [page 186]  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER](#) [page 187]  
[XML\\_MISSING\\_STABLE\\_ID](#) [page 188]

---

[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.18 XML\_UPLOAD\_IN\_VIEW

Checks usage of FileUpload and AddPicture tags.

An application might include a UI element that offers file upload functionality (`sap.ca.ui.FileUpload` or `sap.ca.ui.AddPicture`). To make sure that only allowed content can be uploaded, the file content must be checked by a virus scanner before it is stored on the database (details can be found in the security guidelines).

### Warning Message

Uploaded files shall be sent to VSI 2.0 before stored on DB

### Rule Details

The rule will find every `AddPicture` tag and `FileUpload` tag with `uploadEnabled` attribute set to true.

### How to fix

Determine which virus scan profile is used and include this information in the documentation. After sending a mail (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:** [Fiori XML Validator Rules \[page 172\]](#)

### Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)

[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_BOOKMARK\\_PERFORMANCE \[page 194\]](#)

## 11.1.5.3.4.2.19 XML\_BOOKMARK\_PERFORMANCE

Checks your setting of serviceRefreshInterval

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 backend. 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 300 (5 minutes).

### Warning Message

A value of more than 0 and less than 300 for the property serviceRefreshIntervall may result in performance limitations.

### Rule Details

The following patterns are considered warnings:

```
<core:View xmlns:core="sap.ui.core" xmlns:ui="sap.ca.ui"
    xmlns:suite="sap.suite.ui.commons" xmlns:layout="sap.ui.layout" xmlns:html="http://
    www.w3.org/1999/xhtml"
    xmlns="sap.m" xmlns:footerbar="sap.ushell.ui.footerbar"
    controllerName="cus.crm.myaccounts.view.S360">
    <Page id="page" title="{i18n>DETAIL_TITLE}" showNavButton="true">
        <content>
            <layout:Grid class ="sapSuiteUtiHeaderGrid sapSuiteUti
sapCRMmyAccountsHeader" defaultSpan="L6 M6 S12" vSpacing="0" >
```

```
<layout:content>
  ...
<footerbar:AddBookmarkButton serviceRefreshInterval="20" />
```

**Parent topic:** [Fiori XML Validator Rules \[page 172\]](#)

## Related Information

[DG\\_XML\\_FOOTER\\_BUTTON\\_TEXT\\_ICON \[page 173\]](#)  
[DG\\_XML\\_LIST\\_BASE\\_SHOW\\_NO\\_DATA \[page 174\]](#)  
[DG\\_XML\\_NO\\_DUPLICATE\\_ICONS \[page 176\]](#)  
[DG\\_XML\\_NO\\_SINGLE\\_TAB \[page 177\]](#)  
[XML\\_COMMONS\\_USAGE \[page 179\]](#)  
[XML\\_DEPRECATED \[page 180\]](#)  
[XML\\_DIALOG\\_IN\\_VIEW \[page 181\]](#)  
[XML\\_FORM\\_USAGE \[page 182\]](#)  
[XML\\_ICON\\_ACCESSIBILITY \[page 183\]](#)  
[XML\\_ICON\\_BUTTON\\_ACCESSIBILITY \[page 184\]](#)  
[XML\\_IMAGE\\_ACCESSIBILITY \[page 185\]](#)  
[XML\\_LAYOUT\\_USAGE \[page 186\]](#)  
[XML\\_METADATA\\_MEDIA\\_SRC\\_WITHOUT\\_FORMATTER \[page 187\]](#)  
[XML\\_MISSING\\_STABLE\\_ID \[page 188\]](#)  
[XML\\_PAGE\\_ACCESSIBILITY \[page 190\]](#)  
[XML\\_TABLE\\_ACCESSIBILITY \[page 191\]](#)  
[XML\\_TITLE\\_ACCESSIBILITY \[page 192\]](#)  
[XML\\_UPLOAD\\_IN\\_VIEW \[page 193\]](#)

### 11.1.5.3.5 Validation of neo-app.json Files

A project's neo-app.json file is validated on opening in the code editor.

When you open or make changes to your project's neo-app.json application descriptor file in SAP Web IDE, there is automatic schema validation, and errors are displayed as annotations.

For more information about the required format of the neo-app.json file, see [Application Descriptor File](#).

**Parent topic:** [Checking Code \[page 112\]](#)

## Related Information

[Configure Code Checking \[page 113\]](#)

[Code Checking Annotations \[page 114\]](#)

[JavaScript Validation \[page 116\]](#)

[XML Validation \[page 170\]](#)

[Using the Problems View \[page 196\]](#)

[Validation of manifest.json Files \[page 197\]](#)

### 11.1.5.3.6 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 \[page 112\]](#)

## Related Information

[Configure Code Checking \[page 113\]](#)

[Code Checking Annotations \[page 114\]](#)

[JavaScript Validation \[page 116\]](#)

[XML Validation \[page 170\]](#)

[Validation of neo-app.json Files \[page 195\]](#)

[Validation of manifest.json Files \[page 197\]](#)

### 11.1.5.3.7 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 \[page 112\]](#)

## Related Information

[Configure Code Checking \[page 113\]](#)

[Code Checking Annotations \[page 114\]](#)

[JavaScript Validation \[page 116\]](#)

[XML Validation \[page 170\]](#)

[Validation of neo-app.json Files \[page 195\]](#)

[Using the Problems View \[page 196\]](#)

## 11.1.5.4 Developing Apps Using SAP Fiori Elements

This section contains information about creating SAP Fiori apps using SAP Fiori elements in SAP Web IDE.

SAP Fiori elements provide designs for UI patterns and predefined templates for commonly used application patterns. You can use SAP Fiori elements to create SAP Fiori applications based on OData services and annotations requiring no JavaScript UI coding. The resulting app uses predefined views and controllers that are provided centrally, so no application-specific view instances are required. The SAPUI5 runtime interprets the metadata and annotations of the underlying OData service and uses the corresponding views while starting the SAP Fiori application.

The predefined views and controllers ensure UI design consistency across similar apps. The metadata-driven development model also significantly reduces the amount of front-end code for each app. This means that the developer can focus on the business logic.

By using the wizard for creating projects in SAP Web IDE, you can generate an SAP Fiori app based on SAP Fiori elements. The following SAP Fiori elements are available:

- [List Report \[page 199\]](#)

You can use the list report template to work with a large list of items. It combines powerful functions for filtering large lists with different ways of displaying the resulting item list.

- [Object Page \[page 199\]](#)

On the object page you can display, edit, and create objects, as well as save drafts. It is suitable for both simple objects and more complex, multi-faceted objects. The object page view offers the best possible support for multiple devices.

- [Analytical List Page \[page 201\]](#)

Analytical list page is an SAP Fiori application for performing detailed analytics. Use this template to build a landing page for your SAP Fiori applications. It helps you to quickly and easily combine transactional and analytical data in the form of chart or table visualizations.

- [Overview Page \[page 202\]](#)

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](#).

### 11.1.5.4.1 Prerequisites

Provides a list of the prerequisites you must meet before using the SAP Fiori Fiori Elements application.

- You have subscribed to SAP Web IDE and completed all configuration steps, including Git settings, destinations, and connection to remote systems.

For more information, see [Getting Started \[page 24\]](#).

- 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](#).

## 11.1.5.4.2 Create List Report and Object Page Applications

You use SAP Web IDE to create an SAP Fiori app containing a list report and object pages.

### Context

As an app developer, you must define a configuration in the SAP Web IDE. The main aspects are the destinations to the back-end metadata, navigation between pages, and page design (as pages may contain several templates).

 Note

This step-by-step procedure is also available as a video: .

### Procedure

- In the SAP Web IDE, from the [File](#) menu, choose  [New](#)  [Project from Template](#). The system starts the wizard for new projects.
- Follow the guided procedure:

| Step                    | Action   |
|-------------------------|--|
| 1. Template Selection   | Select <a href="#">List Report Page</a> and click <a href="#">Next</a> .   |
| 2. Basic Information    | The following fields are mandatory: <ul style="list-style-type: none"> <li>○ <a href="#">Project Name</a></li> <li>○ <a href="#">Title</a></li> <li>○ <a href="#">Application Component Hierarchy</a></li> </ul> Choose <a href="#">Next</a> . |
| 3. Data Connection      | <ol style="list-style-type: none"> <li>Choose <a href="#">Service Catalog</a> and select the desired data source from the list.</li> <li>Choose a service and then choose <a href="#">Next</a>.</li> </ol>                                     |
| 4. Annotation Selection | Select the required annotation file and then choose <a href="#">Next</a> .   |

| Step                      | Action  |
|---------------------------|---|
| 5. Template Customization | <ol style="list-style-type: none"> <li>Under <i>Data Binding</i>, complete the fields as follows:           <ul style="list-style-type: none"> <li><i>OData Collection</i> – this is a mandatory field</li> <li><i>OData Navigation</i> – select the relevant navigation option</li> </ul> </li> <li>Choose <i>Next</i> and <i>Finish</i>.</li> </ol> |

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.

### 11.1.5.4.3 Create a Worklist Application

You use SAP Web IDE to create an SAP Fiori app containing a worklist.

#### Context

#### Procedure

- In the SAP Web IDE, from the *File* menu, choose *New* *Project from Template*. The system starts the wizard for new projects.
- Follow the guided procedure:

| Step                  | Action  |
|-----------------------|---|
| 1. Template Selection | Select <i>Worklist</i> and click <i>Next</i> .  |
| 2. Basic Information  | <p>The following fields are mandatory:</p> <ul style="list-style-type: none"> <li><i>Project Name</i></li> <li><i>Title</i></li> <li><i>Application Component Hierarchy</i></li> </ul> <p>Choose <i>Next</i>.</p> |
| 3. Data Connection    | <ol style="list-style-type: none"> <li>Choose <i>Service Catalog</i> and select the desired data source from the list.</li> <li>Choose a service and then choose <i>Next</i>.</li> </ol>                          |

| Step                      | Action  |
|---------------------------|---|
| 4. Annotation Selection   | Select the required annotation file and then choose <i>Next</i> .   |
| 5. Template Customization | <ol style="list-style-type: none"> <li>1. Under <i>Data Binding</i>, complete the fields as follows:           <ul style="list-style-type: none"> <li>o <i>OData Collection</i> – this is a mandatory field</li> <li>o <i>OData Navigation</i> – select the relevant navigation option</li> </ul> </li> <li>2. Choose <i>Next</i> and <i>Finish</i>.</li> </ol> |

3. Open your project (already selected in project list).
4. Open the *webapp* folder.
5. Select *Component.js* and choose *Run*.

#### 11.1.5.4.4 Create an Analytical List Page Application

You can create your own analytical list page applications using the Analytical List Page plugin in SAP Web IDE.

#### Procedure

1. Click  *File* > *New* > *Project from Template* .
2. Select *Category* > *SAP Fiori Elements* > *Analytical List Page*.
3. Enter *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. Select *OData Collection* for Data binding. (Optional) Enter the required *App Descriptor Settings*, or you can modify later in the App Descriptor file.
7. Confirm the project information and choose *Finish*.
8. Build and run your application.
  - o Open your project (already selected in project list).
  - o Open the *webapp* folder.
  - o Select *Component.js* and choose *Run*.

## 11.1.5.4.5 Create an Overview Page Application

You can create your own overview page applications using the Overview Page plugin in SAP Web IDE.

### Procedure

1. Click  [File](#)  [New](#)  [Project from Template](#).
2. Select *Category* > *SAP Fiori Elements* > *Overview Page Application*.
3. Enter *Project Name*, and fill in the *App Descriptor Data*
4. Select a data source and service from the list.
5. (Optional) Add annotation files.  
If you add more than one annotation file, you can determine files based on the order in which they are loaded.
6. Fill in the required overview page details, paying attention to the following:
  - The default value of the Datasource Alias is the name of the data service selected in the *Data Connection* step.
  - Users filter content on the overview page according to the fields defined in the selected *EntityType*.
7. Confirm the project information and choose *Finish*.

### Results

The wizard creates the project structure in the workspace under a new folder with the project name that you specified.

### Related Information

[Add Cards to an Overview Page \[page 202\]](#)

## 11.1.5.4.5.1 Add Cards to an Overview Page

Add cards to populate the overview page that you created.

### Procedure

1. In the *Development* tab, select the overview page project that you created, and choose  [File](#)  [New](#)  [Card](#).

### **i** Note

If you have created a Multi-Target Application, select the Multi-Target Application project that you created, and choose **File > New > Card**.

2. Select an existing data source, or create a new data source for the card.
3. Select one of the following card types:

| Option                     | Description  |
|----------------------------|--|
| <b>List Card</b>           | Displays an array of items in a vertical list. A number of list types are available.   |
| <b>Link List Card</b>      | Displays an array of items in a vertical list with title, picture, icon, or subtitle.  |
| <b>Table Card</b>          | Displays items in a three-column table.  |
| <b>Stack Card</b>          | A collection of single-object cards. When opened, users can take action on the individual items in the stack.  |
| <b>Analytic Chart Card</b> | These type of chart cards show data in a variety of formats. For example, they can be cards that display data in a series of data points connected by straight lines, that use bubbles to visualize the data dimension, or in columns or stacked columns to help view multiple measures or dimensions. |

### **i** Note

Overview Page lets you to configure view switch and KPI header section for List, Table, and Analytic Chart cards. Selecting the checkbox:

- *Select to enable view switch for this card* lets you configure multiple views and apply different filtering and sorting options in the card.
- *Select to add KPI header for this card* lets you configure KPI header information in the card.

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](#).

## 11.1.5.5 Using the Outline Pane for JavaScript Files

The Outline pane helps you to understand the structure of JavaScript files and to navigate through the code.

### Context

The Outline pane provides a clear view of the main entities and coding structure of JavaScript files by displaying the full hierarchy of a file's functions and objects.

When you open the Outline pane, it expands to the first level of the hierarchy. Functions and objects are denoted by different icons. You can expand and collapse the nodes to explore nested functions and objects.

- Function nodes display the function name and parameters, if they exist, and function assignment names of anonymous functions assigned to variables. Other unnamed functions are displayed with the label *function()*.
- Object nodes display the object name. Unnamed objects are displayed with the label *object{}*.

When you click a node in the Outline pane, the corresponding line or block is selected in the Code Editor pane, enabling you to navigate quickly to the desired object and function code.

The Outline pane refreshes when:

- You open a JavaScript file
- You move between JavaScript file Code Editor panes
- You save changes to an open JavaScript file

If there is a fatal syntax error in the JavaScript code, the Outline pane displays up to the location of the error in the code.

When you move from a JavaScript file Code Editor pane to a Code Editor pane that does not support the outline feature, the Outline pane is cleared.

### Note

- You can view the outline of files that contain a maximum of 130,000 characters.
- When a function is a parameter in a constructor, for example, `var x = new Object(param1, function() {});`, it is not displayed as a separate node in the outline.
- Array content is not displayed in the outline.
- Static variable definition is supported; however, dynamic assignment is not supported. In the following example, `a` appears as an object in the outline:

```
var a = x;
var x = function() {...};
```

## Procedure

To open the Outline pane for a specific JavaScript file:

1. Ensure that the required JavaScript file Code Editor pane is in focus.
2. In the right sidebar, choose  (Outline).  
The Outline pane opens displaying the outline expanded to the first level.

## 11.1.5.6 Creating an HTML5 Application Descriptor File

Create an HTML5 Application Descriptor file in a project that has not been imported or created via the project wizards. This is required for defining cloud connectivity for external resources required by the application, and allows you to run the application properly in the SAP Web IDE.

### Procedure

In SAP Web IDE:

1. Select your project folder (for example, *myProject*).
2. From the *File* menu, select ► *New* ► *HTML5 Application Descriptor* ▶.
3. Enter routes for paths `/resources` and `/test-resources` with service `sapui5` in your `neo-app.json` file:

```
{  
    "path": "/resources",  
    "target": {  
        "type": "service",  
        "name": "sapui5",  
        "entryPath": "/resources"  
    },  
    "description": "SAPUI5 Resources"  
},  
{  
    "path": "/test-resources",  
    "target": {  
        "type": "service",  
        "name": "sapui5",  
        "entryPath": "/test-resources"  
    },  
    "description": "SAPUI5 Test Resources"  
}
```

For more information, see [Application Descriptor File](#).

4. Enter routes for your remote systems. For example, you can add an ABAP OData system for the path `/sap/opu/odata` and a destination `myRemoteDestination`. For connecting remote systems, see [Connect to ABAP Systems \[page 28\]](#).

```
{  
    "path": "/sap/opu/odata",  
    "target": {  
        "type": "destination",  
        "name": "myRemoteDestination",  
        "entryPath": "/sap/opu/odata"  
    },  
    "description": "Target OData system"  
}
```

## 11.1.6 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 206]

You can create tests in your project using a wizard.

### 11.1.6.1 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   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.

### 11.1.7 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.

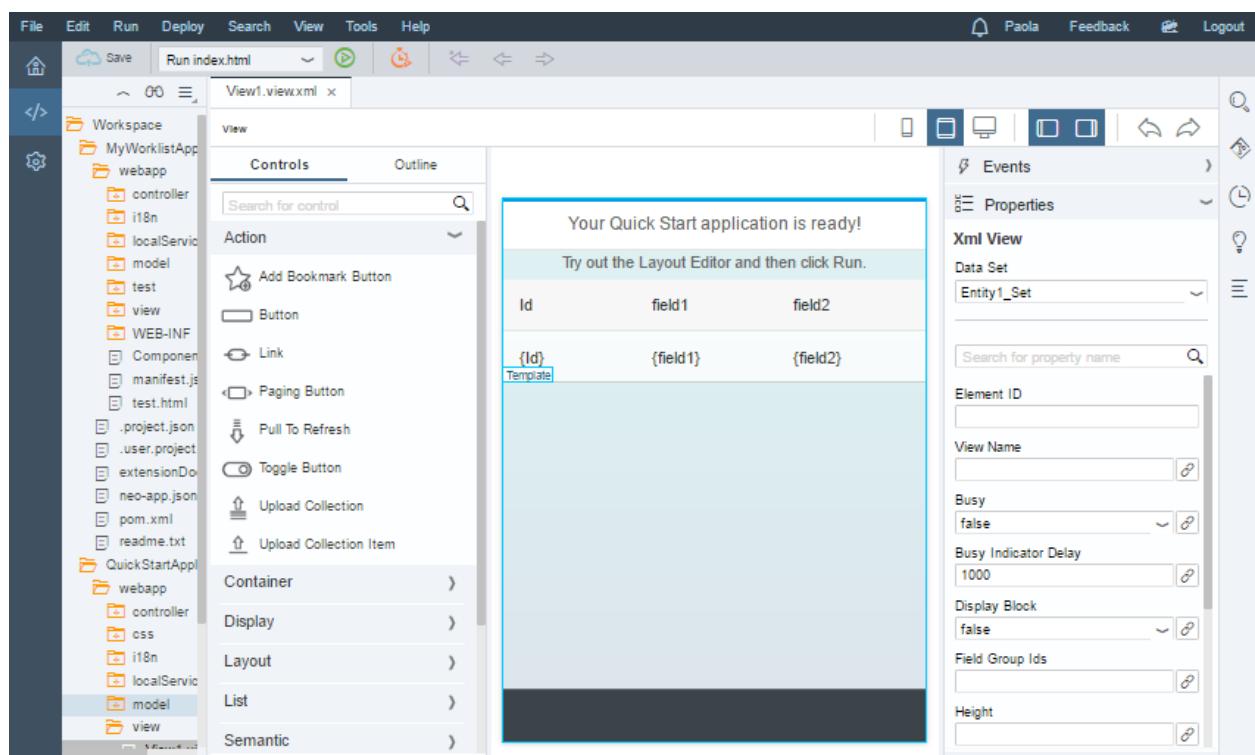
#### i Note

The layout editor is not supported in the Safari browser.

For more information on known issues regarding the layout editor, see [Known Issues \[page 429\]](#).

## Layout Editor Landscape

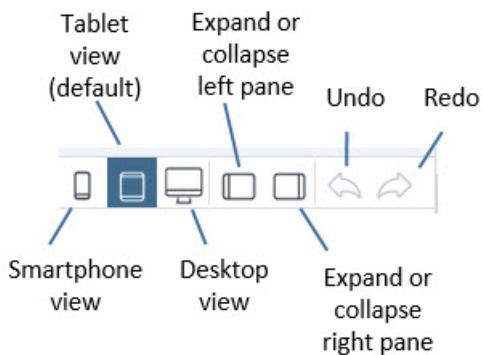
The layout editor is composed of a canvas, a pane on the left that includes the *Controls* and *Outline* tabs, and a pane on the right that includes the *Events* and *Properties* panes.



## Toolbar

The buttons on the layout editor toolbar allow you to:

- Change the device format of the canvas to smartphone, tablet, or desktop view.
- Change the application's SAPUI5 version.
- Expand and collapse the panes to the right and left of the canvas.
  - The pane on the left side includes the *Controls* and *Outline* tabs.
  - The pane on the right side includes the *Properties* and *Events* panes.
- Undo and redo actions.



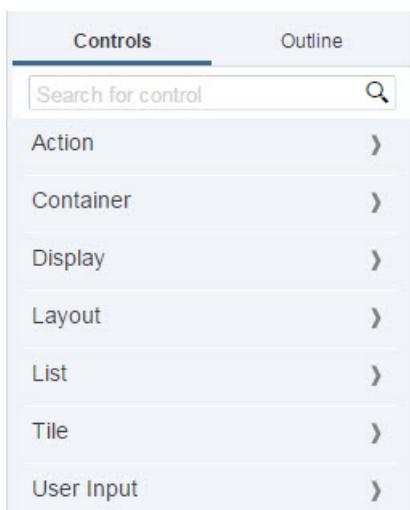
## Controls Tab

You can expand or collapse each section by clicking the arrow on each section header. You can also search for controls by entering the control name in the search field at the top of the *Controls* tab. The relevant sections expand to display the controls that match the search criteria.

### Note

Make sure to delete the search criteria if you want to expand other sections.

You can drag and drop controls from the *Controls* tab onto the canvas. For more information, see [Add Controls from the Controls Tab \[page 216\]](#).



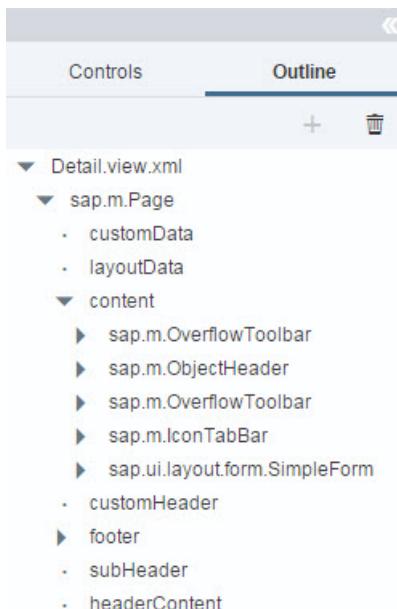
You can find the list of available controls in [SAPUI5 Controls Supported in the Layout Editor \[page 225\]](#).

## 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.

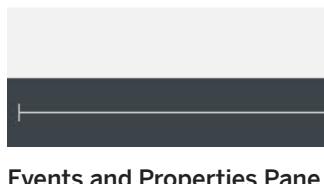
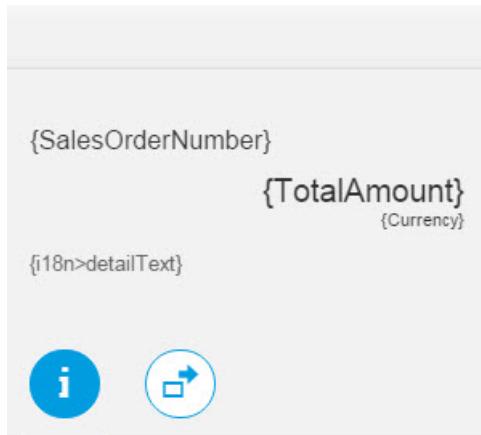
For more information, see [Add Controls from the Outline Tab \[page 213\]](#).



## 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.

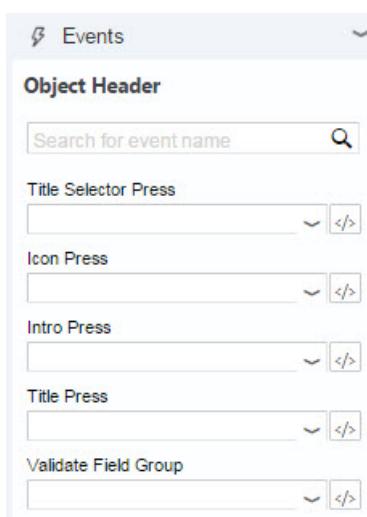


### Events and Properties Pane

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.

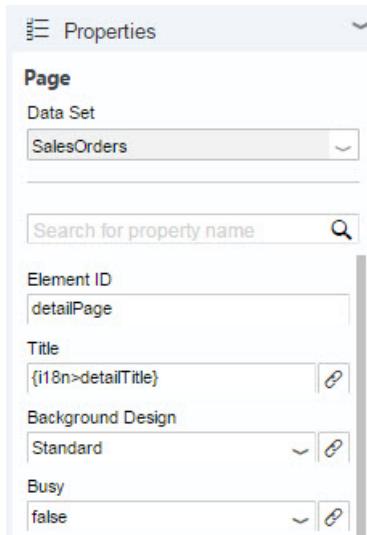


## Properties Pane

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 \[page 219\]](#), [Bind Data to a Simple Control \[page 221\]](#), and [Bind Data to an Aggregate-Type Control \[page 222\]](#).



### 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*.

[Working with the Layout Editor \[page 212\]](#)

An overview of the steps required to edit a project using the layout editor.

#### [Creating a Quick Start Application with the Layout Editor \[page 224\]](#)

Quickly create a new application using the layout editor.

#### [SAPUI5 Controls Supported in the Layout Editor \[page 225\]](#)

Provides a list of SAPUI5 controls that are supported in the layout editor.

#### [Try It: Build an Application with the Layout Editor \[page 238\]](#)

Get an overview of the features that are available with the layout editor by following this tutorial for building an application.

## Related Information

[Working with the Layout Editor \[page 212\]](#)

[SAPUI5 Controls Supported in the Layout Editor \[page 225\]](#)

### 11.1.7.1 Working with the Layout Editor

An overview of the steps required to edit a project using the layout editor.

## Prerequisites

- You have opened SAP Web IDE in Google Chrome or Internet Explorer 11.
- You have created a project using the *Fiori Master-Detail Application* template. In the `view` folder of this project, you can edit the following views:
  - `Detail.view.xml`
  - `Master.view.xml`

## Procedure

1. From the context menu of the XML view, choose  [Open with > Layout Editor](#).
2. Edit 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 \[page 213\]](#)

You can add controls to the canvas from the *Outline* tab.

#### [Delete Controls from the Outline Tab \[page 214\]](#)

You can remove controls from the *Outline* tab.

#### [Create a New Function \[page 215\]](#)

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

#### [Create a Navigation Between Views \[page 215\]](#)

You can create a navigation between views using the layout editor.

#### [Add Controls from the Controls Tab \[page 216\]](#)

Add controls to the canvas by using drag and drop functionality.

#### [Keyboard Support \[page 217\]](#)

Use the keyboard to move selected controls or navigate within the view that you opened with the layout editor.

#### [Layout Editor Binding Capabilities \[page 218\]](#)

In the layout editor, you can bind properties of controls or control aggregations to data fields, `i18n` models, and label annotations.

### 11.1.7.1.1 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:** [Working with the Layout Editor \[page 212\]](#)

#### Related Information

[Delete Controls from the Outline Tab \[page 214\]](#)

---

[Create a New Function \[page 215\]](#)  
[Create a Navigation Between Views \[page 215\]](#)  
[Add Controls from the Controls Tab \[page 216\]](#)  
[Keyboard Support \[page 217\]](#)  
[Layout Editor Binding Capabilities \[page 218\]](#)  
[Delete Controls from the Outline Tab \[page 214\]](#)  
[SAPUI5 Controls Supported in the Layout Editor \[page 225\]](#)

## 11.1.7.1.2 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 *Delete*  button.

The control is removed from the view.

**Task overview:** [Working with the Layout Editor \[page 212\]](#)

### Related Information

[Add Controls from the Outline Tab \[page 213\]](#)  
[Create a New Function \[page 215\]](#)  
[Create a Navigation Between Views \[page 215\]](#)  
[Add Controls from the Controls Tab \[page 216\]](#)  
[Keyboard Support \[page 217\]](#)  
[Layout Editor Binding Capabilities \[page 218\]](#)  
[Add Controls from the Outline Tab \[page 213\]](#)  
[SAPUI5 Controls Supported in the Layout Editor \[page 225\]](#)

### 11.1.7.1.3 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*.

 Note

The function name you enter must be a valid JavaScript function name.

After entering the name for your new function, the code is created by the layout editor in the code editor. To go directly to that code, click the *Go to code*  icon next to the new function name in the *Events* pane.

**Task overview:** [Working with the Layout Editor \[page 212\]](#)

#### Related Information

- [Add Controls from the Outline Tab \[page 213\]](#)
- [Delete Controls from the Outline Tab \[page 214\]](#)
- [Create a Navigation Between Views \[page 215\]](#)
- [Add Controls from the Controls Tab \[page 216\]](#)
- [Keyboard Support \[page 217\]](#)
- [Layout Editor Binding Capabilities \[page 218\]](#)
- [Layout Editor \[page 207\]](#)

### 11.1.7.1.4 Create a Navigation Between Views

You can create a navigation between views using the layout editor.

#### Prerequisites

- You have enabled the *Storyboard* perspective in the *Preferences* perspective, on the *Features* tab.

- You have created a freestyle project.

## Procedure

1. In the *Development* perspective, in your Workspace, right-click the view to which you want to add a navigation, then in the context menu, choose ► *Open With* ► *Layout Editor* ▾.

The view opens in the layout editor.

2. From the *Controls* pane, drag a control and drop it onto the view canvas.

3. Select the control and in the *Events* pane, in the *Press* field, click the Open Menu  button and select *Navigate To*.

4. In the *Navigate To* dialog box, in the *View* dropdown list, select the view to which you want to navigate, choose *OK*, then choose *Save*.

In the *Storyboard* perspective, the navigation now appears as an arrow between the views you configured.

**Task overview:** [Working with the Layout Editor \[page 212\]](#)

## Related Information

[Add Controls from the Outline Tab \[page 213\]](#)

[Delete Controls from the Outline Tab \[page 214\]](#)

[Create a New Function \[page 215\]](#)

[Add Controls from the Controls Tab \[page 216\]](#)

[Keyboard Support \[page 217\]](#)

[Layout Editor Binding Capabilities \[page 218\]](#)

### 11.1.7.1.5 Add Controls from the Controls Tab

Add controls to the canvas by using drag and drop functionality.

Select the control that you want to add to the canvas from the *Controls* tab on the left side of the layout editor area and drag it to the drop target on the canvas. A tooltip displays the drop targets as you drag the control around the canvas.

#### Example

If you want to add an *HBox* container with a *Button* control to your view, do the following:

1. Choose the *Controls* tab.
2. Open the *Container* section by clicking the arrow to the right of the section title.
3. Select the *HBox* control and drag it to the canvas. Drop it at the position where you want it to appear.

4. Open the *Action* section by clicking the arrow to the right of the section title.
5. Select the *Button* control and drag it to the canvas.
6. Drop the *Button* control onto the *HBox* container.

**Parent topic:** [Working with the Layout Editor \[page 212\]](#)

## Related Information

- [Add Controls from the Outline Tab \[page 213\]](#)
- [Delete Controls from the Outline Tab \[page 214\]](#)
- [Create a New Function \[page 215\]](#)
- [Create a Navigation Between Views \[page 215\]](#)
- [Keyboard Support \[page 217\]](#)
- [Layout Editor Binding Capabilities \[page 218\]](#)

### 11.1.7.1.6 Keyboard Support

Use the keyboard to move selected controls or navigate within the view that you opened with the layout editor.

#### Selecting Controls

If you selected a control in the canvas of the layout editor, you can move the selection using the arrow keys:

- **UP ARROW**: moves the selection to the parent of the selected control
- **DOWN ARROW**: moves the selection to the child of the selected control
- **LEFT ARROW**: moves the selection to the control that is up/to the right of the selected control (within the same aggregation)
- **RIGHT ARROW**: moves the selection to the control that is down/to the left of the selected control (within the same aggregation)
- **CTRL**+ 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:** [Working with the Layout Editor \[page 212\]](#)

## Related Information

[Add Controls from the Outline Tab \[page 213\]](#)

[Delete Controls from the Outline Tab \[page 214\]](#)

[Create a New Function \[page 215\]](#)

[Create a Navigation Between Views \[page 215\]](#)

[Add Controls from the Controls Tab \[page 216\]](#)

[Layout Editor Binding Capabilities \[page 218\]](#)

## 11.1.7.1.7 Layout Editor Binding Capabilities

In the layout editor, you can bind properties of controls or control aggregations to data fields, `i18n` models, and label annotations.

**Parent topic:** [Working with the Layout Editor \[page 212\]](#)

## Related Information

[Add Controls from the Outline Tab \[page 213\]](#)

[Delete Controls from the Outline Tab \[page 214\]](#)

[Create a New Function \[page 215\]](#)

[Create a Navigation Between Views \[page 215\]](#)

[Add Controls from the Controls Tab \[page 216\]](#)

[Keyboard Support \[page 217\]](#)

## 11.1.7.1.7.1 Binding Data

In the layout editor, you can bind properties of controls or control aggregations to an artifact in the OData service.

### Prerequisites

You have defined a data set for the view that you are working on, by doing one of the following:

- If you are opening a view that has no data set defined for it, the *Data Binding* dialog box opens, where you can define a data set.
- Select the view, and define the data set from the dropdown list.

#### Caution

If you change the data set that is defined for the current view, the existing data bindings might become invalid.

### Overview

The following types of bindings are possible:

- Properties of controls
- Aggregations of controls

#### 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.

#### Note

If your application does not consume an OData service, you can add the *OData Service* component to it.

#### [Define Entity Set \[page 220\]](#)

Before you can bind data to a control, you need to define an entity set.

#### [Bind Data to a Simple Control \[page 221\]](#)

You can bind data to a simple control.

#### [Bind Data to an Aggregate-Type Control \[page 222\]](#)

You can bind data to an aggregate-type control, which creates a template.

#### [Bind to the i18n Model \[page 223\]](#)

You can bind a control property to the i18n model or create a new i18n entry.

#### [Bind to a Label Annotation \[page 223\]](#)

You can bind a control property to a label annotation.

## Related Information

[SAPUI5 API Reference](#)

### 11.1.7.1.7.1.1 Define Entity Set

Before you can bind data to a control, you need to define an entity set.

#### Prerequisites

- You have created a project in the layout editor. For more information, see [Working with the Layout Editor \[page 212\]](#).
- You have dragged a control from the *Controls* pane to the canvas. For more information, see [Add Controls from the Controls Tab \[page 216\]](#).

#### Context

#### Procedure

1. In the canvas, select the control for which you want to define an entity set.
2. In the *Properties* pane, click the icon.

The *Select Entity Set* dialog box opens.

3. Select and configure one of the entity sets for your control.

##### Note

Most users should use either of the first two entity sets listed below.

- *Use ancestor's entity set.*
- *Define entity set and set the selected control as template.*
- *Define dummy entity set for the selected control.*

##### Restriction

The "dummy" entity set is only for advanced users.

4. Choose OK.

---

**Task overview:** [Binding Data \[page 219\]](#)

## Related Information

[Bind Data to a Simple Control \[page 221\]](#)

[Bind Data to an Aggregate-Type Control \[page 222\]](#)

[Bind to the i18n Model \[page 223\]](#)

[Bind to a Label Annotation \[page 223\]](#)

### 11.1.7.1.7.1.2 Bind Data to a Simple Control

You can bind data to a simple control.

#### Procedure

1. On the canvas, select the desired control for which you want to define data binding.
2. In the *Properties* pane to the right of the canvas, do one of the following:
  - To the right of the property to which you want to bind data, click the *Binding*  button, and then in the *Data Binding* dialog box:
    1. In the *Data Fields* list, double-click one or more data fields that you want to add to the expression.  
The data fields are automatically concatenated to the string in the *Expression* box.
    2. Click *OK* or first manually edit the expression string and then click *OK*.
  - In the *Properties* pane, for a field or dropdown list, manually enter the required expression within curly brackets {...} according to the data set that you selected.

**Task overview:** [Binding Data \[page 219\]](#)

## Related Information

[Define Entity Set \[page 220\]](#)

[Bind Data to an Aggregate-Type Control \[page 222\]](#)

[Bind to the i18n Model \[page 223\]](#)

[Bind to a Label Annotation \[page 223\]](#)

## 11.1.7.1.7.1.3 Bind Data to an Aggregate-Type Control

You can bind data to an aggregate-type control, which creates a template.

### Prerequisites

- You have defined an entity set. For more information, see [Define Entity Set \[page 220\]](#).

### 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 \[page 219\]](#)

### Related Information

[Define Entity Set \[page 220\]](#)

[Bind Data to a Simple Control \[page 221\]](#)

[Bind to the i18n Model \[page 223\]](#)

[Bind to a Label Annotation \[page 223\]](#)

[Aggregation Binding](#)

## 11.1.7.1.7.1.4 Bind to the i18n Model

You can bind a control property to the i18n model or create a new i18n entry.

### Procedure

1. On the canvas, select the desired control for which you want to define i18n model binding.
2. In the *Properties* pane to the right of the canvas, do one of the following:
  - To the right of the property to which you want to bind data, click the *Binding*  button, and then in the dialog box that appears:
    1. From the drop down list, select *i18n*, double-click one or more entries that you want to add to the expression.  
The entries are automatically concatenated to the string in the *Expression* box.
    2. Click *OK* or first manually edit the expression string and then click *OK*.
  - In the *Properties* pane, for a field or dropdown list, manually enter the required expression within curly brackets {...}.
  - Click + to add a new i18n entry.

**Task overview:** [Binding Data \[page 219\]](#)

### Related Information

[Define Entity Set \[page 220\]](#)

[Bind Data to a Simple Control \[page 221\]](#)

[Bind Data to an Aggregate-Type Control \[page 222\]](#)

[Bind to a Label Annotation \[page 223\]](#)

## 11.1.7.1.7.1.5 Bind to a Label Annotation

You can bind a control property to a label annotation.

### Context

You can bind a control property to a label annotation that resides in the Odata metadata file. Other annotation files are not supported.

## Procedure

1. On the canvas, select the desired control for which you want to define a label annotation binding.
2. In the *Properties* pane to the right of the canvas, do one of the following:
  - To the right of the property to which you want to bind data, click the *Binding*  button, and then in the dialog box that appears:
    1. From the drop down list, select *Labels*, double-click one or more annotation that you want to add to the expression.  
The annotations are automatically concatenated to the string in the *Expression* box.
    2. Click *OK* or first manually edit the expression string and then click *OK*.
  - In the *Properties* pane, for a field or dropdown list, manually enter the required expression within curly brackets {...}.

**Task overview:** [Binding Data \[page 219\]](#)

## Related Information

[Define Entity Set \[page 220\]](#)

[Bind Data to a Simple Control \[page 221\]](#)

[Bind Data to an Aggregate-Type Control \[page 222\]](#)

[Bind to the i18n Model \[page 223\]](#)

## 11.1.7.2 Creating a Quick Start Application with the Layout Editor

Quickly create a new application using the layout editor.

## Context

You can create a new application using the layout editor directly from the SAP Web IDE Welcome page.

After adding controls and binding data to them, you can run the application using the provided mock data.

## Procedure

1. On the SAP Web IDE Welcome page, click the *Quick Start with Layout Editor* tile.

Alternatively, choose   .

2. Provide a name for your new application.

A new application opens in the layout editor.

**i Note**

This new application will run on mock data and is not for productive use.

3. Add any controls that you require by doing one of the following:
  - Drag and drop them onto the canvas from the *Controls* tab on the left side of the canvas.
  - Add them from the *Outline* tab on the left side of the canvas.
4. In the properties pane to the left of the canvas, bind data to each control as required.

## Related Information

[Layout Editor \[page 207\]](#)

[Add Controls from the Outline Tab \[page 213\]](#)

[Binding Data \[page 219\]](#)

### 11.1.7.3 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 \[page 213\]](#).

**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<br>sap.m.ActionListItem                        | Button that is used to fire actions when pressed.   |
| Action Select<br>sap.m.ActionSelect                              | Provides a list of predefined items that allows end users to choose options and additionally trigger some actions.  |
| Add Bookmark Button<br>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<br>sap.ui.vbm.AnalyticMap                           | Renders a map based on a GeoJSON source.  |
| App<br>sap.m.App   | The root element of an SAPUI5 mobile application. It inherits from the <code>NavContainer</code> control and thus provides its navigation capabilities. <code>App</code> provides certain header tags to the HTML page that are relevant for mobile apps. |
| Bar<br>sap.m.Bar   | Centers a control like a title while having other controls on its left and right.   |
| Busy Indicator<br>sap.ui.core.BusyIndicator                      | Provides methods to show or hide a waiting animation that covers the whole page and blocks user interaction.  |
| Button<br>sap.m.Button   | Allows users to trigger actions.  |
| Calendar<br>sap.ui.unified.Calendar                              | Basic calendar that is used for <code>DatePickers</code> .  |
| Calendar Legend<br>sap.ui.unified.CalendarLegend                 | Legend for the <code>Calendar</code> control. Displays special date colors with their corresponding description.  |
| Carousel<br>sap.m.Carousel                                       | Navigates through a list of controls by swiping right or left.  |
| Check Box<br>sap.m.CheckBox                                      | Allows the user to select one or multiple items from a list.  |
| Column<br>sap.m.Column   | Allows definition of column-specific properties that are applied when rendering a <code>List</code> control.  |

| SAPUI5 Control Name                              | Description  |
|--|--|
| Column List Item<br>sap.m.ColumnListItem         | Used with cell aggregation to create rows for the sap.m.Table control.   |
| Combo Box<br>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.  |
| Custom List Item<br>sap.m.CustomListItem         | With content aggregation, can be used to customize standard list items that are not provided by SAPUI5. ListItem type is applied to CustomListItem as well.<br><br><b>i Note</b><br>Content aggregation allows any control. Complex responsive layout controls (such as Table and Form) should not be aggregated as content. |
| Custom Tile<br>sap.m.CustomTile                  | Displays application-specific content in the Tile control.   |
| Date Picker<br>sap.m.DatePicker                  | Date input control with a calendar used as a date picker.  |
| Detail Page<br>sap.m.semantic.DetailPage         | An sap.m.semantic.ShareMenuPage control that supports certain semantic buttons that have default semantic-specific properties and are eligible for content aggregation.  |
| Display List Item<br>sap.m.DisplayListIItem      | Used to represent a label and a value.   |
| Feed Input<br>sap.m.FeedInput                    | Allows the user to enter text for a new feed entry and then post it.   |
| Feed List Item<br>sap.m.FeedListIItem            | Provides a set of properties for text, sender information, and time stamp.   |
| Flex Box<br>sap.m.FlexBox                        | Builds the container for a flexible box layout.  |
| Fullscreen Page<br>sap.m.semantic.FullscreenPage | An sap.m.semantic.ShareMenuPage control that supports certain semantic buttons that have default semantic-specific properties and are eligible for content aggregation.  |
| Geo Map<br>sap.ui.vbm.GeoMap                     | A map control that allows the user to position multiple visual objects on top of a map.  |

| SAPUI5 Control Name                                 | Description   |
|---|---|
| Grid<br>sap.ui.layout.Grid                          | Layout that positions its child controls in a 12-column flow layout.                                      |
| Group Header List Item<br>sap.m.GroupHeaderListItem | Used to display the title of a group and act as a separator between groups in sap.m.List and sap.m.Table. |
| HBox<br>sap.m.HBox                                  | Builds the container for a horizontal flexible box layout.  |
| Horizontal Layout<br>sap.ui.layout.HorizontalLayout | Provides support for horizontal alignment of controls.  |
| Icon<br>sap.ui.core.Icon                            | Uses an embedded font instead of a pixel image.   |
| Icon Tab Bar<br>sap.m.IconTabBar                    | Represents a collection of tabs with associated content.  |
| Icon Tab Filter<br>sap.m.IconTabFilter              | Represents a selectable item inside an Icon Tab Bar control.  |
| Icon Tab Header<br>sap.m.IconTabHeader              | Displays a number of Icon Tab Filter and Icon Tab Separator controls.                                     |
| Icon Tab Separator<br>sap.m.IconTabSeparator        | Icon used to separate two Icon Tab Filter controls.   |
| Image<br>sap.m.Image                                | Wrapper around the IMG tag.   |
| Input<br>sap.m.Input                                | Allows users to input data.   |
| Input List Item<br>sap.m.InputListIem               | List item used for a label and an input field.  |
| Invisible Text<br>sap.ui.core.InvisibleText         | Used to bring hidden texts to the UI for screen reader support.   |
| Item<br>sap.ui.core.Item                            | Control base type.  |

| SAPUI5 Control Name                         | Description   |
|---|---|
| Label<br>sap.m.Label                        | Used in SAPUI5 mobile applications to provide label text for other controls.  |
| Link<br>sap.m.Link                          | Used to trigger actions or to navigate to other applications or web pages.  |
| List<br>sap.m.List                          | Provides a container for all types of list items.   |
| List Item<br>sap.ui.core.ListItem           | Used in lists or list-like controls, such as <code>DropdownBox</code> .   |
| Master Page<br>sap.m.semantic.MasterPage    | An <code>sap.m.semantic.SemanticPage</code> control that supports certain semantic buttons that have default semantic-specific properties and are eligible for content aggregation. |
| Message Strip<br>sap.m.MessageStrip         | Allows the embedding of application-related messages in the application.  |
| Nav Container<br>sap.m.NavContainer         | Handles hierarchical navigation between <code>Page</code> controls or other fullscreen controls.  |
| Object Attribute<br>sap.m.ObjectAttribute   | Displays a text field that can be normal or active.   |
| Object Header<br>sap.m.ObjectHeader         | Allows the user to easily identify a special object.  |
| Object Identifier<br>sap.m.ObjectIdentifier | Display control that allows the user to easily identify a specific object.  |
| Object List Item<br>sap.m.ObjectListItem    | Display control that provides summary information about an object as an item in a list.   |
| Object Number<br>sap.m.ObjectNumber         | Displays number and number unit properties for an object.   |
| Object Status<br>sap.m.ObjectStatus         | Status information that can be either text with a value state, or an icon.  |
| Overflow Toolbar<br>sap.m.OverflowToolbar   | Container based on <code>sap.m.Toolbar</code> that provides overflow when its content does not fit in the visible area.   |

| SAPUI5 Control Name                             | Description  |
|---|--|
| Page<br>sap.m.Page                              | Basic container for a mobile application screen.   |
| Paging Button<br>sap.m.PagingButton             | Allows users to navigate between items and entities.   |
| Panel<br>sap.m.Panel                            | Container for controls that has a header and content.  |
| Progress Indicator<br>sap.m.ProgressIndicator   | Shows the progress of a process in a graphical way.  |
| Pull To Refresh<br>sap.m.PullToRefresh          | Triggers the refresh event.  |
| Radio Button<br>sap.m.RadioButton               | Control similar to CheckBox, but it allows the user to choose only one of a predefined set of options.   |
| Radio Button Group<br>sap.m.RadioButtonGroup    | Used as a wrapper for a group of sap.m.RadioButton controls, which then can be used as a single UI element.  |
| Rating Indicator<br>sap.m.RatingIndicator       | Used to rate content.  |
| Search Field<br>sap.m.SearchField               | Allows users to input a search string.   |
| Segmented Button<br>sap.m.SegmentedButton       | Horizontal control made of multiple buttons, which can display a title or an image.  |
| Select<br>sap.m.Select                          | Provides a list of items that allows the user to select an item.   |
| Select List<br>sap.m.SelectList                 | Displays a list of items that allows the user to select an item.   |
| Semantic Page<br>sap.m.semantic.SemanticPage    | An enhanced sap.m.Page control that can contain controls with semantic meaning. 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<br>sap.m.semantic.ShareMenuPage | An sap.m.semantic.SemanticPage control that supports a Share menu in the footer.   |

| SAPUI5 Control Name                          | Description   |
|--|---|
| Simple Form<br>sap.ui.layout.form.SimpleForm | Provides an API for creating simple forms. Inside a <code>SimpleForm</code> control, a <code>Form</code> control is created along with its <code>FormContainers</code> control and <code>FormElements</code> control, but the complexity in the API is removed. |
| Slider<br>sap.m.Slider                       | User interface control that allows the user to adjust values within a specified numerical range.  |
| Standard List Item<br>sap.m.StandardListItem | List item that provides the most common use cases, such as image, title, and description.   |
| Standard Tile<br>sap.m.StandardTile          | Displayed in the <code>Tile</code> container.   |
| Switch<br>sap.m.Switch                       | User interface control on mobile devices that is used for switching between binary states.  |
| Table<br>sap.m.Table                         | Provides a set of sophisticated and convenient functions for responsive table design.   |
| Text<br>sap.m.Text                           | Used for embedding longer text paragraphs that need text wrapping into your application.  |
| Text Area<br>sap.m.TextArea                  | Allows multiline text input.  |
| Tile Container<br>sap.m.TileContainer        | Container that arranges same-size tiles on carousel pages.  |
| Title<br>sap.ui.core.Title                   | Used to aggregate other controls.   |
| Toggle Button<br>sap.m.ToggleButton          | Control that toggles between pressed and normal state.  |
| Toolbar<br>sap.m.Toolbar                     | Horizontal container that is usually used to display buttons, labels, selects, and other input controls.  |
| Toolbar Separator<br>sap.m.ToolbarSeparator  | Creates a visual separator between toolbar items.   |
| Toolbar Spacer<br>sap.m.ToolbarSpacer        | Adds horizontal space between toolbar items.  |

| SAPUI5 Control Name                                  | Description  |
|--|--|
| Upload Collection<br>sap.m.UploadCollection          | Allows users to upload single or multiple files.                       |
| Upload Collection Item<br>sap.m.UploadCollectionItem | Provides information about uploaded files.                             |
| VBox<br>sap.m.VBox                                   | Builds the container for a vertical flexible box layout.               |
| Vertical Layout<br>sap.ui.layout.VerticalLayout      | Layout in which the content controls are rendered one below the other. |

## Outline Tab

The SAPUI5 controls listed below are available only from the [Outline](#) tab in the layout editor.

### 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. |

| SAPUI5 Control Name              | Description   |
|----------------------------------|---|
| 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. |
| 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. |

| SAPUI5 Control Name                          | Description  |
|--|--|
| 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.  |
| 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.   |

| SAPUI5 Control Name                           | Description   |
|---|---|
| sap.suite.ui.commons.GenericTile              | Tile control that displays a title, description, and customizable main area.  |
| 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 <code>FormContainer</code> controls, each of which consists of <code>FormElement</code> controls.   |
| sap.ui.layout.form.FormContainer        | Group inside a <code>Form</code> .  |
| sap.ui.layout.form.FormElement          | Row in a <code>FormContainer</code> control.  |
| sap.ui.layout.form.FormLayout           | Base layout used to render a <code>Form</code> control.   |
| sap.ui.layout.form.GridLayout           | Renders a <code>Form</code> control using an HTML table-based grid.   |
| sap.ui.layout.form.ResponsiveGridLayout | Renders a <code>Form</code> control using a responsive grid.  |
| sap.ui.layout.form.ResponsiveLayout     | Renders a <code>Form</code> control with a responsive layout.   |
| sap.ui.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 <code>Browse ....</code>   |
| sap.ui.unified.FileUploaderParameter    | Represents a parameter for the <code>FileUploader</code> , which is rendered as a hidden input field.   |
| sap.ui.unified.Menu                     | Interactive element that provides a choice of different actions to the user.  |
| sap.ui.unified.MenuItem                 | Standard item used inside a menu. Represents an action that can be selected by a user in the menu or that can be used as a submenu that organizes the actions hierarchically. |
| sap.ui.unified.ShellOverlay             | Opened in front of an <code>sap.ui.unified.Shell</code> control.  |
| sap.ui.unified.SplitContainer           | Provides a main content and a secondary content area.   |
| sap.ui.ux3.ExactArea                    | Consists of two sections: a toolbar and a content area where arbitrary controls can be added.   |
| sap.ui.ux3.FeedChunk                    | Unit that is embedded, standalone or multiple, into a <code>Feed</code> control.  |
| sap.ui.ux3.Feeder                       | Lean common feed, or a comment feed, with a text commit function.   |
| sap.uxap.ObjectPageHeader               | Static part of an <code>Object</code> page header.  |

| SAPUI5 Control Name              | Description                            |
|----------------------------------|--|
| sap.uxap.ObjectPageHeaderContent | Dynamic part of an Object page header. |

## 11.1.7.4 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 \[page 238\]](#)

[Create an OData Model File \[page 239\]](#)

[Create a Project for Your New Application \[page 240\]](#)

[Add Controls to Your New Application \[page 241\]](#)

### 11.1.7.4.1 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 \[page 239\]](#).
2. You have created a new project from a template according to the instructions in the topic [Create a Project for Your New Application \[page 240\]](#).

### Related Information

[Create an OData Model File \[page 239\]](#)

[Create a Project for Your New Application \[page 240\]](#)

[Add Controls to Your New Application \[page 241\]](#)

## 11.1.7.4.1.1 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="DivisionName" 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>
```

- Save the file to your computer with the file name `SalesOrderService_metadata.xml`.

## 11.1.7.4.1.2 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

- Open SAP Web IDE in the Google Chrome browser.
  - 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.
- In the *Template Selection* wizard step, select the *SAP Fiori Master Detail Application* tile and then choose the *Next* button.
  - In the *Basic Information* wizard step, enter a project name. Choose the *Next* button.

#### Note

The project name must start with a letter or an underscore and may contain alphanumeric characters, periods, and underscores. It may not end with a period.

- In the *Data Connection* wizard step, select the *File System* source. Then choose the *Browse* button and navigate to the file `SalesOrderService_metadata.xml` that you created in the topic [Create an OData Model File \[page 239\]](#). Choose the *Next* button.
- In the *Template Customization* wizard step, enter or select the mapping data in the fields and dropdown lists according to the tables below.

#### *Application Settings Section*

| Field or Dropdown List   | Value                                  |
|--------------------------|--|
| <i>Title</i>             | <code>Sales Orders</code>              |
| <i>Project Namespace</i> | <code>&lt;project_namespace&gt;</code> |

| Field or Dropdown List   | Value  |
|--------------------------|--|
|                          | <p><b>i Note</b></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>            | <b>Sales Order</b>         |
| <i>Detail Text</i>      | <b>Sales Order Details</b> |
| <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.

## 11.1.7.4.2 Add Controls to Your New Application

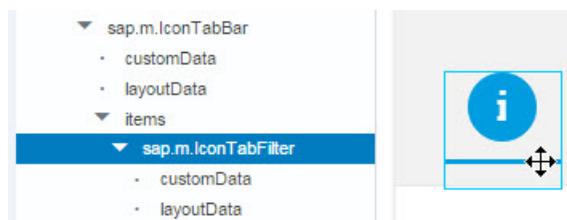
Steps for adding controls to your new sales order tracking application.

### 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 \[page 240\]](#), 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:

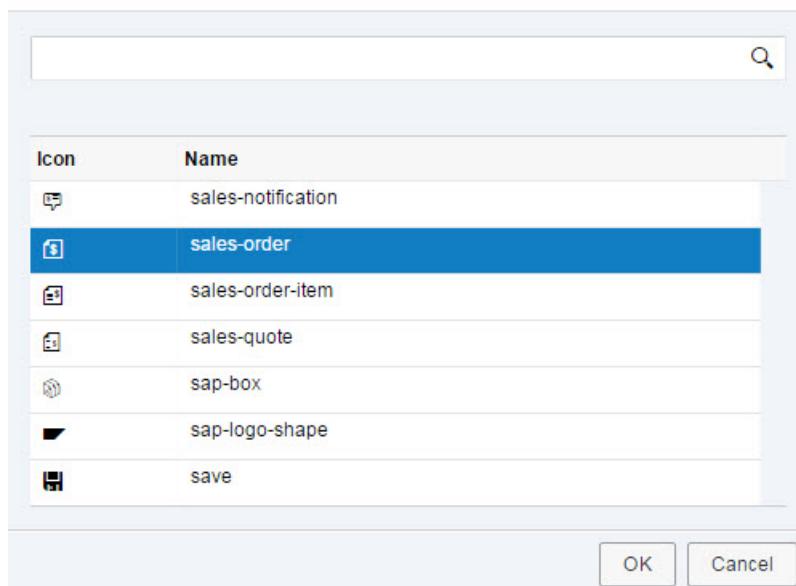
- a. On the canvas, select the first *Icon Tab Filter*  control on the left side of the *Icon Tab Bar* control.

In the *Outline* tab on the left side of the canvas, *sap.m.IconTabFilter* is selected.

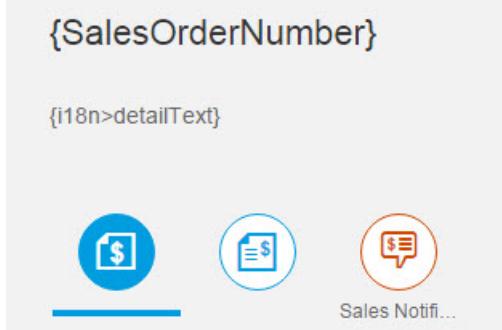


- b. In the *Properties* pane to the right of the canvas, in the *Icon* field, choose the *Select Icon* button to open the *Select Icon* dialog box. Scroll down and select the *sales-order* icon and then choose *OK*.

Select Icon



- 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:
- 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.
  - From the *Controls* tab, drag the *Icon Tab Filter* control to the canvas and drop it on the *Icon Tab Bar* control.
  - Change its icon to the *sales-notification* icon in the same way as you changed the icons in step 3.
  - In the *Properties* pane, in the *Count* field, clear the value by deleting it.
  - 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 2:           | <input type="text"/> |  |

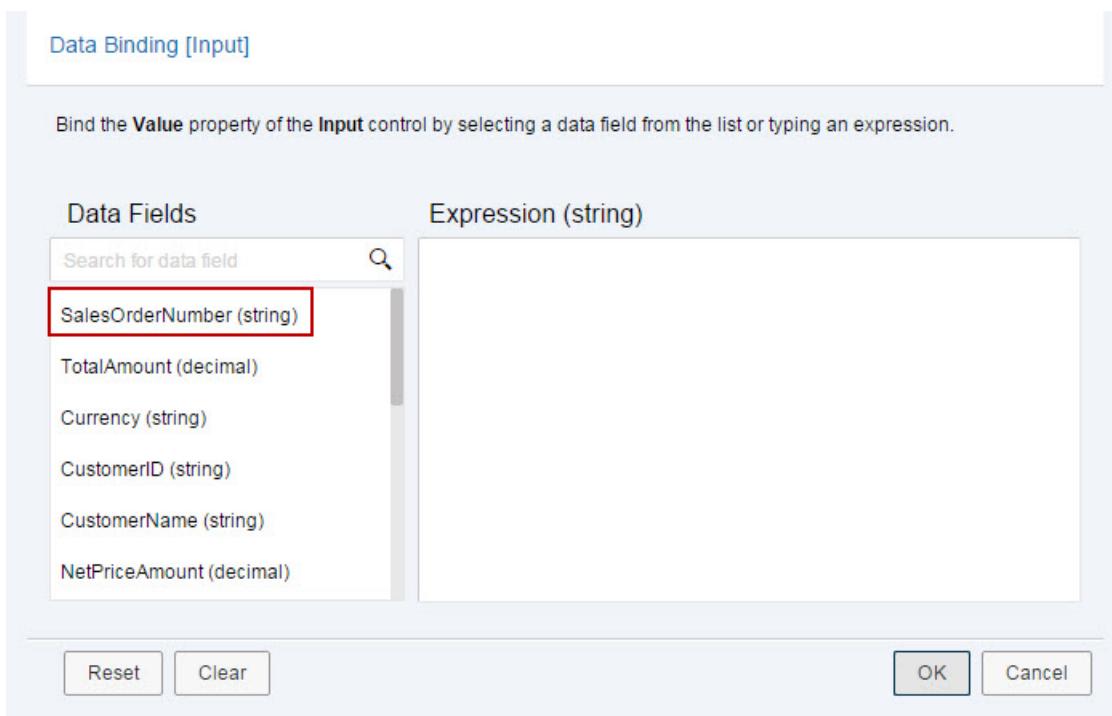
- g. In the canvas, in the *Simple Form* control, select the label that you just created and then in the properties pane to the right of the canvas, change its text to **Order Date**.
- h. Change the text of the *Label 2* control of the third input field in the same way to **Status**.

**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 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.

The screenshot shows a Fiori application interface. On the left, a sidebar titled "Sales Orders" lists six sales orders with their numbers and amounts. On the right, the main area is titled "Sales Order" and displays detailed information for the first order, including its number (SalesOrderNumber 1), amount (623.16), currency (Currency 1), and a "Sales Order Details" section. Below this are three circular icons: a blue one with a dollar sign, a blue one with a document, and an orange one with a speech bubble. A "Sales Notifi..." label is partially visible. At the bottom, there's a "My Sales Notifications" section with fields for "Sales Order Number" (containing "SalesOrderNumber 1") and "Order Date". A gear icon is at the bottom left, and a refresh icon is at the bottom right.

## 11.1.8 Annotation Modeler

The annotation modeler provides an intuitive user interface for annotating OData services. It enables you to build and enhance the user interface of model-based applications in the cloud by adding and editing UI annotations of the local annotation file.

You can also override annotations from other sources, such as metadata, by editing them locally in your SAP Web IDE project.

To run and use annotation modeler with your annotation file, some prerequisites need to be fulfilled. See [Prerequisites for Working with Annotation Modeler \[page 250\]](#).

### 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 \[page 66\]](#).

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 \[page 276\]](#).

For more information about annotation modeler, watch the following video:

[Getting started with the SAP Web IDE annotation modeler](#) 

## Related Information

[Architecture \[page 246\]](#)

[User Interface \[page 248\]](#)

[Prerequisites for Working with Annotation Modeler \[page 250\]](#)

[Adding Local Annotation Files to the Project \[page 252\]](#)

[Working with Annotation Modeler \[page 253\]](#)

## 11.1.8.1 Architecture

### Infrastructure

Annotation modeler is embedded in the SAP Web IDE infrastructure and can be used directly in SAP Web IDE. Simply enable the Annotation Modeler feature, see [Enable Additional Features \[page 392\]](#).

### 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](#)

### 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.

## 11.1.8.2 User Interface

| Node                    | Edit Qualifier                         | Key Information  | Expression Type | Value               | Actions                                   |
|-------------------------|--|--|-----------------|---------------------|---|
| Entity Types            |  |  |                 |                     |   |
| SEPMRA_C_PD_ProductType |  |  |                 |                     |   |
| Local Annotations       |  | Source: localAnnotations                                 |                 |                     | + <span style="float: right;">Edit</span> |
| UI.Facets               | <span style="font-size: 2em;">✎</span> |  |                 |                     | + <span style="float: right;">Edit</span> |
| UI.CollectionFacet      |  | ID: GeneralInformation; Label: General Information       |                 |                     | + <span style="float: right;">Edit</span> |
| ID                      |  |  | String          | GeneralInformation  | + <span style="float: right;">Edit</span> |
| Label                   |  |  | String (I18n)   | General Information | + <span style="float: right;">Edit</span> |
| *Facets                 |  |  |                 |                     | + <span style="float: right;">Edit</span> |
| UI ReferenceFacet       |  | Label: General Information; Target: (@UI Identificati... |                 |                     | + <span style="float: right;">Edit</span> |
| UI.ReferenceFacet       |  | Label: Second Facet; Target: (to_ProductSalesData...     |                 |                     | + <span style="float: right;">Edit</span> |
| External Annotations    |  | Source: SEPMRA_PROD_MAN_ANNO_MDL                         |                 |                     |   |
| External Annotations    |  | Source: mainService                                      |                 |                     |   |
| Properties              |  |  |                 |                     |   |
| Product                 |  |  |                 |                     |   |
| Local Annotations       |  | Source: localAnnotations                                 |                 |                     | + <span style="float: right;">Edit</span> |
| External Annotations    |  | Source: SEPMRA_PROD_MAN_ANNO_MDL                         |                 |                     |   |
| Price                   |  |  |                 |                     |   |
| Currency                |  |  |                 |                     |   |

Annotation Modeler    Code Editor

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 \[page 266\]](#).

- You can edit *Local Annotations* from the currently open file in annotation modeler. See [Edit Annotations \[page 255\]](#). 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 \[page 262\]](#).
- 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 \[page 263\]](#).

#### 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 \[page 268\]](#).

- 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 \[page 264\]](#).
- You can concatenate several values for all terms containing the [Value](#) property. See [Concatenate Values \[page 265\]](#).
- You can insert parameters in any part of a URL. See [Use Parameters in URLs \[page 265\]](#).

#### 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 \[page 257\]](#).

#### 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 \[page 266\]](#).

## ➔ 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 \[page 34\]](#).

### 11.1.8.3 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 \[page 269\]](#).

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 \[page 66\]](#).

### 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 \[page 253\]](#) and [Import Existing Annotation Files \[page 252\]](#).

## 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 \[page 253\]](#)

### 11.1.8.4 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 \[page 252\]](#).
- Create a new annotation file. See [Create New Annotation Files \[page 253\]](#).

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 \[page 252\]](#)

[Create New Annotation Files \[page 253\]](#)

### 11.1.8.4.1 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 > Annotation File](#) in the context menu or click the same path in the [File](#) menu.

3. Select a system that contains annotations, select a matching OData service, and select an annotation file.

The annotation file is imported to the folder and is available for editing if the prerequisites are met. See [Prerequisites for Working with Annotation Modeler \[page 250\]](#).

**i Note**

- If needed, you can rename the annotation file.
- You cannot import a file that contains no annotations.

### 11.1.8.4.2 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.

### 11.1.8.5 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.

- Check whether the [Prerequisites for Working with Annotation Modeler \[page 250\]](#) have been met.

 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 \[page 254\]](#).
- You can preview the annotated data if your project supports running the application from the workspace. See [Run Applications from the Workspace \[page 312\]](#).

## Related Information

[Open Local Annotation Files in Annotation Modeler \[page 254\]](#)

[Edit Annotations \[page 255\]](#)

[Validation of the Annotation File \[page 267\]](#)

### 11.1.8.5.1 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 \[page 34\]](#).

### Open Annotation Modeler from the Context Menu of the Annotation File

- Right-click the local annotation file of your project.
- Choose  *Open With Annotation Modeler*.

## 11.1.8.5.2 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 \[page 266\]](#).

#### 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 \[page 256\]](#).  
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.

#### 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.

#### Note

If an annotation term appears in the local annotation file more than once, you have to enter a qualifier or an ID (for facets) for all but one of the annotation terms so that they are uniquely identified.

- Make sure that the entered **ID** is unique among all annotations.
  - Make sure that the **qualifier** is unique for the annotation term.
5. Use the *Actions* column to add subnodes and to delete nodes.
  6. Save your entries.
    - When you save, annotation modeler validates the entire annotation file.
    - Run your application to see the results with the preview function of SAP Web IDE.

## Related Information

[Add Annotation Terms, Items, and Records to the Local Annotation File \[page 257\]](#)

[Change the Order of Annotations and Collection Items \[page 263\]](#)

[Create Internationalized Labels \[page 264\]](#)

[Concatenate Values \[page 265\]](#)

[Use Parameters in URLs \[page 265\]](#)

[Edit External Annotations \[page 266\]](#)

### 11.1.8.5.2.1 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 \[page 246\]](#), section *Scope*.

#### Procedure

1. Choose the *Select Targets* button.

The top-level target types are displayed in alphabetical order. The entity types for targets that are already included in the annotation structure are automatically selected.

#### Note

The list of targets has a hierarchical structure similar to that of the metadata. Not all nodes are visible at all times.

- To view and select the entity type properties, expand the following nodes: `Entity Types/<entity type name>/Properties`.

- To view and select the entity sets, expand the following nodes: Entity Containers/<entity container name>/Entity Sets.
  - To view and select the function imports, expand the following nodes: Entity Containers/<entity container name>/Function Imports.
  - To view and select the function import parameters, expand the following nodes: Entity Containers/<entity container name>/Function Imports/<function import name>/Parameter.
2. You can search the tree by entering a search term.

The tree is filtered and expanded to show every target matching the search. If you clear the search term, the original state of the tree is recovered.

**i Note**

You can search again to find additional targets to select. All your previous selections remain and become visible as soon as you clear the search.

3. Select or deselect the respective checkboxes.
4. Choose *OK*.

## Results

The newly selected targets are displayed in the annotation structure. Once you have defined at least one annotation for each target, they are added to the annotation file as well.

### 11.1.8.5.2.2 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 \[page 256\]](#).
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 \[page 258\]](#)

### 11.1.8.5.2.2.1 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.

#### Note

In this example, the field group that includes the technical data already exists in the backend. We will create a new [UI.ReferenceFacet](#) and reference to the target [UI.FieldGroup](#).

The screenshot shows an SAP Fiori object page for a product. At the top, there's a header with the SAP logo and the product name "10'' Portable DVD player HT-2001". Below the header, there's a thumbnail image and some basic product details: Category: TV, Video & HiFi (TV, Video & HiFi), Sub-Category: Portable Players, and Supplier: Panorama Studios (100000004). To the right, there's a "Product Description" section with the text "10'' LCD Screen, storage battery holds up to 8 hours".

The page has three main tabs at the bottom: PRODUCT INFORMATION, SALES DATA, and PRICE. The PRICE tab is currently selected, showing a large green box around the content area.

In the PRODUCT INFORMATION section, there's a "General Information" field group containing the following data:

- Product: 10'' Portable DVD player (HT-2001)
- Product Unit Price: 449.99 USD
- Sub-Category: Portable Players

The SALES DATA section is highlighted with a green box. It includes a "Delivery Year / Delivery Month" dropdown set to "20K" and a bar chart showing sales volume. The x-axis represents months, and the y-axis represents sales volume, with a scale from 0 to 15K. The bars are colored in various shades of blue, green, yellow, and pink.

### 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 \[page 254\]](#).

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 \[page 256\]](#).
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.

| Node                    | Edit Qualifier    | Key Information                              | Expression Type | Value   | Actions                             |
|-------------------------|-------------------|--|-----------------|---|-------------------------------------|
| Entity Types            |                   |  |                 |   |                                     |
| SEPMRA_C_PD_ProductType |                   |  |                 |   |                                     |
| Local Annotations       |                   | Source: localAnnotations                     |                 |   | <a href="#">+</a>                   |
| UI.Facets               | <a href="#">+</a> |  |                 |   | <a href="#">+</a> <a href="#">-</a> |
| UI.CollectionFacet      |                   | ID: ProductInfo; Label: Product Information  |                 |   | <a href="#">+</a> <a href="#">-</a> |
| ID                      |                   |  | String          | ProductInfo   | <a href="#">+</a> <a href="#">-</a> |
| Label                   |                   |  | String (i18n)   | Product Information                                       | <a href="#">+</a> <a href="#">-</a> |
| *Facets                 |                   |  |                 |   | <a href="#">+</a>                   |
| UI.ReferenceFacet       |                   | Label: Second Facet; Target: {@UI.Chart#...} |                 |   | <a href="#">+</a> <a href="#">-</a> |
| UI.ReferenceFacet       |                   |  |                 |   | <a href="#">+</a> <a href="#">-</a> |
| *Target                 |                   |  | AnnotationPath  | Navigation <a href="#">+</a> Annotation <a href="#">▼</a> | <a href="#">+</a>                   |
| External Annotations    |                   | Source: SEPMRA_PROD_MAN_ANNO_M...            |                 |   |                                     |
| External Annotations    |                   | Source: mainService                          |                 |   |                                     |

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*.

- o If you leave the *Navigation* field empty, you can reference annotations only from the currently selected OData entity type.
  - o 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 the SAP logo and navigation icons. Below the header, the product name "10'' Portable DVD player" and code "HT-2001" are displayed. To the left is a small thumbnail image. On 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" (which is selected), "SALES DATA", and "PRICE". Under "PRODUCT INFORMATION", 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 these sections is a "Sales Data" section containing a bar chart titled "Delivery Year / Delivery Month". The chart has two y-axis labels: "20K" and "15K". The bars are colored in various shades of blue, green, yellow, and pink. At the bottom left of the page, there's a "Tip" section with the text: "To find out more about how facets annotations are used in applications based on the ObjectPage template, see [Sections](#)".

### 11.1.8.5.2.3 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 \[page 256\]](#).
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.

## 11.1.8.5.2.4 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.

## 11.1.8.5.2.5 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

## 11.1.8.5.2.6 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](#)

## 11.1.8.5.2.7 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.

## 11.1.8.5.2.8 Edit External Annotations

Annotations defined in the backend or in other external sources cannot be edited directly but can be overridden by cloning them to local annotations. When you clone an annotation from the backend to the local file, the clone is inserted in the currently open annotation file where you can edit it.

## Prerequisites

Verify that the `manifest.json` file contains the `dataSources` object with an array of one or more *ODataAnnotation* types. See [Prerequisites for Working with Annotation Modeler \[page 250\]](#).

### 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 \[page 269\]](#).

## 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 \[page 256\]](#).
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 \[page 255\]](#).

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)*.

#### 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.

#### 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.

### 11.1.8.5.3 Validation of the Annotation File

Annotation modeler checks your `annotation.xml` file for inconsistencies in vocabulary specifications and other issues that can cause an unstable UI.

#### 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 \[page 268\]](#)

### 11.1.8.5.3.1 Warnings and Errors in Annotation Modeler

Annotation modeler validates the local annotation file based on the vocabulary definitions and OData standards. Error and warning messages notify you about incompatibilities during the development process and support you in resolving them.

- When you save a project, the local annotation file is validated.
- When you leave a field, the field is validated.
- If the validation process discovers an error, such as an empty value for mandatory property or wrong input format, error and warning statuses are updated: the 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.

### 11.1.8.5.3.2 Check Error Descriptions in the Problems View

In the *Problems* view, you can check your `annotations.xml` file for issues resulting from validation checks run by the annotation modeler. This includes inconsistencies regarding vocabulary specifications and other issues that can cause an unstable UI.

#### Procedure

1. To check the `annotations.xml` file for issues specific to 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 \[page 196\]](#).

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.

## 11.1.8.6 Troubleshooting

| Error Message   | Issue Description and Solution   |
|---|--|
| <p><b>OData metadata cannot be loaded from destination &lt;destination name&gt;. This error can appear for several reasons:</b></p> <ul style="list-style-type: none"><li><b>The login data is incorrect: provide the correct credentials.</b></li><li><b>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.</b></li><li><b>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.</b></li></ul> | <p>The annotation file cannot be loaded because the OData metadata file cannot be loaded from the back end.</p> <p><b>i Note</b></p> <p>Annotation modeler loads metadata based on the URI entry defined in the <code>manifest.json</code> file. This is usually a URI pointing to a back-end destination. The <code>localUri</code> setting is not used, which prevents inconsistency between back-end metadata and its local copy.</p> <ul style="list-style-type: none"><li>Make sure that you have entered the correct credential for accessing the back-end system.</li><li>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.</li><li>Check the relevant entry of the <code>dataSources</code> setting of type OData in the <code>manifest.json</code> file (app descriptor).<br/>Make sure that the OData Service is correct and that there is an entry in the <code>neo-app.json</code> file of the same project that represents at least a part of the string of the URI of your data source.</li><li>Check the destination that is referred to by the <code>neo-app.json</code> entry. The name of the destination must be the same as defined in SAP Cloud Platform Cockpit. Also make sure that the <code>WebIDEUsage</code> property contains either <code>odata_abap</code> or <code>odata_xs</code>, but not both.<br/>You can also use destinations of <code>WebIDEUsage odata_gen</code>. If your destination addresses a dedicated service with its full URL, you must specify the additional <code>WebIDEAdditionalData</code> property with the <code>full_url</code> value.</li></ul> <p>For more information, see <a href="#">Prerequisites for Working with Annotation Modeler [page 250]</a> and <a href="#">Connect to ABAP Systems [page 28]</a>.</p> |

| Error Message  | Issue Description and Solution  |
|--|---|
| <p><b>The file &lt;file name&gt; cannot be edited, because it is not registered in the manifest.json file (app descriptor).</b></p>              | <ul style="list-style-type: none"> <li> <b>The local annotation file cannot be opened.</b><br/>           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, <a href="#">New Annotation File Settings</a> or <a href="#">Import Annotation File</a>). Please define the annotation file for an existing data source in the <code>manifest.json</code> file. See <a href="#">Modifying the Application Descriptor Configuration [page 78]</a> for further instructions.         </li> <li> <b>The external annotation file cannot be opened.</b><br/>           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.<br/>           To modify external annotations, clone them to your local annotation file to edit them. For more information, see <a href="#">Edit External Annotations [page 266]</a>.<br/>           If your project does not have a local annotation file, create a new local annotation file. For more information, see <a href="#">Create New Annotation Files [page 253]</a>.         </li> </ul> |
| <p><b>The annotation xml file cannot be loaded because it contains xml format errors. Open the Problems view to view and fix the errors.</b></p> | <p>The xml file is invalid due to xml format errors and cannot be opened in annotation modeler. Open the <a href="#">Problems</a> view to view and fix the errors. Then, reopen the annotation file in annotation modeler.</p> <p>For more information, see <a href="#">Warnings and Errors in Annotation Modeler [page 268]</a>.</p>   |
| <p><b>Invalid file: DataServices definition is missing.</b></p>  | <p>Check that the annotation xml file includes an / <code>edmx:DataServices/Schema</code> node.<br/>           You can create a valid file by creating an annotation file.<br/>           For more information, see <a href="#">Create New Annotation Files [page 253]</a>.</p>   |
| <p><b>There is no OData data source defined in the manifest.json. Please define an OData data source.</b></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><b>OData metadata cannot be loaded from destination &lt;destination name&gt;. The destination is not correctly maintained at the SAP Cloud Platform.</b></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> |

## 11.1.9 Storyboard

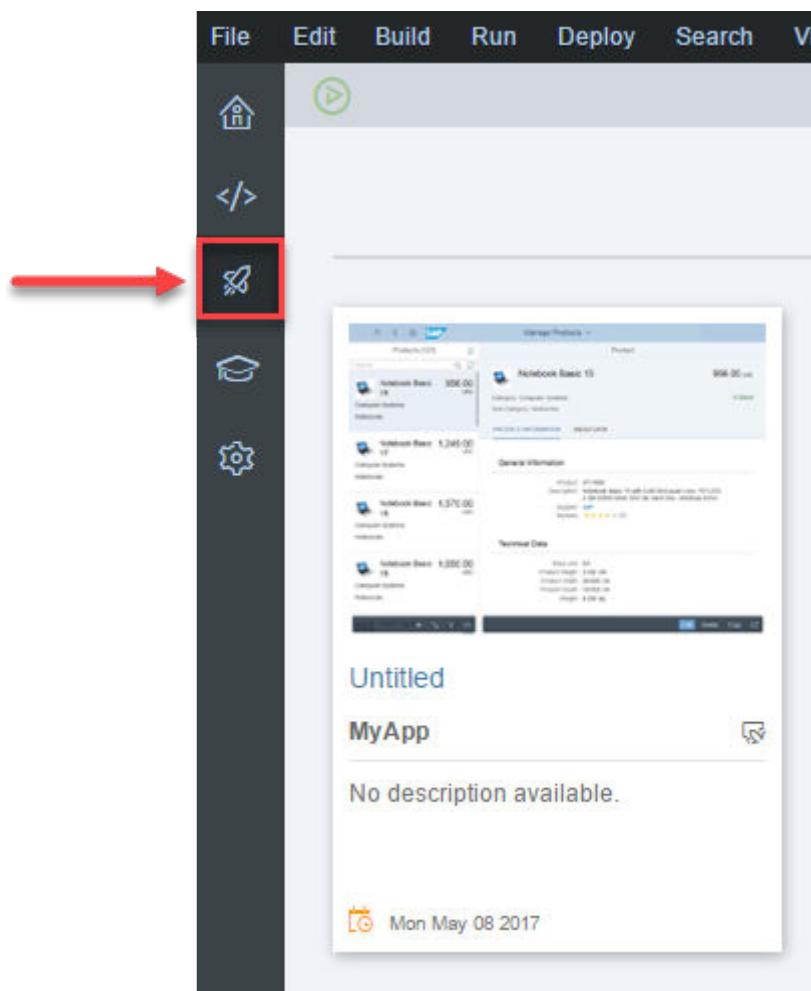
The storyboard provides a visual representation of the application's UI including its pages (views), navigations, and the services and entities that it uses.

### Prerequisites

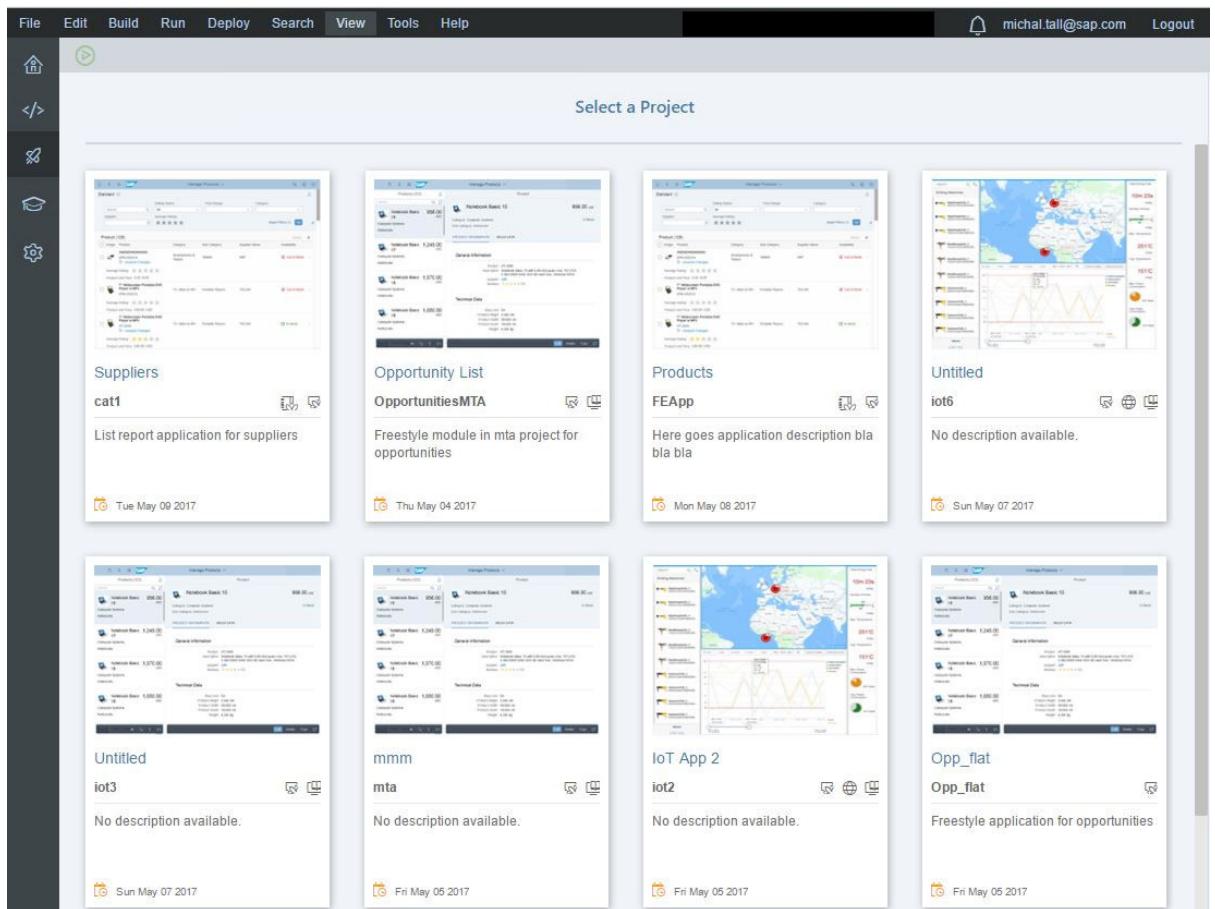
- You have enabled the *Storyboard* perspective in the *Preferences* perspective, on the *Features* tab.
- You have created a freestyle or SAP Fiori Elements project.

### Procedure

1. Open the *Storyboard* perspective from the left sidebar.



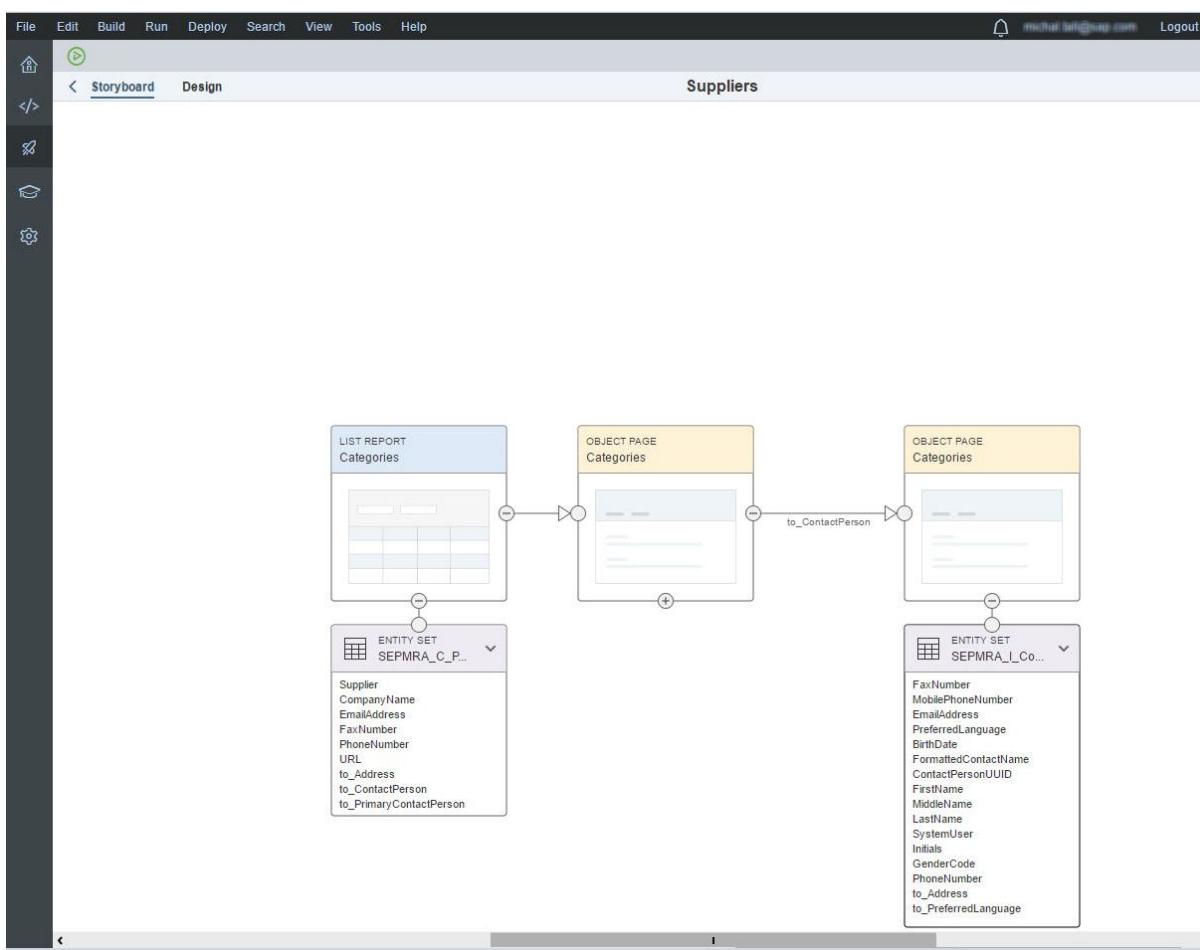
A preview of all the projects supported by the storyboard in your workspace are displayed.



There are two types of project, freestyle and SAP Fiori Elements.

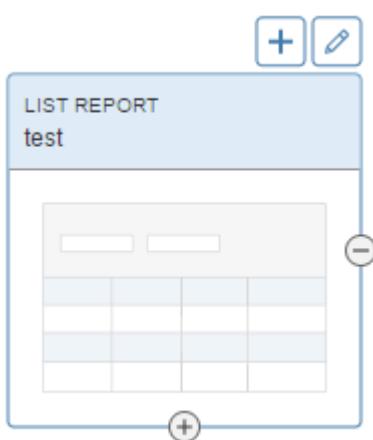
2. Select the desired project.

The *Storyboard* tab opens showing all the views in the project with the navigations between them each project and the respective entity sets.

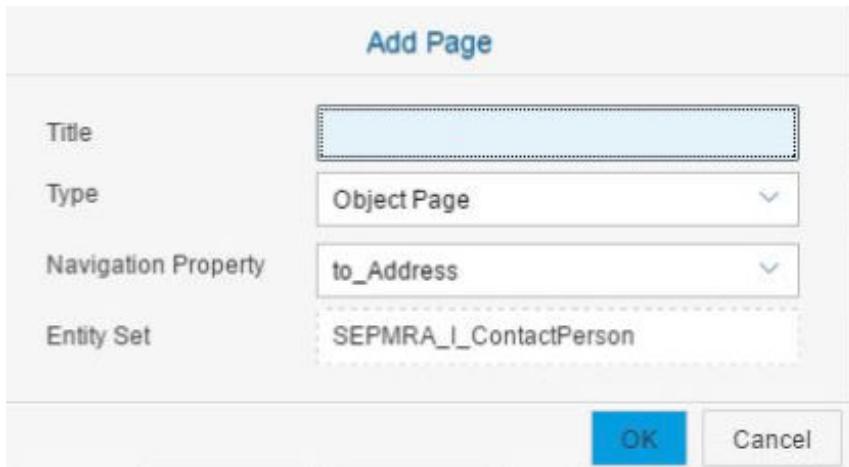


3. For SAP Fiori Elements projects, you can add new views for Object Page and List Report.

- a. Click to add a new view.



- b. Fill in the relevant fields and click **OK**.



4. To edit a view, click  or go to the *Design* tab.
  - For an SAP Fiori Elements view, the UI Adaptation editor opens. For more information, see [UI Adaptation Editor \[page 276\]](#).
  - For a view from a freestyle application, the layout editor opens. For more information, see [Layout Editor \[page 207\]](#).

### 11.1.9.1 Add View

You can add a new view in the *Storyboard* perspective.

#### Prerequisites

- You have enabled the *Storyboard* perspective in the *Preferences* perspective, on the *Features* tab.
- You have created a freestyle project.

#### Procedure

1. In the Storyboard perspective, choose *New View*.
2. In the *Template Customization* wizard, in the *View Name* field, enter a name for the new view.
3. Choose *Next*, then choose *Finish*.

#### Results

The new view appears in the *Storyboard* perspective, and in the *Development* perspective, a view and controller are created in your Workspace and the manifest file is updated accordingly.

## 11.1.9.2 Delete View

You can delete a view from the *Storyboard* perspective.

### Prerequisites

- You have created one or more views in your project.

### Procedure

1. In the *Storyboard* perspective, select the view you want to delete.
2. Press `Delete` on your keyboard and then choose *Yes* in the *Confirmation Needed* dialog box.

### Results

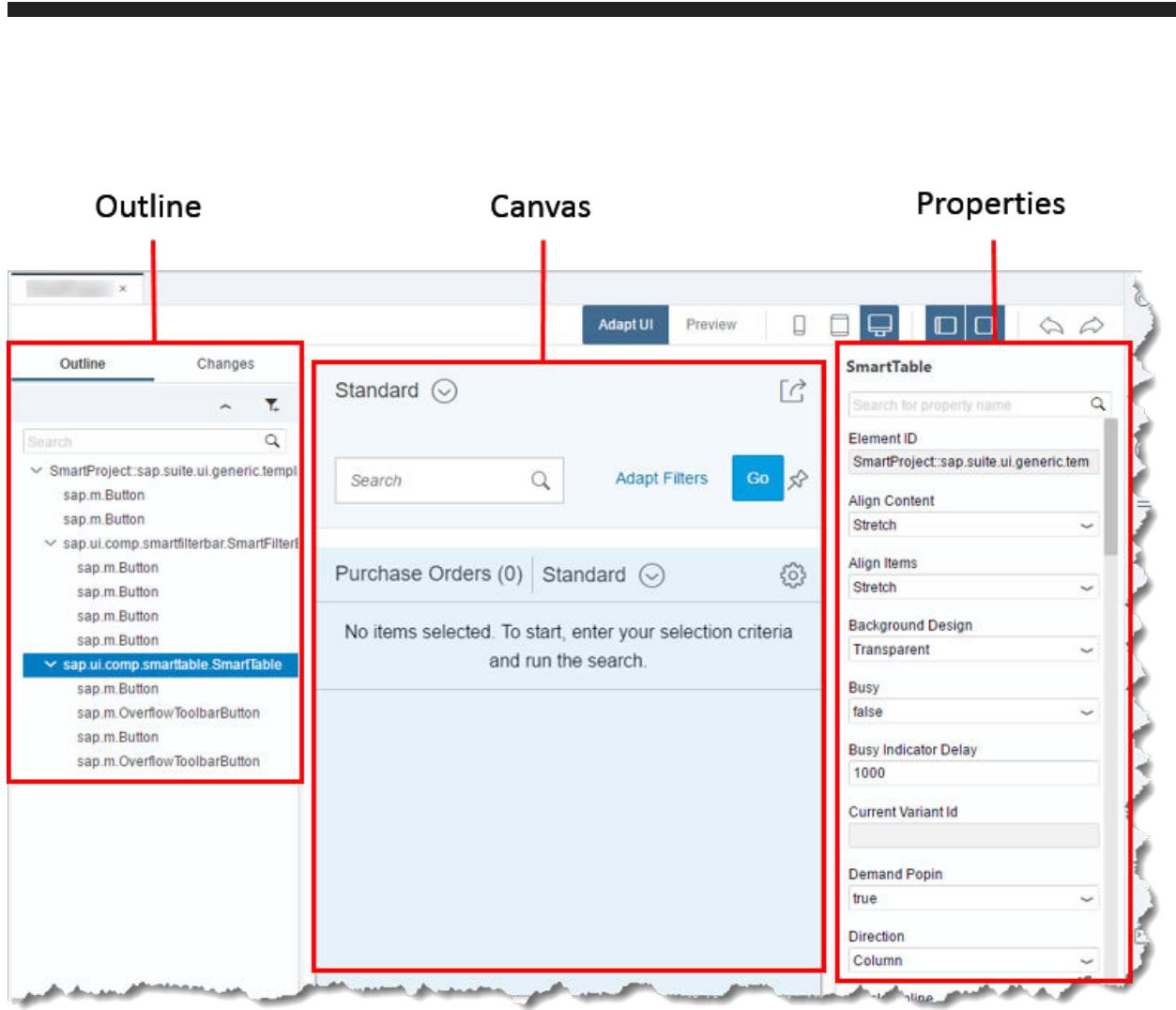
The view is removed from the *Storyboard* perspective and the view and its controller are deleted from the Workspace and the manifest file is updated accordingly.

## 11.1.10 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

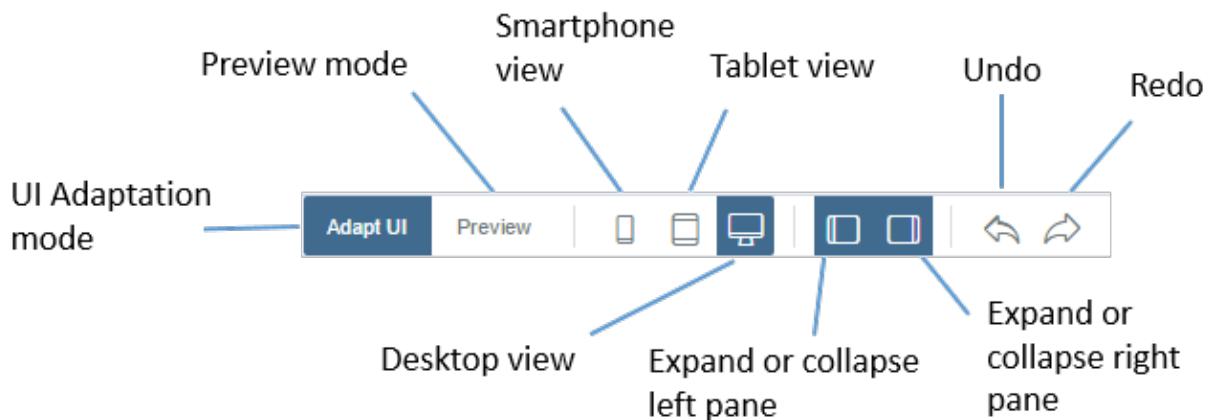
If you switch between modes, your changes are saved and copied to the workspace.

- 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 \[page 278\]](#)

[UI Adaptation Options \[page 280\]](#)

[Developing Apps with SAP Fiori Elements](#)

### 11.1.10.1 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 a Template \[page 51\]](#).
- 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

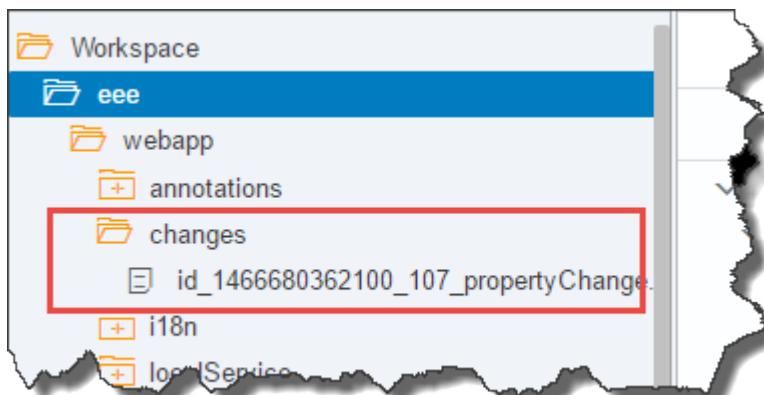
UI Adaptation can only be used for smart controls with stable IDs and for making property changes to non-smart controls.

- 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 \[page 329\]](#).

## Related Information

[Configure Run Configurations for the UI Adaptation Editor \[page 75\]](#)

### 11.1.10.2 UI Adaptation Options

List of possible changes you can make to your application at runtime using the UI adaptation mode.

|                                       |   |
|---------------------------------------|---|
| <b>Change properties</b>              | <ol style="list-style-type: none"><li>1. Select the UI element you want to change.</li><li>2. Change the element's properties as needed.</li></ol> <div style="background-color: #f2e0c7; padding: 5px;"><b>i Note</b><p>Not all properties are subject to change.</p></div>  |
| <b>Add new fields</b>                 | <ol style="list-style-type: none"><li>1. Hover over or select a group or a field and choose <i>Add Field</i> from the context menu (right-click).</li><li>2. Select the fields from the list of available fields that you want to add to the UI.<br/>You can also search for field labels and tooltips, or sort the fields in alphabetical order.</li><li>3. To apply your adaptations, choose <i>OK</i>.</li></ol>   |
| <b>Add new groups</b>                 | <ol style="list-style-type: none"><li>1. Hover over or select a group or select the form it's contained in and choose <i>Add Group</i> from the context menu (right-click).<br/>Its default title is <i>New Group</i>, but you can rename it to whatever you want.</li><li>2. To apply your adaptations, press <code>[ENTER]</code> or <code>[ESC]</code>.</li></ol>  |
| <b>Add sections to an object page</b> | <ol style="list-style-type: none"><li>1. Hover over or select a section and choose <i>Add Section</i> from the context menu (right-click).</li><li>2. Select the sections from the list of available sections that you want to add to the UI.<br/>You can also search for sections or sort them in alphabetical order.</li><li>3. To apply your adaptations, choose <i>OK</i>.</li></ol> <div style="background-color: #f2e0c7; padding: 5px;"><b>i Note</b><p>If all available sections are placed on the object page, you cannot use this function and it's grayed out in the context menu.</p></div> |

|  |  |
|--|--|
| <b>Rename fields and groups</b>                      | <ol style="list-style-type: none"> <li>1. Double-click a field or group. You can also hover over or select it and choose <i>Rename Field</i> or <i>Rename Group</i> from the context menu (right-click).</li> <li>2. Rename the field label or group title.</li> <li>3. To apply your adaptations, press <b>[ENTER]</b>; to quit, press <b>[ESC]</b>.</li> </ol>   |
| <b>Move fields, groups, and object page sections</b> | <ol style="list-style-type: none"> <li>1. Drag a field, group, or section.</li> <li>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.</li> </ol>  |
| <b>Cut and paste fields and groups</b>               | <ol style="list-style-type: none"> <li>1. Hover over or select a field or group and choose <i>Cut</i> 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.</li> <li>2. To paste a cut field, hover over or select a highlighted group or a field in a highlighted group and choose <i>Paste</i> from the context menu. To paste a cut group, hover over or select a group in the highlighted forms and choose <i>Paste</i> from the context menu.</li> </ol> <p><b>i Note</b></p> <p>To get rid of the highlighting and exit pasting, press <b>[ESC]</b>.</p> |

## 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.

## 11.1.11 Using Source Control (Git)

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 this are several circular icons representing Git operations: Pull, Fetch, Rebase, Merge, Show Stash, Reset, and Fetch from Gerrit. A sidebar on the right contains icons for search, history, and other tools.

The main area is titled "Commit" and features a "Status table" with the following data:

| Status | Name                | Stage                    | Discard |
|--------|---------------------|--------------------------|---------|
| M      | webapp/Component.js | <input type="checkbox"/> |         |
| M      | webapp/index.html   | <input type="checkbox"/> |         |

Below the status table is a "Commit Description" field with the placeholder "Insert commit description". At the bottom, there is a note: "A Change ID for Gerrit will not be added" followed by a checkbox labeled "Amend Changes". There are three buttons at the bottom right: "Commit", "Push", and "Stash".

To open the *Git* pane:

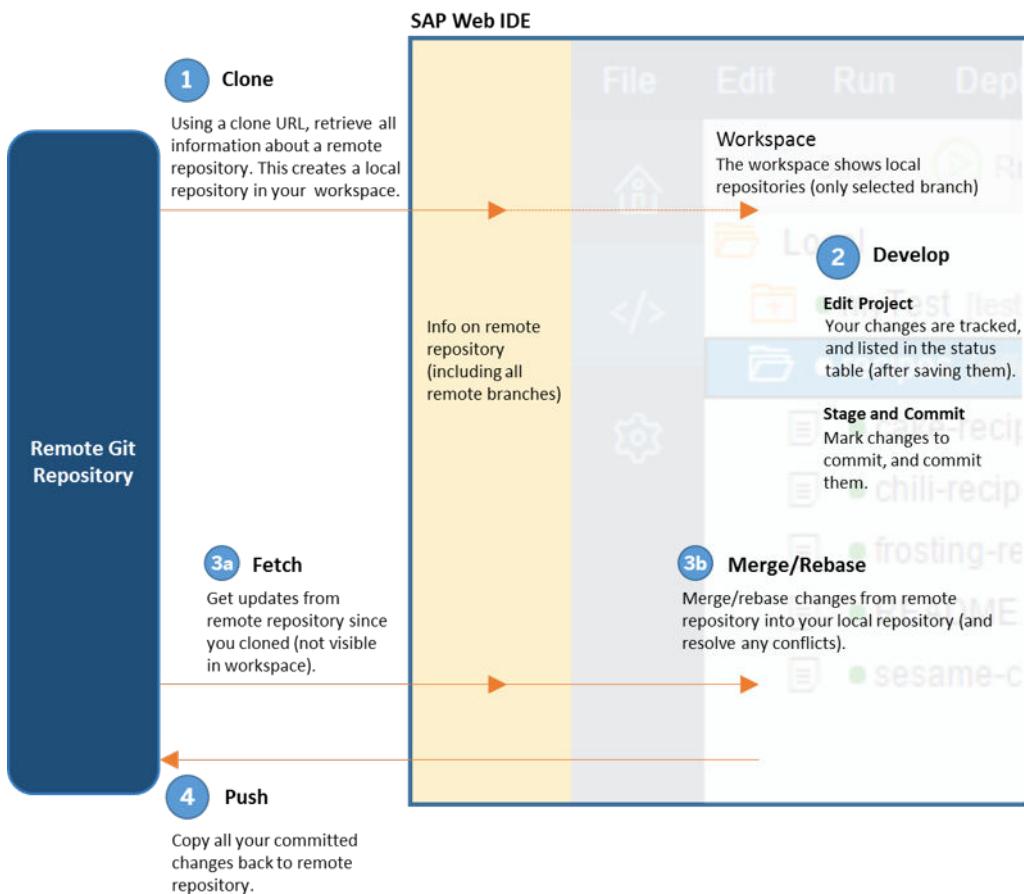
1. From the workspace, select a Git repository
  2. From the right sidebar, choose (Git pane).
- **Git History Pane:** Lets you view the commits for different branches, as well as perform some commands on those commits.
- To open the *Git History* pane:

1. From the workspace, select a Git repository
  2. From the right sidebar, choose  (Git History pane).
- **Git 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 \[page 86\]](#).

## 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 \[page 287\]](#)

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 \[page 289\]](#)

You can manage the connectivity to your on-premise Git repository.

## [Clone Repositories \[page 291\]](#)

You can clone an existing Git repository into your workspace.

## [Initialize a Local Git Repository \[page 292\]](#)

You can initialize a local repository for any project that is not already connected to a Git repository.

## [Set a Remote Repository \[page 292\]](#)

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 \[page 293\]](#)

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 \[page 294\]](#)

Rebasing enables you to take all the committed changes from one branch and incorporate them into a different branch.

## [Merge Changes \[page 295\]](#)

You can incorporate all the changes from one branch into another in a single commit.

## [Pull Changes \[page 296\]](#)

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 \[page 296\]](#)

The staging table shows changed files, and lets you select files to stage.

## [Commit Changes \[page 300\]](#)

You can commit changes to the repository locally.

## [Push Changes \[page 301\]](#)

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.

## [Multiple Local Branches \[page 302\]](#)

From the Git pane, you can check out a local branch, add a new local branch, and remove a local branch.

## [Create Remote Branches \[page 304\]](#)

You can create a new branch in the remote repository.

## [Git History \[page 305\]](#)

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 \[page 308\]](#)

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.

## Related Information

### [Git Troubleshooting \[page 426\]](#)

## 11.1.11.1 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*.

### 11.1.11.1.1 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              |

## 11.1.11.2 Connect to your Corporate Git System

You can manage the connectivity to your on-premise Git repository.

### Context

#### 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>  |

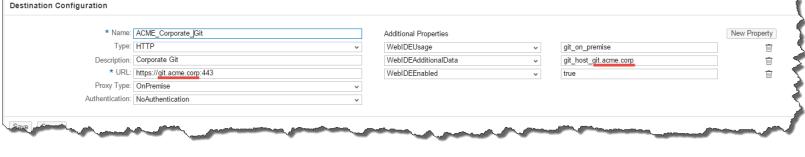
For the system you just added, specify the resources to enable, using the following settings:

| 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).

4. Define your corporate Git destination. For more information, see [Connect to ABAP Systems \[page 28\]](#).

- a. In the SAP Cloud Platform cockpit, select [Connectivity](#) [Destinations](#).
- b. Select [New Destination](#).
- c. In the *Destination Configuration* section, set the *Proxy Type* to *OnPremise*.
- d. In the *Additional Properties* section, configure the following:

| Property             | Value  |
|----------------------|--|
| WebIDEEnabled        | true   |
| WebIDEUsage          | git_on_premise   |
| WebIDEAdditionalData | git_host_<Git host name><br><br>The Git host name must be entered as it appears in the URL. For example, if the URL is <code>https://git.acme.corp:443</code> , the host name would be <code>git.acme.corp</code> .<br><br>So you would enter in the field: <code>git_host_git.acme.corp</code><br><br> |

### Note

When you define the cloud connector, there are two types of hosts: Internal and Virtual. Make sure to use the internal host.

## Next Steps

Test your new corporate Git system by cloning a repository from your corporate Git system (see [Clone Repositories \[page 291\]](#)) or by initializing a local repository, setting the remote repository, fetching and pushing to the remote repository (see [Initialize a Local Git Repository \[page 292\]](#)).

### 11.1.11.3 Clone Repositories

You can clone an existing Git repository into your workspace.

#### Procedure

1. From the *File* menu, choose ► *Git* ► *Clone Repository* ▾.

**i** Note

You can clone a Git repository only if your Git user settings have been defined. You can do this in the *Git Settings* dialog box from the *Tools* menu. If you have not defined these settings, then before the *Clone Repository* dialog box opens, SAP Web IDE checks if your Git user name and email address exist. If not, SAP Web IDE extracts this information from the identity provider defined in your account and updates Git.

2. In the *URL* field, enter the Git repository URL and press .
3. If your remote Git system works with Gerrit, select *Add configuration for Gerrit*.
4. Choose *Clone*. The cloning starts. When the process is finished, the content of the repository appears in the workspace.

### 11.1.11.3.1 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 ► *Project Settings* ► *Git Repository Configuration* ▾.
2. Configure the Git repository as follows:
  - To create a new entry, choose *Add Entry* and type the relevant values in the *Key* and *Value* fields.

**i** Note

Use the following format for the *Key* entry: <section>. <name>. For example, `user.name`

**i** Note

The *Key* field is mandatory and cannot be duplicated.

- To edit an entry, choose  (Edit).

- To delete an entry, choose  (Delete).
3. Choose  (Save).

## 11.1.11.4 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 \[page 292\]](#)

## 11.1.11.5 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   *File > Git > Initialize Local Repository*  *Git > Set Remote* .
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 \[page 292\]](#)

### 11.1.11.6 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*.

## 11.1.11.7 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 \[page 294\]](#)

## 11.1.11.7.1 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.

## 11.1.11.8 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.

## 11.1.11.9 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.

## 11.1.11.10 Stage Files

The staging table shows changed files, and lets you select files to stage.

### Context

Whenever a file is updated, added, or deleted, it appears in the staging table, but is not staged. After staging, you can commit the staged files.

### Procedure

In the staging table of the *Git Pane*, choose the *Staged* toggle button to the right of the change that you want to stage.

#### Note

To stage all files in the staging table, select the checkbox at the top of the table, and choose *Stage Selected* from the dropdown list.

You can edit a file listed in the staging table by clicking  (More Actions) and choosing *Edit*. This option is enabled for all files, except for deleted.

You can delete a file listed in the staging table by clicking  (More Actions) and choosing *Delete*. This removes the file from the staging table and from the workspace.

## Related Information

[Discard Changes \[page 298\]](#)

### 11.1.11.10.1 Compare Code

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 table in the *Git* pane.

You can also compare two committed versions from the *Git History* pane.

#### Procedure

1. In the *Git Pane*, double-click a file in a staging table row, or click  (More Actions), and choose *Compare*.

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.

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.

#### 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.

## 11.1.11.10.2 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 staging table of the [Git Pane](#), select the checkbox in the row that contains the change that you want to discard, and choose [Discard](#).

All changes that you made to the file are removed.

**i** Note

To discard all files, select the checkbox at the top of the table, and choose [Discard](#). All unstaged files in the staging table are discarded.

## 11.1.11.10.3 Stash Changes

If you made some changes that you are not yet ready to commit, you can stash (store away) and revert them from your working directory, and resume working on them later.

### Context

**i** Note

This feature is not available in SAP Web IDE personal edition.

## Procedure

1. In the *Git Pane*, choose the *Stash* button.
2. Optional: modify the *Description* for the stash. By default, the description is *on <branch> : <creation time stamp>*.

Each time you perform a stash operation, the new stash is saved at the top of the stash list. Meaningful descriptions will help you to locate the stash that you want to apply.

3. Choose *Stash*.

## Results

The changes are stored in a new stash, and reverted from your working directory. The list of changed files is emptied.

## Apply Stashes

### Context

If you have stashed changes, you can apply the stashed changes to the files in your working directory by clicking the *Apply Stash* icon in the Git pane toolbar.

You may need to resolve conflicts between the version of a file in your stash and the version in your working directory.

- If the file in your working directory has been committed and has conflicts with the stashed changes, the conflicts are shown in the file in the working directory.
- If the file in your working directory has not been committed, you may not be able to apply the stash. Either commit or reset the changes to the file in your working directory, and then apply the stash again.

### i Note

It is recommended to stage the newly created files before stashing them. For example, consider the following scenario:

1. Create file 1.
2. Change file 2.
3. Stash your changes. This stashes both new file 1 and the changes to file 2.
4. Change file 2 and commit the changes.
5. Apply the stash.

In this case, the stashed changes to file 2 are applied successfully, though you will get a conflict to resolve. But the new file 1 is not applied and is lost.

You can avoid this issue by staging the new file 1 before stashing, or even committing the new file and not stashing it.

## Procedure

1. Choose *Apply Stash*.
2. From the dropdown list, select the stash that you want to apply.
3. Choose one of the following options:
  - *Apply*: Reapply the stashed changes to your working directory, and keep the stash in the stash list.
  - *Pop*: Reapply the stashed changes to your working directory, and remove the stash from the stash list.
  - *Drop*: Delete the stashed changes so they can no longer be applied to your working directory. The working directory is unaffected.
4. Choose *Continue*.

## 11.1.11.11 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.

## 11.1.11.12 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.

#### Example

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

- You receive notifications about problems with warning severity only.  
You can choose to continue with the push, or you can view and fix the problems before pushing.
- You receive notification about problems with error and warning severities.  
You can view the problems but you cannot continue with the push until the errors are fixed.

## 11.1.11.13 Multiple Local Branches

From the Git pane, you can check out a local branch, add a new local branch, and remove a local branch.

### Checking Out a Local Branch

#### Procedure

In the *Git Pane*, select the desired local branch.

##### Note

If you have uncommitted changes in your workspace, a dialog box containing the list of conflicting files opens. Choose *Cancel* to abort, or *Reset and Checkout* to remove all uncommitted changes.

The selected branch is checked out. The name of the selected branch is shown in the workspace next to the name of the project.

### Creating a New Local Branch

#### Context

You can create a new local branch referencing any available remote or local branch.

## Procedure

1. In the *Git Pane*, choose  (Add Branch).

The *Create a New Branch* dialog box appears.

2. From the *Source Branch* dropdown list, select the desired local or remote branch.
3. Enter a name for the new local branch.
4. Choose *OK*.

 Note

If you have uncommitted changes in your workspace, a dialog box containing the list of conflicting files opens. Choose *Cancel* to abort, or *Reset and Checkout* to remove all uncommitted changes.

The new local branch is created and checked out.

## Deleting a Local Branch

### Context

If there is only one branch available, it cannot be deleted.

## Procedure

1. In the *Git Pane*, choose  (Delete Branch). The *Delete Branch* dialog box appears showing all the branches of the selected repository.
2. Select one or more branches that are not checked out.
3. Choose *Delete*.

The selected branches are deleted.

## 11.1.11.13.1 Reset Local Branches

You can delete all new objects and references that were added to an existing local branch to make it identical to its remote branch.

### Context

When you reset a branch, all unsynced committed changes are removed, and all staged and unstaged files are reverted to their original state in the local copy of the respective remote branch.

### Procedure

1. In the *Git Pane*, choose *Reset*.
2. Select the branch that you want to revert back to.

**i** Note

The branch that is currently checked out is automatically disabled. It cannot be selected. By default, the corresponding remote branch is selected.

3. Choose a *Reset Type*.

**i** Note

If you choose a *Hard* reset, all changes are removed.

4. Choose *OK* to reset the branch.

## 11.1.11.14 Create Remote Branches

You can create a new branch in the remote repository.

### Context

To work on the remote branch you just created, you still must check out the branch by creating a new local branch for this new remote branch.

**i** Note

To create a local branch, you can choose the menu  *Git*  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

[Multiple Local Branches \[page 302\]](#)

### 11.1.11.15 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.

## Related Information

[Git Commands from the Git History Pane \[page 306\]](#)

### 11.1.11.15.1 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"> <li>1. Choose <i>Tag</i>.</li> <li>2. Enter a name for the new tag.</li> <li>3. Choose <i>OK</i>.</li> </ol> |

|                      |   |   |
|----------------------|---|---|
| Cherry-pick a change | <p>Allows you to apply the changes of the selected commit to only the current branch.</p> <p><b>i Note</b></p> <p>You can only cherry-pick a commit 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.</p> | <ol style="list-style-type: none"> <li>1. Choose <i>Cherry-Pick</i>.</li> <li>2. Choose <i>OK</i> to confirm the commit.</li> <li>3. If there is a conflict, a warning is displayed. From the Git pane, resolve any conflicts as follows:           <ol style="list-style-type: none"> <li>1. Open the Git pane and find the conflicting files in the staging table.</li> <li>2. Edit the files to resolve the conflicts.</li> <li>3. Stage the edited files.</li> <li>4. Commit the changes.</li> </ol> </li> </ol>      |
| Revert a commit      | <p>Allows you to undo all the changes that were incorporated in the selected commit.</p> <p><b>i Note</b></p> <p>You can only revert a commit 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.</p>       | <ol style="list-style-type: none"> <li>1. Select the commit in the list of commits and choose <i>Revert</i>.</li> <li>2. If there is a conflict, a warning is displayed. Open the Git pane to resolve any conflicts as above.</li> </ol>  |
| Check out a commit   | <p>Allows you to take the code from a specific commit and create a new branch based on it.</p>  | <ol style="list-style-type: none"> <li>1. Select the commit in the list of commits, and choose <i>Check Out</i>.</li> <li>2. Enter a name for the new branch.</li> <li>3. 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.</li> </ol> <p><b>i Note</b></p> <p>When checking out an already committed change, conflicts can occur. To resolve these conflicts, make sure that the conflicting file is not open in the staging table.</p> |

|                 |   |  |
|-----------------|---|--|
| Compare commits | <p>Allows you to compare a modified version of your code with the original version from the staging tables in the Git pane.</p> <p>You can also compare two committed versions.</p> | <p>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.</p> <p>For more information, see <a href="#">Compare Code [page 297]</a>.</p> <p>To compare two committed version:</p> <ol style="list-style-type: none"><li>1. Select two commits from the list.</li><li>2. Select the common file to compare.</li><li>3. Choose <a href="#">Compare</a>.</li></ol> |
|-----------------|---|--|

## 11.1.11.16 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 menu, choose [Clone Repository](#) or [Set Remote](#).

If you select [Set Remote](#), your project must have been initialized as a local repository with [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 \[page 291\]](#)

[Initialize a Local Git Repository \[page 292\]](#)

[Set a Remote Repository \[page 292\]](#)

[View Changes in the Gerrit Pane \[page 309\]](#)

---

[Fetch Changes from Gerrit \[page 310\]](#)

## 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](#).

### 11.1.11.16.1 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 \[page 308\]](#)

### 11.1.11.16.2 Fetch Changes from Gerrit

When your repository is set up to work with Gerrit, you work with Git as normal, and review code changes in your Gerrit system. You can also fetch a change from Gerrit and create a local branch from the change, and then collaborate with a colleague using Gerrit before merging the change in Git.

## Procedure

1. From the workspace, select your project.
2. Open the *Git Pane*.
3. Choose *Fetch from Gerrit*.
4. Enter the change that you want to download, and choose *OK*.

Use the ref specification for the change, in the form of `refs/changes/79/2565079/3`. This is available under *Download* in the standard Gerrit UI.

The changes are fetched into a new local branch, and the branch is checked out.

 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.

### 11.1.12 Running Applications in Development Mode

Evaluate the progress of your application's development by running an application to test in-development functionality and design.

You can test your application from the workspace, running in simulators, on devices, with mock data, in the SAP Fiori launchpad environment, or with predefined URL parameters. You can define run configurations to determine how to preview your application. For more information, see [Create Run Configurations \[page 314\]](#).

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 \[page 312\]](#)

Run and test your application using different run configurations.

#### [Create Run Configurations \[page 314\]](#)

Create run configurations that define how your project or unit test is executed.

#### [Run Applications in a Frame \[page 319\]](#)

Check the localizations and functionality of your application by running it in a frame..

#### [Run Applications in the SAP Fiori Launchpad Environment \[page 320\]](#)

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 \[page 322\]](#)

You can run your application as a unit test to determine if it is working properly.

#### [Run Applications with Mock Data \[page 322\]](#)

Run a web application using a client mock server to test your application without connecting to the OData provider.

## 11.1.12.1 Run Applications from the Workspace

Run and test your application using different run configurations.

### Context

You can run and test an application using different parameters to ensure the quality and robustness of the application. You can create and save run configurations with different parameters and then run the application or test it with these run configurations without adjusting the application code. For more information about run configurations, see [Create Run Configurations \[page 314\]](#).

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"><li>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.</li><li>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"><li>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 <a href="#">Create Run Configurations [page 314]</a>.</li><li>If a single match is found, a new web application run configuration is created in the background and the file runs automatically.</li><li>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.</li></ul></li></ul> | <ul style="list-style-type: none"><li>Open without a frame</li><li>Run without mock data</li><li>Run with all existing URL parameters</li></ul> |
| SAP Fiori Launchpad Sandbox | <ul style="list-style-type: none"><li>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 <a href="#">Run Applications in the SAP Fiori Launchpad Environment [page 320]</a>.</li><li>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.</li></ul>  | <ul style="list-style-type: none"><li>Open without a frame</li><li>Run without mock data</li><li>Run with all existing URL parameters</li></ul> |

| Run As    | Default Behavior   | Default Settings |
|-----------|--|------------------|
| Unit Test | <ul style="list-style-type: none"> <li>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.</li> <li>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"> <li>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 <a href="#">Create Run Configurations [page 314]</a>.</li> <li>If a single match is found, a new unit test run configuration is created in the background and the file runs automatically.</li> <li>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.</li> </ul> </li> </ul> |                  |

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](#).

## 11.1.12.2 Create Run Configurations

Create run configurations that define how your project or unit test is executed.

### Context

SAP Web IDE provides default run configurations for your projects. You can create and configure additional run configurations that define how your project or unit test is executed. You can also create a new run configuration by duplicating an existing run configuration and then editing it.

### Procedure

- 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 \[page 73\]](#).
- 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 \[page 315\]](#)

Define general settings for the run configuration.

#### [URL Components Tab \[page 316\]](#)

Define navigation information within the application.

#### [Advanced Settings Tab \[page 317\]](#)

Define advanced settings for the run configurations.

## 11.1.12.2.1 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"><li>• If you chose <i>Run Configurations</i> from the context menu of a file that runs the application, this file name is automatically displayed.</li><li>• If you chose <i>Run Configurations</i> for a project that has only one file that can run the application, this file name is automatically displayed.</li><li>• If there are more files can that run the application, you can select the required file from the drop-down list.</li></ul> |
| Preview Mode         | <p>By default, your application opens without a frame. This preview mode makes issue detection more apparent, as SAP Web IDE runs only the application.</p> <p>Select <i>With Frame</i> to open your application in a frame with configurable viewing options. For more information, see <a href="#">Run Applications in a Frame [page 319]</a>.</p> <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;"><p><b>i Note</b></p><p>This option is not available for unit tests.</p></div>                                   |

| Setting           | Description   |
|-------------------|---|
| Mock Data         | <p>Select <a href="#">Run with mock data</a> 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"> <li>• You have configured settings for the mock server. For more information, see <a href="#">Configure Mock Data Usage [page 72]</a>.</li> <li>• The application that you want to run uses the SAPUI5 OData model with JSON format.</li> </ul> <p><b>i Note</b></p> <ul style="list-style-type: none"> <li>• This option is only available for projects of type SAP Fiori</li> <li>• This option is only available for services of type OData Version 2.</li> <li>• This option is not available for unit tests.</li> </ul>                                      |
| Support Assistant | <p>Select <a href="#">Run with Support Assistant</a> to enable the Support Assistant tool in your web application.</p> <p>Use the Support Assistant tool to check whether your application is built according to the best practices for building SAPUI5 apps. The tool uses a set of pre-defined rules to check all aspects of an application, for example, accessibility, performance, and data-binding. With a simple click, you can check the current state of your application. After execution, you can check the results and apply corrective measures based on the outcome. The tool aims to reduce maintenance and consulting times and to streamline SAPUI5 app development.</p> <p>For more information, see <a href="#">Support Assistant</a>.</p> |

**Parent topic:** [Create Run Configurations \[page 314\]](#)

## Related Information

[URL Components Tab \[page 316\]](#)

[Advanced Settings Tab \[page 317\]](#)

### 11.1.12.2.2 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"> <li>1. To add a new row for an additional parameter, choose <i>Add Parameter</i>.</li> <li>2. In the new row, enter a parameter name and its value.</li> </ol> |
| URL Hash Fragment | <p>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.</p> <p>In the full navigation URL, the fragment identifier is appended after the URL parameters, and is preceded by a hash (#) delimiter.</p>  |

**Parent topic:** [Create Run Configurations \[page 314\]](#)

## Related Information

[General Tab \[page 315\]](#)

[Advanced Settings Tab \[page 317\]](#)

### 11.1.12.2.3 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 <a href="#">Use another version</a> and select the required version.</p> <p><b>i Note</b></p> <p>If a SAPUI5 version is specified in the project <code>neo-app.json</code> file, that is the default version.</p> <p><b>i Note</b></p> <p>If a minimum SAPUI5 version is specified in the project <code>manifest.json</code> file, the version list displays only versions equal to or higher than the specified version.</p> |
| Application Destination Resources | <p>You can test your application with a destination system that is different from the one defined in the application <code>neo-app.json</code> file. Map the destinations that are defined in your project to any system that is included in your SAP Cloud Platform account.</p>   |
| Application Resources             | <p>If your project references any SAPUI5 resources, you can change the version of the library to use for this run configuration. By default, the version that you selected when you made the reference is used.</p> <p>Choose <a href="#">Get Library Versions</a> 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 <a href="#">Use my workspace first</a>. 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 Configurations \[page 314\]](#)

## Related Information

[General Tab \[page 315\]](#)

[URL Components Tab \[page 316\]](#)

### 11.1.12.3 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.

 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.

 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.

## 11.1.12.4 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 backend services.

### [Test a Single SAP Fiori Application \[page 320\]](#)

Test your SAP Fiori application in a simplified SAP Fiori launchpad environment.

### [Test Multiple SAP Fiori Applications in the FLP Sandbox \[page 321\]](#)

Test the interaction of multiple SAP Fiori applications in a simplified SAP Fiori launchpad environment.

### 11.1.12.4.1 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 \[page 73\]](#).

#### Procedure

1. In the workspace, select the `Component.js` file.
2. In the main toolbar or context menu, choose *Run*.
  - Choose a recently-used run configuration from the list. The list displays only configurations that you can run for the selected project.
  - Choose  *Run As*  *SAP Fiori Launchpad Sandbox*. The application runs with the last-used run configuration that matches the SAP Fiori component project type. If there is no matching run configuration, a default run configuration is created and used.
  - Choose *Run Configurations* to create a new run configuration and run your project.  
For more information, see [Create Run Configurations \[page 314\]](#).

**Task overview:** [Run Applications in the SAP Fiori Launchpad Environment \[page 320\]](#)

## Related Information

[Test Multiple SAP Fiori Applications in the FLP Sandbox \[page 321\]](#)

### 11.1.12.4.2 Test Multiple SAP Fiori Applications in the FLP Sandbox

Test the interaction of multiple SAP Fiori applications in a simplified SAP Fiori launchpad environment.

#### Prerequisites

- Make sure you have 2 (or more) applications in your SAP Web IDE workspace.

#### Procedure

1. From the workspace, select the app from which you want to navigate.
2. From the context menu, select *Enable App To App Navigation*. The *App To App Navigation* dialog box opens.
3. From the *Navigate to* dropdown list, select one or more applications to which you want to enable navigation.
4. Click *Enable*. A new `FLPSandbox` project is created in the workspace. This project contains a `neo-app.json` file with routes to the origin and target applications, as well as a `fioriSandboxConfig.json` file with an intent section for each application.

For more information on the `fioriSandboxConfig.json` file, see [Local Configuration File for the Launchpad Sandbox](#).

5. Implement the navigation code in the origin application as a callback to the navigation event.

The following APIs contain information that can help you establish the navigation between apps:

- [CrossApplicationNavigation](#)
- [NavigationHandler](#)

#### i Note

Make sure to use the same name for the intent in the `fioriSandboxConfig.json` file and for the navigation. This name should be aligned with the name used in the productive SAP Fiori Launchpad.

6. Right-click the *FLP Sandbox* project and select *Run* *App2App Navigation* to test that the navigation works.

*App2App Navigation* is a run configuration generated by default when the App2App navigation is enabled. It is an *SAP Fiori Launchpad Sandbox* configuration which has the *Use my workspace first* checkbox selected.

**Task overview:** [Run Applications in the SAP Fiori Launchpad Environment \[page 320\]](#)

---

## Related Information

[Test a Single SAP Fiori Application \[page 320\]](#)

### 11.1.12.5 Run Unit Tests on Applications

You can run your application as a unit test to determine if it is working properly.

#### Context

You can configure as many unit tests as you want and then run them on different HTML files in your project. There are two ways to configure and run a unit test for your application:

- Using a customized run configuration.  
For more information, see [Create Run Configurations \[page 314\]](#).
- Using a default run configuration.  
For more information, see [Run Applications from the Workspace \[page 312\]](#).

### 11.1.12.6 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 \[page 72\]](#).

#### 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.

## 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 \[page 324\]](#)

[Create Run Configurations \[page 314\]](#)

### 11.1.12.6.1 Edit Mock Data

You can model the service data that you want to use as mock data in your application.

## Prerequisites

- The service must be of type OData Version 2.
- The project must be of type SAP Fiori.

## Context

Depending on the project template that was used to create the project, the metadata file and mock data files are stored in the following folders:

| Project                               | Location of Metadata File | Location of Mock Data Files      |
|---------------------------------------|---------------------------|----------------------------------|
| Based on new SAPUI5 project templates | webapp/localService       | webapp/localService/<br>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.

### 11.1.12.6.2 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 \[page 72\]](#).

## 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.

### 11.1.13 Building Applications

You can build your project using Grunt.

Grunt is a task runner based on the `Node.js`<sup>®</sup> runtime environment. Grunt allows you to automate tasks that front-end developers perform on a regular basis, such as minifying JavaScript and CSS files, unit testing, linting files to check for errors, compiling CSS preprocessor files (LESS, SASS), and more.

To use Grunt capabilities in the full-stack development version, you do not need to set up your environment by installing the `Node.js` runtime environment or Grunt. Instead, you just need to add two mandatory files to your project, `package.json` and `Gruntfile.js`, which contain configurations for running npm and Grunt. After that, you activate the Grunt build from your project context menu.

The `package.json` and `Gruntfile.js` configuration files are part of your project sources. These files allow you to apply exactly the same Grunt build for the project from a CLI within your continuous integration process outside of SAP Web IDE.

SAP supports the best practice configurations for SAPUI5 applications using `grunt-openui5`, which is a set of Grunt plugins provided by SAP that improves application performance considerably in the productive environment. For more information, see [Using Grunt to build an SAPUI5 app](#).

### Related Information

[Run a Grunt Build \[page 326\]](#)

## 11.1.13.1 Run a Grunt Build

Instructions for enabling and configuring a Grunt build.

### Context

To enable the Grunt build for your project, you first need to add two mandatory configuration files to the root of your project: `package.json` and `Gruntfile.js`.

### Procedure

1. To create each file, in the SAP Web IDE *Workspace*, right-click your project folder, then choose  **New > File**. Add the required content for each file as follows:

- `package.json`

Enter the following code in the `package.json` file:

```
{  
  "name": "grunt-build",  
  "version": "0.0.1",  
  "description": "Grunt build",  
  "private": true,  
  "devDependencies": {  
    "@sap/grunt-sapui5-bestpractice-build": "1.3.33"  
  }  
}
```

When you select *Build* in the context menu, these settings instruct npm to install Grunt and the `grunt-sapui5-bestpractice-build` Grunt plugin that contains tasks for building your SAPUI5 project.

#### Note

The `grunt-sapui5-bestpractice-build` Grunt plugin is published on the SAP npm registry. To run the Grunt build using this plugin outside of SAP Web IDE Full-Stack, such as from a CLI as part of the CI process, add the following configuration option to the npm configuration file:

```
@sap:registry=https://npm.sap.com/
```

For more information, see <https://docs.npmjs.com/files/npmrc>.

- `Gruntfile.js`

Include the tasks listed below as shown in the code example:

```
module.exports = function(grunt) {  
  'use strict';  
  grunt.loadNpmTasks('@sap/grunt-sapui5-bestpractice-build');  
  grunt.registerTask('default', [  
    'lint',  
    'clean',  
    'build'  
  ]);  
}
```

```
};
```

These settings instruct Grunt to load the plugin that provides tasks for building SAPUI5 applications and running the tasks in the specified order:

- lint
  - Validates the project code using `ESLint` according to the rules defined in the `.eslintrc` configuration file located in the root of your project.

#### Note

- If the `.eslintrc` file is not found, the code is validated against default `ESLint` rules.
- The default `ESLint` rules do not align with projects generated from SAP Web IDE templates. If your project was generated from an SAP Web IDE template, we recommend using the generated rules as described below.
- Central rule definition is not considered for the `lint` task in the Grunt build. For more information, see [Setting JavaScript Rules for All Users](#).

You can import your own configuration file or generate one in SAP Web IDE as follows:

1. In the `Workspace`, right-click your project root folder, then choose  `Project Settings`  `Code Checking`  `JavaScript`.
2. In the `Validator` dropdown list, select `Basic JavaScript`.
3. Add or modify the code formatting rules according to your requirements and then choose `Save`. If you want to display the generated file, you need to first enable hidden files from the SAP Web IDE toolbar:  `View`  `Show Hidden Files`.

#### Note

You can skip the linting step for your project by removing the `lint` task from the `Gruntfile.js` file.

- clean
    - Cleans the `dist` target folder from the previous build results.
  - build
    - Produces a new build output in the `dist` folder of your project that is ready and optimized for better performance in the productive environment. The following tasks are executed during the build:
      - Minification of `.css` files
      - Minification of JavaScript files (minified files)
      - Copying of the original files to the `dist` folder with `-dbg` suffix added for debugging purposes
      - Generation of the `Component-preload.js` and `Component-preload-dbg.js` preload files for the debug and minified files
      - Minification of the preload file
2. To run the Grunt build, in the `Workspace`, right-click your project and select `Build`.

When the build finishes successfully, a new `dist` folder appears in your project folder and is automatically added to `.gitignore`, ensuring that the folder is not checked in the project's Git repository. Any build errors are reported to the console, which you can display from the SAP Web IDE toolbar by selecting  `View`  `Console`.

### Caution

The `dist` folder and its contents should be treated as read-only. Don't create or change files in this folder – in any case, all changes are always overwritten by the following build.

### Note

In the standard SAP Web IDE, it is possible to influence source/target/excluded folders and files using the [SAPUI5 Client Build Configuration](#) screen. This functionality is not available in SAP Web IDE Full-Stack. The source folder is always `webapp`; the target folder is always `dist`; all folders and files are included. The build is always triggered before deployment.

### Tip

Our recommendation is always to include the `Gruntfile.js` and `package.json` files described in this topic if you want to deploy minified or bundled code.

## Related Information

[Grunt Build in SAP Web IDE](#) 

## 11.1.13.2 Run Build Results

After building the application using Grunt you can test the build results.

## Procedure

1. In the Workspace, in the project folder, locate and open the `dist` folder, then right-click the required HTML file.
2. In the context menu, choose [Run](#), then choose the configuration you require.

## Related Information

[Application Build \[page 337\]](#)

[Running Applications in Development Mode \[page 311\]](#)

## 11.1.14 Deploying Applications

You can deploy new applications from SAP Web IDE to different servers.

### [Deploy Applications to the SAPUI5 ABAP Repository \[page 329\]](#)

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 \[page 331\]](#)

You can update your applications that reside in the SAPUI5 ABAP repository.

### [Deploy Applications to SAP Cloud Platform \[page 332\]](#)

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 \[page 334\]](#)

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 \[page 337\]](#)

You can check whether your application has been deployed to SAPUI5 ABAP Repository and/or SAP Cloud Platform.

### [Application Build \[page 337\]](#)

Applications in production environments generally require a build to package resources in an optimal format, for example, to improve loading and delivery.

### 11.1.14.1 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 \[page 31\]](#).

#### Procedure

1. In your workspace, right-click the desired project.
2. Select [Deploy > Deploy to SAPUI5 ABAP Repository](#).
3. From the [System](#) dropdown list, select the desired system.

#### 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 \[page 331\]](#).

4. Select the *Deploy a new application* radio button and choose *Next*.
5. Provide a name and meaningful description for the application.
6. Choose *Browse*.

**i Note**

The *Browse* button is enabled only after providing a name for the application.

**i Note**

If you selected an S/4HANA system, a prefix is added to application name and a package is assigned automatically. The *Browse* button in this case is disabled.

7. In the *Package Selection* search field, enter the name of the desired package and choose *OK*.
8. Choose *Next*.
9. If the selected package is local, choose *Finish*. If it requires transport, select a transport request for your application using one of the following options:
  - Enter a request number.
  - Create a new request and enter its description.

**i Note**

The request ID is automatically generated.

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 Grunt is used to build the application, only the content of the build target folder (called `dist`) is deployed. Otherwise, if the application contains a `webapp` folder, only the content within this folder is deployed.

For more information, see [Application Build \[page 337\]](#).

---

## Related Information

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

### 11.1.14.1.1 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 \[page 317\]](#)).
- Always test your application on the server after deployment.

### 11.1.14.2 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.

**i Note**

If you receive a message informing you that the system you selected supports only local object creation, implement SAP Note [2046730](#). Otherwise, applications whose packages are not \$TMP will not be updated.

7. If the selected package is local, choose *Next* and then *Finish*. If the package requires transport, choose *Next* to select or create a transport request for your application using one of the following options:
  - Enter a request number.
  - Create a new request and enter its description.

**i Note**

The request ID is automatically generated.

- Select a request in which you are involved from the table.

8. Choose *Next* and then *Finish*.
9. A list showing updated files that will be overwritten and new files that have been added is displayed. Choose *OK* to confirm the update.

### 11.1.14.3 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 SAP Cloud Platform 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 \[page 334\]](#).

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 SAP Cloud Platform 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 Grunt is used to build the application, only the content of the build target folder (called `dist`) is deployed.

For more information, see [Application Build \[page 337\]](#).

### 11.1.14.4 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 \[page 332\]](#).
- 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 [Creating a Site Instance in 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](#).

#### Procedure

1. In the workspace, right-click the desired application and choose [Deploy](#) [Register to SAP Fiori Launchpad](#).

#### i Note

After deploying an application, you can also access the wizard by choosing [Register to SAP Fiori Launchpad](#) from the [Successfully Deployed](#) dialog box.

2. Choose [Next](#).

3. In the *General Information* step, select the provider account to which you want to register the application.
4. Enter the application name.
5. Optional: enter a description and intent, and then choose *Next*.
6. In the *Tile Configuration* step, choose the tile type:
  - *Static* - Enter a title, a subtitle, and choose *Browse* to select an SAPUI5 icon for the tile.
  - *Dynamic* - SAP Web IDE fetches the application's OData service (if the service is already configured in the application's Configuration.js or Component.js files). If the service is not already configured, enter it manually in the *Service URL* field and then choose *Get Collections*.

**i Note**

The service path should be relative, for example, /sap/opu/odata/iwfnd/RMTSAMPLEFLIGHT.

The service URL is used to get the service's addressable collections. Select the desired addressable collection from the *Collection* dropdown list.

The tile is updated with the `count` property of the selected collection.

**i Note**

The service must have at least one addressable collection.

The *Number Unit* field is populated with a default value of the entity type of the selected collection. This value can be edited.

The *Refresh Rate (Sec)* field is populated with a default value of 10 seconds. This value can be edited.

**i Note**

This field accepts only numbers because it expects a number of seconds.

7. Choose *Next*.
8. In the *Assignment* step, depending on the SAP Fiori launchpad that you selected, assign the tile to a *Site*, a *Catalog*, and a *Group*.
  - Sites are SAP Fiori launchpad sites.
  - Catalogs are authorization objects used to enable role-based access to apps and groups for a particular launchpad site.
  - Groups are a titled grouping in which apps (represented by tiles) are organized in a launchpad site.
9. Choose *Next*.
10. Choose *Finish* to confirm and register your application to SAP Fiori launchpad.

## Results

After the registration is complete, a success dialog box appears with a link to your application on SAP Fiori launchpad.

**i Note**

For the application to run, it must be started.

## 11.1.14.4.1 Connect a Project to the SAP Cloud Platform Git Repository

When you deploy the application to SAP Cloud Platform, the source code is not automatically pushed to the SAP Cloud Platform Git. You can connect your project to any Git repository and push your changes there later.

### Prerequisites

You must be an account administrator.

### Context

The application source code should be managed in Git.

### Procedure

1. Go to Tools SAP Cloud Platform Cockpit Git Repositories .

The screenshot shows the SAP HANA Cloud Platform Cockpit interface. On the left, a sidebar menu is visible with various options like Java Applications, Database Systems, and Git Repositories. The 'Git Repositories' option is highlighted with a red box. The main area displays a table of Git repositories. Each row contains the repository name, status (ACTIVE), and a progress bar indicating usage. The table includes rows for fioriapp, fioriapp076633, fioriappl, fioriappnewone, fioribase, fioricarrinew, fioridetail, fiorilib, fiorimaster, fiorimd, fiorimdapp, fiorimdapp12, fioritest, firolike, flights, flightss, fromworkspace, fs, full, fullextmounika, fullscreen, and gaia. All repositories are marked as ACTIVE and show a 1% usage level.

| Repository     | Status | Usage (%) |
|----------------|--------|-----------|
| fioriapp       | ACTIVE | 1%        |
| fioriapp076633 | ACTIVE | 1%        |
| fioriappl      | ACTIVE | 1%        |
| fioriappnewone | ACTIVE | 1%        |
| fioribase      | ACTIVE | 1%        |
| fioricarrinew  | ACTIVE | 1%        |
| fioridetail    | ACTIVE | 1%        |
| fiorilib       | ACTIVE | 1%        |
| fiorimaster    | ACTIVE | 1%        |
| fiorimd        | ACTIVE | 1%        |
| fiorimdapp     | ACTIVE | 1%        |
| fiorimdapp12   | ACTIVE | 1%        |
| fioritest      | ACTIVE | 1%        |
| firolike       | ACTIVE | 1%        |
| flights        | ACTIVE | 1%        |
| flightss       | ACTIVE | 1%        |
| fromworkspace  | ACTIVE | 1%        |
| fs             | ACTIVE | 1%        |
| full           | ACTIVE | 1%        |
| fullextmounika | ACTIVE | 1%        |
| fullscreen     | ACTIVE | 1%        |
| gaia           | ACTIVE | 1%        |

2. Create a new Git repository, select it and locate the created Git URL you want to use. For more information, see [Creating a Repository](#).
3. Initialize your local repository and connect it to the remote GIT repository by using the URL you obtained from the cockpit. For more information, see [Initialize a Local Git Repository \[page 292\]](#).

- 
4. Use the Git operations to fetch, commit, and push your changes. For more information, see [Using Source Control \(Git\) \[page 283\]](#).

## 11.1.14.5 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.

## 11.1.14.6 Application Build

Applications in production environments generally require a build to package resources in an optimal format, for example, to improve loading and delivery.

If you include the `Gruntfile.js` and `package.json` files in your project, the Grunt build is activated when you deploy your application to either SAP Cloud Platform or the SAPUI5 ABAP repository. The results of the build (the `dist` folder content) are deployed.

If the `Gruntfile.js` and `package.json` files do not exist in your project, the build step is skipped and the original project sources are deployed.

## Related Information

[Building Applications \[page 325\]](#)

### 11.1.15 Extending SAPUI5 Applications

You can extend SAPUI5 applications residing remotely on the SAPUI5 ABAP repository or SAP Cloud Platform.

 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 \[page 338\]](#)

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 \[page 340\]](#)

You can extend an existing SAP Fiori application that resides on SAP Cloud Platform without importing it to SAP Web IDE.

[Create New Extensions \[page 341\]](#)

Extensions enable you to change the views or the logic of an extended project.

#### 11.1.15.1 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 \[page 31\]](#).

## 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* .
2. Choose  *Select Application* . 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 \[page 331\]](#).

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](#)

### 11.1.15.2 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

 Note

This feature is unavailable in SAP Web IDE personal edition.

#### Procedure

1. From the *File* menu, choose  *New* .
2. Choose  *Select Application* . 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.

## 11.1.15.3 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 \[page 342\]](#)

You can extend an existing SAP Fiori extension project from the extensibility pane, as well as by using the Extension wizard.

#### [Extend Controllers \[page 345\]](#)

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 \[page 347\]](#)

You can extend a view using an extension point.

## [Hide Controls \[page 348\]](#)

You can hide a specific control in the original application.

## [Edit Strings \[page 349\]](#)

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 \[page 350\]](#)

You can replace the extended application's OData service with a new OData service.

## [Replace Views \[page 351\]](#)

You can replace a specific view in an original application with a new view.

### 11.1.15.3.1 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.

#### 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.

#### Note

- You cannot extend elements that do not have configured IDs. In addition, you cannot hide elements that do not have a `visible` property.
- Elements that are set as `visible` manually by the view's controller will not be hidden.
- You can try and hide these controls by replacing the view or extending the controller that hides the control and override the method.

- *Extended Elements* - Shows all UI elements that are already extended.

- *Extensible Elements* - Shows all UI elements that have a configured ID.

- *Extension Points* - Shows all extension points that are available in the application (identifiable by the icon).



You can select a UI element's ID in the *Outline* pane and see it highlighted in the previewed application. Likewise, if you hover over a UI element in the previewed application, the respective element's ID is highlighted in the extensibility pane.

When you click a UI element in the previewed application when in extensibility mode, the element is selected and highlighted in the *Outline* pane. You can deselect it by clicking it again.

#### Note

This two-way selection is available only if *Extensibility Mode* is selected from the menu bar.

#### 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.

#### 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*.

#### Note

If an extended application requires URL parameters and/or a hash fragment, these can be defined as a run configuration in the extension project.

The extensibility pane uses the first run configuration of type `Web Application` defined in the extension project.

For more information, see [Create Run Configurations \[page 314\]](#).

## 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 \[page 341\]](#)

## Related Information

- [Extend Controllers \[page 345\]](#)
- [Extend Views \[page 347\]](#)
- [Hide Controls \[page 348\]](#)
- [Edit Strings \[page 349\]](#)
- [Replace OData Services \[page 350\]](#)
- [Replace Views \[page 351\]](#)

## 11.1.15.3.1.1 Extensibility Pane Troubleshooting

Steps you can take if you have trouble using the Extensibility pane.

The following lists error messages you may receive when using the Extensibility pane, possible causes of the error, and possible solutions.

## Error in the Extensibility Pane

- The first time you open the extensibility pane you might get the following error:

*An error has occurred. Click OK to refresh the application.*

**Solution:**

Click **OK** to refresh the application.

- When in Extensibility Mode, if you select a UI element in the application and it does not appear in the outline (for example when showing only **Extended Elements**), you might get the following error:

**Cannot find the UI element in the outline. Make sure you are using the “Extensible Elements” or “All Elements” outlines.**

**Solution:**

Make sure you are using the **Extensible Elements** or the **All Elements** outlines.

## Unexpected Behavior when Running an Extended Application in the Extensibility Pane

When running an extended application within the extensibility pane you might encounter unexpected behavior, for example, no data is displayed.

**Solution:**

When extending an application which requires URL parameters and/or a hash fragment, these should be defined in the run configuration of the extension project.

### 11.1.15.3.2 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 \[page 341\]](#)

## Related Information

[Extend UI Elements Using the Extensibility Pane \[page 342\]](#)

[Extend Views \[page 347\]](#)

[Hide Controls \[page 348\]](#)

[Edit Strings \[page 349\]](#)

[Replace OData Services \[page 350\]](#)

[Replace Views \[page 351\]](#)

# 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.

### 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  *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.

### 11.1.15.3.3 Extend Views

You can extend a view using an extension point.

#### Prerequisites

You must have defined extension points in the original application.

#### Procedure

1. Select the extension project to which you want to add the extension.
2. From the *File* menu, choose  *New* > *Extension*.
3. Select the extension project to which you want to add the extension and choose *Next*.
4. Select *Extend View/Fragment* and choose *Next*.
5. Select the view or fragment that you want to extend.
6. Select the desired extension point.

 **Note**

Not all views or fragments have extension points.

7. Choose *Next*.
8. Choose *Finish* to add the extension to the selected extension project.

**Task overview:** [Create New Extensions \[page 341\]](#)

#### Related Information

[Extend UI Elements Using the Extensibility Pane \[page 342\]](#)

[Extend Controllers \[page 345\]](#)

[Hide Controls \[page 348\]](#)

[Edit Strings \[page 349\]](#)

[Replace OData Services \[page 350\]](#)

[Replace Views \[page 351\]](#)

## 11.1.15.3.4 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 \[page 341\]](#)

### Related Information

[Extend UI Elements Using the Extensibility Pane \[page 342\]](#)

[Extend Controllers \[page 345\]](#)

[Extend Views \[page 347\]](#)

---

[Edit Strings \[page 349\]](#)  
[Replace OData Services \[page 350\]](#)  
[Replace Views \[page 351\]](#)

### 11.1.15.3.5 Edit Strings

The i18n Resource Text Customization extension allows you to copy the `i18n` folder of the original application to your extended application. This allows you to edit the UI strings in the extended application without altering the original application.

#### Prerequisites

The original application must have an `i18n` folder with at least one `Properties` file that contains the relevant strings.

#### Procedure

1. Select the extension project to which you want to add the extension.
2. From the `File` menu, choose  `New > Extension`.
3. Select the extended application that you want to customize.
4. Choose `Next`.
5. Select the `i18n Resource Text Customization` tile.
6. Choose `Next`.
7. Choose `Finish` to confirm and add the extension.

#### Results

The `i18n` folder of the original application is copied to your extended application, including all its `.properties` files. You can change one or more of the strings in the `.properties` file and run your extended application to see them in runtime. The original application remains unchanged.

**Task overview:** [Create New Extensions \[page 341\]](#)

#### Related Information

[Extend UI Elements Using the Extensibility Pane \[page 342\]](#)

[Extend Controllers \[page 345\]](#)  
[Extend Views \[page 347\]](#)  
[Hide Controls \[page 348\]](#)  
[Replace OData Services \[page 350\]](#)  
[Replace Views \[page 351\]](#)

## 11.1.15.3.6 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.

#### 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 \[page 311\]](#).

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 \[page 341\]](#)

## Related Information

[Extend UI Elements Using the Extensibility Pane \[page 342\]](#)

[Extend Controllers \[page 345\]](#)

[Extend Views \[page 347\]](#)

[Hide Controls \[page 348\]](#)

[Edit Strings \[page 349\]](#)

[Replace Views \[page 351\]](#)

### 11.1.15.3.7 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 \[page 341\]](#)

## Related Information

[Extend UI Elements Using the Extensibility Pane \[page 342\]](#)

[Extend Controllers \[page 345\]](#)

[Extend Views \[page 347\]](#)

[Hide Controls \[page 348\]](#)

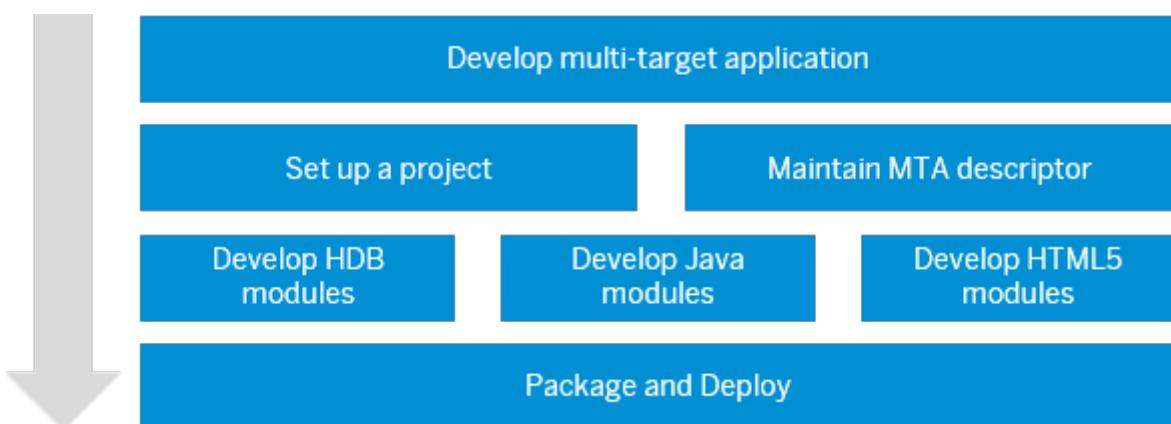
[Edit Strings \[page 349\]](#)

[Replace OData Services \[page 350\]](#)

## 11.2 Developing Multi-Target Applications

Multi-target, or multi-tier applications are comprised of multiple software modules representing the data, business logic and UI tiers. These modules are created with different technologies and are deployed to different target platforms, yet share the same development lifecycle.

The multi-target application concept aims at orchestrating the deployment of all these modules so that all runtime dependencies are properly resolved and the application functions as expected. This is achieved by supplying to the deployment tools a set of descriptors that define the interdependencies and deployment scenarios for all modules in the application.



- [Developing Multi-Target Applications \[page 352\]](#)
- [Setting Up Application Projects \[page 359\]](#)
- [Inside an MTA Descriptor \[page 355\]](#)
- [Developing SAP HANA Database \(HDB\) Modules \[page 362\]](#)
- [Developing Java Modules \[page 366\]](#)
- [Developing HTML5 Modules \[page 375\]](#)
- [Packaging and Deploying Applications to Production Systems \[page 385\]](#)

## Terms and Concepts

| Term                           | Description  |
|--------------------------------|--|
| Multi-target application (MTA) | An application comprised of multiple software modules, which are created with different technologies and deployed to different target platforms, yet share the same lifecycle. In the context of this guide, an "application" is an MTA.   |
| Target platform                | A platform to which a module is deployed, such as an SAP HANA database.  |
| MTA descriptor                 | A YAML file named <code>mta.yaml</code> that contains a list of all entities, such as modules, resources, and properties that belong to an application or are used by it at runtime, and the dependencies between them. It is automatically generated when an MTA project is created or modified, or when a module is added or removed. The developer needs to edit the descriptor manually to define resources, properties, and dependencies, as well as fill in missing information. |
| Module                         | A self-contained application of a certain type, which is developed, packaged, and deployed to a target platform as part of an MTA.   |

| Term               | Description   |
|--------------------|---|
| Module type        | <p>A type that defines the structure, development technology, and target platform of a module. SAP Web IDE supports the following module types, representing the three application tiers:</p> <ul style="list-style-type: none"> <li>• SAP HANA database (HDB) module - represents the data tier.</li> <li>• Java and Node.js modules - represent the business logic tier.</li> <li>• HTML5 module - represents the UI tier.</li> </ul> <div style="background-color: #ffffcc; padding: 10px;"> <p><b>i Note</b></p> <p>In the current version, Node.js modules are not supported.</p> </div> |
| Resource           | Any resource, such as an external service, property, or environment variable, that is required by a module at runtime but not provided by the module itself.  |
| Property           | A property (key-value pair) of an application, module, or resource, that is used during deployment or runtime.  |
| Parameter          | A reserved variable belonging to a module or resource, whose value is used during deployment or runtime. Parameters can be read-only, write-only, or read-write. The values of writable parameters can be specified in the descriptor.  |
| Dependency         | <p>A relationship between a module and another module, resource, or property, such as <i>provides</i> and <i>requires</i>.</p> <ul style="list-style-type: none"> <li>• <i>provides</i>: indicates the properties or parameters that are provided by a module or resource to other modules.</li> <li>• <i>requires</i>: indicates other modules or resources that are required by a module in order to run.</li> </ul>  |
| Deployment archive | An archive similar to JAR into which all the application artifacts are packaged for deployment.   |

#### [Inside an MTA Descriptor \[page 355\]](#)

The multi-target application (MTA) descriptor contains the metadata of all entities comprising an application or used by it during deployment or runtime, and the dependencies between them.

#### [Setting Up Application Projects \[page 359\]](#)

You can set up a multi-target application (MTA) project by creating it from scratch, importing from an archive, or cloning from a Git repository.

#### [Developing SAP HANA Database \(HDB\) Modules \[page 362\]](#)

An SAP HANA database (HDB) module is a collection of related design-time database artifacts, such as data models, views, or procedures.

#### [Developing Java Modules \[page 366\]](#)

A Java module is a collection of related Java files and service definitions. Java modules implement the business logic of your application, either instead of or in addition to Node.js modules. A Java module can be either a Java Web Archive (WAR) or Java Archive (JAR) built with Apache Maven.

#### [Developing HTML5 Modules \[page 375\]](#)

An HTML5 module is a collection of related HTML5 files that implement the user interface of your application.

#### [Developing SAP S/4HANA Service Extensions \[page 380\]](#)

SAP Web IDE enables you to create an OData V4 service that extends an existing S/4HANA service. The new service exposes additional data from a different source, such as an SAP HANA database.

#### [Packaging and Deploying Applications to Production Systems \[page 385\]](#)

At the last stage of multi-target application (MTA) development, you need to package your application and deploy it to a target production system.

#### [Developing Applications Using the SAP Cloud Platform Programming Model \[page 386\]](#)

You can create a full-stack application with CRUD capabilities in SAP Web IDE using the SAP Cloud Platform programming model (experimental).

### 11.2.1 Inside an MTA Descriptor

The multi-target application (MTA) descriptor contains the metadata of all entities comprising an application or used by it during deployment or runtime, and the dependencies between them.

The MTA descriptor (the `mta.yaml` file located in the root project folder) is automatically generated when an application project is created from scratch, and it is updated when the project properties change or when a module is added or removed. However, not all the necessary information can be generated automatically. You need to maintain the descriptor manually to define resources, properties, and dependencies, as well as fill in missing information.

#### MTA Editor

The MTA descriptor is written in the YAML format, which has strict syntax requirements. You can edit the descriptor in the text-based code editor, but we recommend you use the visual MTA editor because it provides input validation.

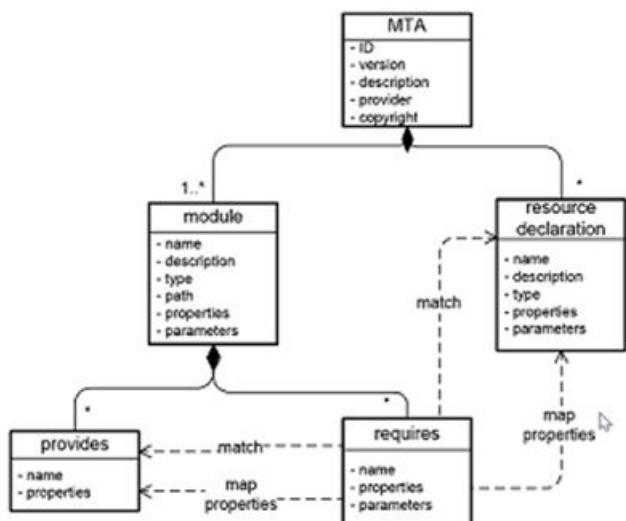
To set the MTA editor as the default for the MTA descriptor, go to [Preferences](#) [Default Editors](#) and set the editor for the [MTA Application Descriptor](#).

##### Note

The MTA visual editor removes comments and formats the file. If you want to add comments, use the code editor. To open the code editor, either make the code editor the default editor or right-click the file and choose [Open With](#) [Code Editor](#).

If you edit the file with the code editor, it is important to use spaces rather than tabs for indentation.

## MTA Descriptor Model



## MTA Descriptor Example

```
ID: com.sap.node.hello.world.db
version: 1.0.0
description: A Hello World sample application
provider: SAP Sample generator
copyright: 2016 SAP SE
modules:
  - name: node-hello-world-db
    type: hdb
    path: db
    requires:
      - name: hdi-container
    provides:
      - name: node-hello-world-db
  - name: node-hello-world
    type: html5
    path: web
    requires:
      - name: uaa
      - name: backend_api
noje.js module
  group: destinations
  properties:
    name: nodejs
  'destinations' variable
    url: ~{url}
    forwardAuthToken: true
  - name: node-hello-world-backend
    type: nodejs
    path: js
    requires:
      - name: node-hello-world-db
      - name: hdi-container
      - name: uaa
    provides:
      - name: backend_api
    properties:
```

```

        url: ${default-url}
# Resources describe required services
resources:
  - name: hdi-container
    type: com.sap.xs.hdi-container
  - name: uaa
    type: com.sap.xs.aaa

```

## MTA Descriptor Elements

| Sections | Element      | Description   | Possible Values  |
|----------|--------------|---|--|
| General  | ID           | The unique application ID.  | <p>Any number of unicode characters in a reverse-URL dot-notation, for example <code>com.sap.mta.sample</code>.</p> <p>The ID should be unique in the target runtime environment.</p>              |
| General  | version      | The application version.  | Should follow the semantic versioning standard format (currently only the basic MA-JOR.MINOR.PATCH format is supported). For more information, refer to the official site for semantic versioning. |
| General  | description, | Optional. A description of the application.   |  |
| General  | provider     | Optional. The application vendor/provider name.   |  |
| General  | copyright    | Optional. The copyright notice of the provider.   |  |
| modules  | name         | The module name.  | Should be unique within the descriptor.  |
| modules  | path         | The relative path to a module from the application root.  | For example, <code>./backend</code>  |
| modules  | type         | The module type, which defines the design-time tools and builders required for the module.  | One of the following values: <code>hdb</code> , <code>nodejs</code> , <code>html5</code> .   |
| modules  | requires     | A subsection of a module section that contains the names of resources, other modules, and/or properties provided by other modules, which are required by the current module to run. | <p>For example:</p> <pre> requires:   - name: backend     properties:       name: external       url: ~{url} </pre>  |

| Sections           | Element     | Description  | Possible Values  |
|--------------------|-------------|--|--|
| modules            | provides    | A subsection of a module section that specifies the properties provided by the current module or resource.   | <p>For example:</p> <pre>provides:   - name: price_opt     properties:       protocol: http       uri:         host.domain</pre> |
| modules, resources | properties  | Optional. A flat or hierarchical collection of properties, provided or required by a module or resource.   | <p>For example:</p> <pre>properties:   - name: node-hello-world-db   - name: hdi-container   - name: uaa</pre>                   |
| modules, resources | parameters  | <p>Optional. Reserved variables that can contain read-only, read-write, or write-only values, which can be used by deployment tools, but not at runtime.</p> <p>Parameters can be referenced with placeholders enclosed in \${ and }, which are resolved during deployment or runtime.</p> | <p>For example:</p> <pre>parameters: domain: price.sap.com</pre>   |
| modules            | group       | Optional. A named group of properties from different providers used as a single lookup object.   | <p>For example:</p> <pre>group: DESTINATIONS</pre>   |
| modules            | description | Optional. A description of the module, which does not appear in the application UI.  |  |

## Related Information

[The Multi-Target Application Model](#)

[Configuring Resource Parameters \[page 358\]](#)

### 11.2.1.1 Configuring Resource Parameters

How to configure resource parameters within and outside an MTA descriptor.

For the application deployment you sometimes need to configure additional parameters for resources. You can configure these parameters directly in the MTA descriptor, for example:

```
resources:
  - name: hdi-container
    type: com.sap.xs.hdi-container
    parameters:
      config:
```

```
schema: ${default-container-name}
```

These configurations can be lengthy. To keep the MTA descriptor short and easy to maintain, you can define them in separate .json files within the project, and specify the file's path in the resource's path of the corresponding resource. For example, it is a good practice to define security configurations in a separate file:

```
resources:
- name: node-uaa
  type: com.sap.xs.uaa
  parameters:
    path: ./xs-security.json
```

Here is how you can specify an external configuration file in the MTA descriptor.

```
resources:
- name: some_resource
  parameters:
    path: myconfig/configuration.json #path to the json file within the project
```

Let's say, the specified file contains a configuration { "config1": "somevalue", "config2": [ 123, 456 ] }.

During the application build, the .json file will not be included in the resulting MTA archive. Rather, the content of this file will be integrated into the deployment descriptor (`mtad.yaml`) in the following way:

```
resources:
- name: some_resource
  parameters:
    config:
      config1: somevalue
      config2:
        - 123
        - 456
```

## 11.2.2 Setting Up Application Projects

You can set up a multi-target application (MTA) project by creating it from scratch, importing from an archive, or cloning from a Git repository.

### Create a Project from Scratch

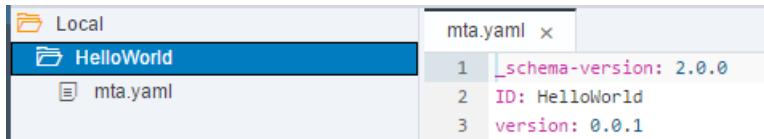
You create an MTA project using a dedicated template provided by SAP Web IDE.

#### Procedure

1. From the `Workspace` menu, choose  `New > Project from Template`.
2. Choose the `Multi-Target Application Project` template and click `Next`.

3. Enter a project name and click *Next*.
4. If needed, modify the *Application ID* and *Version* properties of the project. Both properties appear in the MTA descriptor (`mta.yaml` file).
5. Optional: enter a description of the project.
6. Once you have entered all the mandatory fields, click *Next* or *Finish*.

A new project with the specified name is created in your workspace. The project contains the initial MTA descriptor. For example:



```
mta.yaml x
1 schema-version: 2.0.0
2 ID: HelloWorld
3 version: 0.0.1
```

## Import a Project from an Archive

You can import to the SAP Web IDE workspace a multi-target application project that was previously exported to an archive.

### Prerequisites

A `.zip` archive of the project that you want to import should be available in the file system.

### Procedure

1. In the workspace, select the top-level `Local` folder.
2. Choose   .
3. Click *Browse* to locate and select your archived project file, and choose *Open*. The file name appears in the *File* field.

The destination folder name is displayed in the *Import to* field. By default, this name is the same as for the archive file. You can change the folder name, but not its location.

4. Choose *OK*. The project is created in the specified folder.

If an MTA descriptor (`mta.yaml` file), located in the project root folder, contains the definitions of modules, the corresponding subfolders are automatically converted into modules.

### Related Information

[Archive Import Troubleshooting \[page 421\]](#)

## Clone a Project from Git

You can clone an existing project from a Git repository.

### Procedure

Follow the instructions in [Clone Repositories \[page 291\]](#).

### Related Information

[Inside an MTA Descriptor \[page 355\]](#)

[Customizing Your Project \[page 66\]](#)

## Select a Cloud Foundry Space

You must select a Cloud Foundry space for running your MTA projects in SAP Cloud Platform.

### Context

You can configure a default Cloud Foundry space to be used for running your MTA projects in SAP Cloud Platform.

If you do not want to use the defined space for a specific project, you can select a different one in the  [Project Settings](#)  [Cloud Foundry](#) section.

### Procedure

1. From the [Preferences](#) perspective, choose [Cloud Foundry](#).
2. Select the [API Endpoint](#) and provide your user credentials in the dialog box that opens.
3. Select the [Organization](#) and [Space](#) from the respective dropdown lists.
4. In the [Builder](#) section, the button label indicates the status of builder in the selected space:
  - [Install Builder](#) means that there is no builder in the selected space. Click the button to install the builder.
  - [Reinstall Builder](#) means that there is already a builder in the selected space. Click the button if you experience any build problem, or if you want to update the builder.

## 11.2.3 Developing SAP HANA Database (HDB) Modules

An SAP HANA database (HDB) module is a collection of related design-time database artifacts, such as data models, views, or procedures.

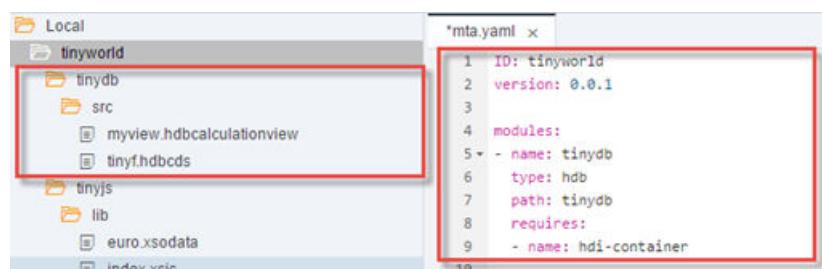
To develop and deploy these artifacts into the SAP HANA database, perform the following steps:



- [Create or Import an HDB Module \[page 363\]](#)
- [Develop Database Artifacts \[page 364\]](#)
- [Inside an MTA Descriptor \[page 355\]](#)
- [Build an HDB Module \[page 365\]](#)

### Module Folder Structure

The following figure depicts a sample HDB module folder structure alongside the corresponding entry in the `mta.yaml`.



| Folder        | Description  |
|---------------|--|
| <module name> | Should not contain any files.  |
| src (default) | The default location in which you store your design-time database artifacts. If needed, you can create additional subfolders for the same purpose. |

### Related Information

[Defining the Data Model in XS Advanced](#)

### 11.2.3.1 Create or Import an HDB Module

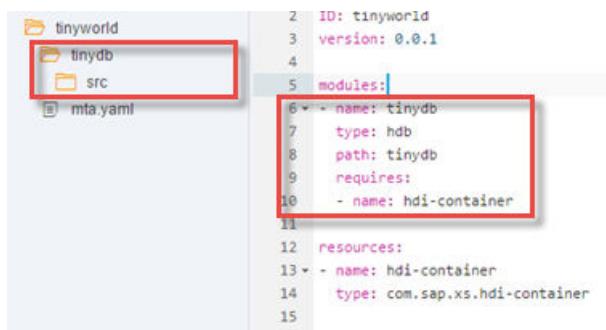
Create a new or import an archived SAP HANA database (HDB) module.

## Create a New Module

### Procedure

From the project context menu, choose **New > SAP HANA Database Module**, and follow the wizard steps to enter the module properties.

A new HDB module with the specified name is created in your project, and a corresponding section is added to the MTA descriptor (`mta.yaml`). For example:



```
2 ID: tinyworld
3 version: 0.0.1
4
5 modules:
6 - name: tinydb
7   type: hdb
8   path: tinydb
9   requires:
10  - name: hdi-container
11
12 resources:
13 - name: hdi-container
14   type: com.sap.xs.hdi-container
15
```

## Import a Module from an Archive

### Prerequisites

The `.zip` module archive that you want to import, which was exported from another MTA project, is available in the file system.

### Procedure

1. From the root folder of the project, choose **File > Import > From File System**.
2. Click **Browse** to locate and select your archive, and choose **Open**. The file name appears in the **File** field. The destination folder is displayed in the **Import to** field. To change this folder, choose **Select Folder**, and browse to the required folder, or create a new folder.  
The specified folder, containing the artifacts extracted from the archive, is created in the project.
3. To make the imported folder a proper module in your project, you need to convert it into a module of the matching type. From the folder context menu, choose **Convert To**, and then the type of the target module.

### **i** Note

The conversion process does not check whether the imported folder structure matches the selected module type. The process does not generate the module artifacts according to the selected type.

The imported module becomes a part of your MTA project, and the module entry is added to the MTA descriptor.

## 11.2.3.2 Develop Database Artifacts

Create the database artifacts required for your module.

### Procedure

From the context menu of the module's `src` subfolder, choose [New](#), and then choose one of the available artifacts.

### **i** Note

If you don't see the required artifact type in the menu, this means that the optional feature, which supports it, is disabled. To enable the relevant feature, follow the instructions in [Enable Additional Features \[page 392\]](#).

| Artifact  | Instructions   |
|---|--|
| <b>Database procedure (.hdbprocedure)</b>   | <p>Choose <a href="#">Procedure</a>, and enter the file name. The new artifact is added to the module, and opens in a dedicated code editor.</p> <p>For information about developing HDB procedures, see <a href="#">Defining the Data Model in XS Advanced</a>.</p>   |
| <b>Calculation view (.hdbscalculatedview)</b><br><b>A design-time definition of a calculation view, which can be referenced in an OData service definition.</b> | <p>Choose <a href="#">Calculation View</a>. Enter a name, label, and select the type and category. The new artifact is added to the module, and opens in the chosen editor.</p> <p>For information about developing calculation views, see <a href="#">SAP Cloud Platform Web IDE Multi-Cloud Version Modeling Guide</a>.</p>  |
| <b>CDS (Core Data Services) Document (.hdcds)</b>   | <p>Choose <a href="#">CDS Artifact</a>, enter a name, and choose a code snippet to add to the new file.</p> <p><b>i</b> Note<br/>The graphical editor only supports the <a href="#">Empty Context</a> snippet.</p> <p>The new artifact is added to the module, and opens in the default editor defined in  <a href="#">Tools</a> <a href="#">Preferences</a> <a href="#">Default Editor</a> <a href="#">HANA CDS Source</a>.</p> <p>For information about developing CDS documents, see:</p> <ul style="list-style-type: none"><li>○ In the text editor: <a href="#">Create a CDS Document (XS Advanced)</a></li></ul> |

| Artifact  | Instructions  |
|---|---|
|   | <ul style="list-style-type: none"> <li>○ In the graphical editor: <a href="#">Getting Started with the CDS Graphical Editor</a>.</li> </ul>   |
| <b>Analytic privilege (.hdbanalyticprivilege)</b> | <p>Choose <a href="#">Analytic Privilege</a>, and enter a name and label. The new artifact is added to the module, and opens in the dedicated editor.</p> <p>For information about developing analytic privileges, see <a href="#">Defining Data Access Privileges</a>.</p> |

#### i Note

Currently, SAP Web IDE provides dedicated editors only for the artifacts listed above. You can develop other artifacts, supported by SAP HANA XS Advanced (XSA), using a text editor. Create a file in the module's `src` subfolder with an appropriate extension, and open it in a text editor.

### 11.2.3.3 Build an HDB Module

The build process uses the design-time database artifacts to generate the corresponding actual objects in the SAP HANA database catalog.

#### Procedure

From the module context menu, choose *Build*.

The steps and messages of the build process are displayed in the console that opens automatically at the bottom of the browser window. You can show and hide the console by choosing ► [View](#) ► [Console](#) from the main menu.

#### i Note

Sometimes the build fails because there is an error in the MTA descriptor (`mta.yaml`). The error messages displayed in the console indicate the corrections you should make in the descriptor. For more information about the descriptor syntax, see [Inside an MTA Descriptor \[page 355\]](#).

After making the corrections, rebuild the module.

By default, the build process generates a database schema with a unique name derived from the `schema` parameter defined in `mta.yaml`. If you want the schema name to be exactly the same as defined, set the `makeUniqueName` parameter to `false`.

For example:

#### Sample Code

```
- name: hdi-container
  parameters:
    config:
      schema: mySpecialSchema
      makeUniqueName: false
  properties:
```

```
hdi-container-name: ${service-name}
type: com.sap.xs.hdi-container
```

However, if you do so and use the same schema name in different projects, naming conflicts might occur in the database during builds.

## Related Information

[Developing SAP HANA Database \(HDB\) Modules \[page 362\]](#)

### 11.2.4 Developing Java Modules

A Java module is a collection of related Java files and service definitions. Java modules implement the business logic of your application, either instead of or in addition to Node.js modules. A Java module can be either a Java Web Archive (WAR) or Java Archive (JAR) built with Apache Maven.

## Prerequisites

You have enabled the optional *Tools for Java Development* feature. For instructions, see [Enable Additional Features \[page 392\]](#).

## Workflow

To develop Java modules, perform the following steps:



- [Create or Import a Java Module \[page 368\]](#)
- [Implementing Java Files \[page 367\]](#)
- [Define Dependencies \[page 369\]](#)
- [Run and Test Java Modules \[page 370\]](#)
- [Debug Java Modules \[page 372\]](#)

## Module Folder Structure

The default files of a new Java module (except Web application with OData support) already contain a basic, ready-to-run "Hello World" servlet. The module is structured following the Apache Maven standard directory layout.

## Using a `settings.xml` File

If you decide to use a `settings.xml` file to provide project-specific Apache Maven settings, be aware that this file overrides default settings. During an update, changes to defaults do not overrule your `settings.xml`. Make sure you read the release notes to identify possible gaps between your `settings.xml` and the recommended default settings.

If needed, create the `settings.xml` next to the `pom.xml` on the same level. For more information about this optional step, see [Define Dependencies \[page 369\]](#)

## Implementing Java Files

In the `src/main/java` folder, create and implement the required Java files.

### Tip

While you are working on a Java file in a Java module, you can get assistance from the following features:

- **Code Assist**  
Provides code proposals as explained below.  
In a Java file, press `Ctrl` + `Space` for Java keyword artifact and local objects suggestions.
- **Code Validation**  
Provides code validation as described below.  
Problems are displayed in the file gutter. In addition, the problems view at the bottom of the screen displays complete problems in your Java module.

## Related Information

[Apache Maven - Introduction to the Standard Directory Layout](#) 

[The SAP HANA XS Advanced Java Run Time](#)

[Developing Multi-Target Applications \[page 352\]](#)

## 11.2.4.1 Create or Import a Java Module

You can create a new Java module or import an archived Java module.

### Create a New Module

#### Procedure

1. From the project context menu, choose **New** **Java Module**, and enter a module name.
2. Define your basic module settings. Depending on your application needs, choose one of the Java module templates:

| Option  | Description   |
|---|---|
| <a href="#">OData V4 Service Using SAP Cloud Platform SDK</a> | Creates a Java Web application module that exposes an OData V4 service. Can be used for extending SAP S/4HANA services.<br>The application is based on the SAP Cloud Platform SDK for service development, and has a standard Maven directory layout. |
| <a href="#">Multi-Module Web Application</a>                  | Produces * .war and * .jar files with dependencies.   |
| <a href="#">Simple Web Application</a>                        | Produces a * .war file that contains a basic ready-to-run "Hello World" servlet.  |
| <a href="#">Spring Boot Application</a>                       | Produces a * .jar file that brings its own runtime.   |
| <a href="#">Web Application with OData V2 Support</a>         | Produces a * .war file and provides an OData V2 endpoint.   |
| <a href="#">Web Application with OData V4 Support</a>         | Produces a * .war file and provides an OData V4 endpoint.   |

In addition, you can enable OData support to automatically expose annotated CDS models of your MTA project's HDB module as OData services.

3. Define other settings. If the default values don't fit your needs, change the Maven POM settings.

#### Related Information

[Maintaining OData Services in XS Advanced](#)

# Import a Module from an Archive

## Prerequisites

The `.zip` module archive that you want to import, which was exported from another MTA project, is available in the file system.

## Procedure

- From the root folder of the project, choose .
- Click *Browse* to locate and select your archive, and choose *Open*. The file name appears in the *File* field. The destination folder is displayed in the *Import to* field. To change this folder, choose *Select Folder*, and browse to the required folder, or create a new folder. The specified folder, containing the artifacts extracted from the archive, is created in the project.
- To make the imported folder a proper module in your project, you need to convert it into a module of the matching type. From the folder context menu, choose *Convert To*, and then the type of the target module.

### Note

The conversion process does not check whether the imported folder structure matches the selected module type. The process does not generate the module artifacts according to the selected type.

The imported module becomes a part of your MTA project, and the module entry is added to the MTA descriptor.

## 11.2.4.2 Define Dependencies

You can define optional dependencies of a Java module after creating it.

## Procedure

- Modify `pom.xml` to add dependencies as found on the Maven Web site.
- If your SAP Web IDE server cannot access the Internet, in particular `https://repo1.maven.org/maven2/`, you can set up your own Maven repository manager. Such custom repository setups can be configured by adding a `settings.xml` file to your Java module.
- To identify outdated dependencies, select your Java module and go to . Use the *Problems* view to see outdated dependencies.

## Related Information

[Create or Import a Java Module \[page 368\]](#)

[Run and Test Java Modules \[page 370\]](#)

[Inside an MTA Descriptor \[page 355\]](#)

<http://search.maven.org/>

<https://maven.apache.org/settings.html>

<https://maven.apache.org/repository-management.html>

<https://repo1.maven.org/maven2/>

### 11.2.4.3 Run and Test Java Modules

You can run and test your Java modules using the tools provided by SAP Web IDE.

#### Prerequisites

Before running a module, make sure that:

- All the relevant dependencies of the module are defined in the MTA descriptor.
- All the modules and resources on which the module is dependent are implemented and available (HDB modules are built).

#### i Note

You don't need to build your module explicitly before running, because it will be built automatically during the run process. The result of that build is then available in your workspace in the `target` folder. The result is either a `.war` or `.jar` file depending on the type of your Java module.

#### Run a Module

To create a running application in the runtime system, and resolve all dependencies defined in `mta.yaml`, run each Java module in your project. From the module context menu, choose [Run](#), and one of the following options:

| Option  | Description   |
|---|---|
| <a href="#">Run Configurations</a>  | <p>Create a new or modify an existing run configuration.</p> <p>In the <i>Browser Window</i> tab, you can optionally choose a service path for your module to open in a new browser window. For example, choose <code>/hello</code> for the Hello World sample servlet generated by the wizard.</p> |
|  <a href="#">Run as Java Application</a> | Run the .war or .jar archive that was last built by Maven.  |

## Use the Run Console

Progress and status messages, generated during the run process, as well as the log records, are displayed in the Run console. The console opens automatically in the main browser tab. You can show and hide the console by clicking the  (*run console*) icon in the bottom pane's toolbar.

The left pane of the console displays a list of all running modules in the current MTA project along with the corresponding run configurations. Each entry is accompanied by a status icon with a tooltip. Click the run configuration name below a module name to view its most recent run log.

The controls displayed at the top of the console enable you to perform the following tasks:

- If the application URL is visible, click it to access the running application in the browser.
  - To view the detailed server log of a run process, click [Logs](#).
  - To clear the log in the console, click  (*clear the log*).
  - To stop running the module, click  (*stop*).
- This causes the module to be removed from the runtime system, so that if you run the module again, the run process starts from the beginning.
- To run the module using a different configuration, select the configuration and choose the  (*run*) button in the task bar.

## Test a Module

Build and test your Java application with unit tests.

- Add unit tests in the `test` folder structure.
- Execute your tests. In the context menu from your module select [Build and Run Tests](#).
- Review the test results.

Each Java module must provide own its JUnit test. The Java module template comes with a sample JUnit test. Use this test to get familiar with the test functionality and the  (*test results*) pane. By default, the  (*test results*) pane shows [Test Results](#), [Stack Trace](#), and [Coverage](#). Also by default, the code coverage is shown in the editor, which enables you to write targeted unit tests for uncovered code. To disable this function, edit your  (*settings*) .

You can also use the  (*test results*) pane to  (*export*) or  (*delete*) all test results for the selected module. Your workspace is connected to your  (*test results*) pane. The results you see, are for the currently selected module,

which is also referenced in the pane as the path to your current module (/<projectname>/<modulename>). If you miss the latest test result, select  (refresh). Also check the logs to see if your test execution failed. If it has, no test result is available.

## Related Information

[Debug Java Modules \[page 372\]](#)

### 11.2.4.4 Debug Java Modules

You can debug your Java modules using the tools in the Debugger pane.

If you have enabled debugging in the run configuration, and set breakpoints in your .java files, the  (Debugger) panel will open as soon as the module is up and running. In this pane, you can perform regular debugging tasks, such as viewing the call stack, examining the variables, stepping in and out of the functions, and more.

#### Note

If you create a new run configuration, and the *Run in debug mode* checkbox under *Debug* is disabled, check if your module's root pom.xml. It should include the correct development mode (`devmode`) Maven profile configuration, which is required in order to attach the source files to the build.

If not, please add the following configuration to the file.

```
<profiles>
    <profile>
        <activation>
            <property>
                <name>devmode</name>
                <value>true</value>
            </property>
        </activation>
        <build>
            <plugins>
                <plugin>
                    <artifactId>maven-war-plugin</artifactId>
                    <configuration>
                        <webResources>
                            <resource>
                                <directory>${project.build.sourceDirectory}</directory>
                                <targetPath>sources</targetPath>
                            </resource>
                        </webResources>
                    </configuration>
                </plugin>
            </plugins>
        </build>
    </profile>
</profiles>
```

## Breakpoints

| Breakpoint Status            | Description   |
|------------------------------|---|
| Enabled standard breakpoint  | Shape outline: blue<br>Shape fill: light blue   |
| Disabled standard breakpoint | Shape outline: blue<br>Shape fill: transparent<br><br>Breakpoints have been explicitly disabled. Disable breakpoints in the <i>Breakpoints</i> section by selecting the corresponding checkbox or deactivate globally by pressing the  ( <i>deactivate breakpoints</i> ) button.   |
| Checkmark                    | Shape outline and fill are not affected.<br><br>You set the breakpoint successfully on a running debugging session. The checkmark indicates that the corresponding source file has been loaded. As a consequence the application will suspend when it reaches the breakpoint.<br><br>A breakpoint without a checkmark can mean that there is no debug session running, the source file has not been loaded, or the breakpoint could not be successfully resolved. An unresolved breakpoint may occur when the workspace resource version and the currently executed resource version don't match. |

## Related Information

[Run and Test Java Modules \[page 370\]](#)

### 11.2.4.5 Use Eclipse to Develop Java Modules

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

## In SAP Web IDE

| Task                      | Information  |
|---------------------------|--|
| 1. Create an MTA project. | <a href="#">Setting Up Application Projects [page 359]</a> |

| Task  | Information   |
|---|---|
| 2. To generate the data objects in the database, develop and build an SAP HANA Database module. | <a href="#">Developing SAP HANA Database (HDB) Modules [page 362]</a> |
| 3. Create a Java module.  | <a href="#">Developing Java Modules [page 366]</a>                    |
| 4. Submit the module to a Git repository, or export it to the file system.                      | <a href="#">Using Source Control (Git) [page 283]</a>                 |
| 5. Switch to Eclipse IDE.   |   |

## In Eclipse IDE

| Task   | Information   |
|--|---|
| 1. Clone the Java module from the Git repository.            | <p>Choose  <a href="#">Window</a> <a href="#">Show View</a> <a href="#">Git Repositories</a> <a href="#">Clone a Git Repository</a> , and add the clone to the view.</p> <p>Alternatively, you can import the exported module from the file system.</p>   |
| 2. Import the Java project into Eclipse.                     | <ol style="list-style-type: none"> <li>Choose  <a href="#">Window</a> <a href="#">Show View</a> <a href="#">Project Explorer</a> <a href="#">Import</a> <a href="#">Existing Maven Projects</a> .</li> <li>In <a href="#">Root Directory</a>, enter the folder to which you have cloned the project, and click <a href="#">Finish</a>.</li> </ol>   |
| 3. Write Java code, then build and run the Java application. | <ol style="list-style-type: none"> <li>From the project context menu, choose  <a href="#">Run As</a> <a href="#">Maven install</a> .</li> <li>Choose  <a href="#">Run As</a> <a href="#">Run on Server</a> .</li> </ol>   |
| 4. Deploy the application to Cloud Foundry.                  | <ol style="list-style-type: none"> <li>In the <a href="#">Select the Server</a> dropdown list, either choose an existing server, and then  <a href="#">Cloud</a> <a href="#">Cloud Foundry</a> , or define the server manually.</li> <li>In the <a href="#">Add and Remove</a> dialog box, add the required resources to the <a href="#">Configured</a> list, and click <a href="#">Finish</a>.</li> <li>In the <a href="#">Application Details</a> dialog box, make sure that the details of your application are correct, and click <a href="#">Next</a>.</li> <li>In the <a href="#">Launch Deployment</a> dialog box, enter the deployment details, and click <a href="#">Next</a>.</li> <li>Bind a service to the application. In the <a href="#">Service selection</a> dialog box, select the <code>hdi</code> container of the HDB module that you previously built in SAP Web IDE. The name has the following pattern: <code>&lt;project_name&gt;-hdihdb-&lt;userid&gt;&lt;workspace&gt;</code>. Click <a href="#">Next</a>.</li> </ol> |

| Task  | Information  |
|---|--|
| 5. Add environment variable PATH that points to the WAR file. | <p>Add the following parameters to the <code>manifest.yml</code> file:</p> <ul style="list-style-type: none"> <li>• path that points to the WAR file.</li> <li>• In the env section, <code>JBP_CONFIG_RESOURCE_CONFIGURATION</code></li> </ul> <p>For example:</p> <pre>... path: /Users/&lt;UserID&gt;/workspace/mtad4/java4/target/ java4-0.0.1-SNAPSHOT.war env:   JBP_CONFIG_RESOURCE_CONFIGURATION: '[tomcat/ webapps/ROOT/META-INF/context.xml:   {"service_name_for_DefaultDB" : "&lt;name of the hdi-container&gt;"}]'</pre> |
| 6. Submit the Java project to the same Git repository.        |  |
| 7. Switch to SAP Web IDE.                                     |  |

## In SAP Web IDE

| Task   | Information   |
|--|---|
| 1. Clone the Java module from the Git repository, and import it to your MTA project. | <a href="#">Create or Import a Java Module [page 368]</a>                             |
| 2. Develop an HTML5 module as required.  | <a href="#">Developing HTML5 Modules [page 375]</a>                                   |
| 3. Run and test the application, then deploy it to Cloud Foundry.                    | <a href="#">Packaging and Deploying Applications to Production Systems [page 385]</a> |

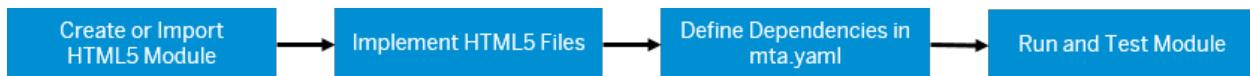
### 11.2.5 Developing HTML5 Modules

An HTML5 module is a collection of related HTML5 files that implement the user interface of your application.

SAP Web IDE supports the following HTML5 module types:

- SAPUI5 application - an SAPUI5 app including an optional view.
- CRUD master-detail module - displays data from an OData service using the master-detail pattern, provides CRUD operations.
- List Report application - based on the list report and object page SAP Fiori elements.

To develop HTML5 modules, perform the following steps:



- [Create or Import an HTML5 Module \[page 377\]](#)
- [Implementing HTML5 Files \[page 376\]](#)
- [Inside an MTA Descriptor \[page 355\]](#)
- [Running Applications in Development Mode \[page 311\]](#)

## Module Folder Structure

The following figure depicts a sample basic HTML5 module folder structure alongside the corresponding entry in the `mta.yaml`.

The screenshot shows a file browser on the left and a code editor on the right. The file browser displays a folder structure with a red box highlighting the 'tinyui' folder containing 'resources', 'index.html', 'package.json', and 'xs-app.json'. Below it is 'mta.yaml'. The code editor shows the contents of 'mta.yaml' with a red box highlighting the 'tinyui' service definition. The code is as follows:

```
17   - name: hdi-container
18   #----- exposes SERVICE URL to consumers
19   provides:
20   - name: tinyjs_api
21   properties:
22     | service_url: ${default-url}
23   #
24   | name: tinyui
25   type: html5
26   path: tinyui
27   # -- requires tinyjs service URL
28   requires:
29   - name: tinyjs_api
30   group: destinations
31   properties:
32     | name: tinyjs_url
33     | url: ~{service_url}
```

| Folder                           | Description  |
|----------------------------------|--|
| <code>&lt;module name&gt;</code> | Contains the <code>xs-app.json</code> (application router configuration), and <code>package.json</code> (application router details and dependencies) files. |
| <code>resources</code>           | Contains the HTML5 resource files, including the default <code>index.html</code> file.   |

### ⚠ Caution

Do not change the version of `approuter` in the `package.json` of the module.

## Implementing HTML5 Files

In the `\resources` subfolder, create and implement the required HTML5 files. You can design the XML views using the SAP Web IDE layout editor.

For information about developing HTML5 apps with SAPUI5, see [SAPUI5: UI Development Toolkit for HTML5](#).

#### [Create or Import an HTML5 Module \[page 377\]](#)

You can create a new or import an archived HTML5 module.

## Related Information

[Layout Editor \[page 207\]](#)

[Developing Multi-Target Applications \[page 352\]](#)

[Create List Report and Object Page Applications \[page 199\]](#)

### 11.2.5.1 Create or Import an HTML5 Module

You can create a new or import an archived HTML5 module.

#### [Define Data Connections \[page 378\]](#)

You need to define connections to the data sources that you want to access from your application.

#### [Create Destinations for Cloud Foundry Services \[page 379\]](#)

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

## Create a New Module

### Procedure

1. From the project context menu, choose ► **New** ► **HTML5 Module** ▶.
2. Depending on your application needs, choose one of the HTML5 module templates.
3. Enter the module name, and set the relevant module properties.
4. In the *Data Connection* wizard step, define a connection to the appropriate data source. For more information, see [Define Data Connections \[page 378\]](#).

A new HTML5 module with the specified name is created in your project, and a corresponding section is added to the MTA descriptor (`mta.yaml`). For example:

```
version: 0.0.1
modules:
- name: tinydb
  type: hdb
  path: tinydb
  requires:
- name: hdi-container
- name: tinyjs
  type: nodejs
  path: tinyjs
- name: tinyui5
  type: html5
  path: tinyui5
```

### Note

The default `index.html` file of the new module implements a basic Hello World application.

## Import a Module from an Archive

### Prerequisites

The `.zip` module archive that you want to import, which was exported from another MTA project, is available in the file system.

### Procedure

1. From the root folder of the project, choose   .
2. Click *Browse* to locate and select your archive, and choose *Open*. The file name appears in the *File* field.  
The destination folder is displayed in the *Import to* field. To change this folder, choose *Select Folder*, and browse to the required folder, or create a new folder.  
The specified folder, containing the artifacts extracted from the archive, is created in the project.
3. To make the imported folder a proper module in your project, you need to convert it into a module of the matching type. From the folder context menu, choose *Convert To*, and then the type of the target module.

### Note

The conversion process does not check whether the imported folder structure matches the selected module type. The process does not generate the module artifacts according to the selected type.

The imported module becomes a part of your MTA project, and the module entry is added to the MTA descriptor.

### 11.2.5.1.1 Define Data Connections

You need to define connections to the data sources that you want to access from your application.

The following table describes how to define connections to the available data source types.

| Data Source Type | Instructions   |
|------------------|--|
| Current Project  | Browse your project to locate and select a Java or Node.js module that exposes an OData service. |

| Data Source Type     | Instructions  |
|----------------------|---|
| File System          | Browse your file system to locate and select the required service <code>metadata.xml</code> document.                                     |
| Workspace            | Browse your workspace to locate and select the required service <code>metadata.xml</code> document.                                       |
| Service Catalog      | From the dropdown list, select a system to connect to the required service, and select this service from the <code>Services</code> table. |
| Service URL          | From the dropdown list, select a system to connect to the required service, and enter the relative URL of this service.                   |
| SAP API Business Hub | From the dropdown list, select an API package containing the required service, and select this service from the table below.              |

When you select a remote service, you are prompted to provide credentials for the requested system.

## 11.2.5.1.2 Create Destinations for Cloud Foundry Services

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

### Context

To create a destination within the wizard, the Cloud Foundry service must already be deployed and running.

Once you create the destination with the wizard, the service is available in the list of services under `Service URL`.

### Procedure

- In the `Data Connection` step of the HTML5 Module wizard, select `configure a new destination`.
- Fill out the following fields:

| Field                | Description   |
|----------------------|---|
| Destination Name     | A name for the destination  |
| Global Subaccount ID | The ID of your global subaccount that contains your Neo environment |

| Field                      | Description   |
|----------------------------|---|
| Global Subaccount Password | The password of your global subaccount that contains your Neo environment |
| Cloud Foundry Endpoint     | The endpoint you received when signing up for a Cloud Foundry environment |
| Service Name               | The name of your Cloud Foundry service                                    |
| Cloud Foundry Org          | The organization in which your Cloud Foundry service is running           |
| Cloud Foundry Space        | The space in which your Cloud Foundry service is running                  |

3. Select *OK*.

## Results

The destination is now available in the list of services under *Service URL*.

### 11.2.6 Developing SAP S/4HANA Service Extensions

SAP Web IDE enables you to create an OData V4 service that extends an existing S/4HANA service. The new service exposes additional data from a different source, such as an SAP HANA database.

You can do this using the dedicated SAP S/4HANA Service Extension Tools.

## Prerequisites

You have enabled the following additional SAP Web IDE features:

- SAP S/4 HANA Extension Tools
- SAP HANA Database Explorer
- Tools for SAP Cloud Platform App Development (Experimental)- only in the trial environment

For more information, see [Enable Additional Features \[page 392\]](#).

## Workflow

- 
1. [Create an SAP S/4HANA Service Extension Project \[page 381\]](#)  
You can use the dedicated SAP S/4HANA Service Extension project template to create an extension of an existing SAP S/4HANA service.
  2. [Create an Extended Data Model \[page 382\]](#)  
You define an extended data model for your service, and create the corresponding data objects.
  3. [Configure Access to the Source Service \[page 383\]](#)  
To enable your application to access the source SAP S/4HANA service, configure the destination in the MTA descriptor.
  4. [Implement the Java Module \[page 384\]](#)  
In the Java module you expose the OData V4 service interface and implement the CRUD operations on the exposed data entities.
  5. [Build and Deploy the Service to Cloud Foundry \[page 385\]](#)  
You can build and deploy your service to Cloud Foundry directly from SAP Web IDE.

## Related Information

[SAP Cloud Platform SDK for service development](#)

### 11.2.6.1 Create an SAP S/4HANA Service Extension Project

You can use the dedicated SAP S/4HANA Service Extension project template to create an extension of an existing SAP S/4HANA service.

## Context

This is an MTA (multi-target application) project template. You can use it to create an OData V4 service that exposes data from an existing SAP S/4HANA service extended by custom data. You define and store the custom data in an SAP HANA database.

## Procedure

From the *Workspace* menu, choose  [New > Project from Template](#), and choose the *SAP S/4HANA Service Extension* template.

Follow the steps described in [Setting Up Application Projects \[page 359\]](#).

## Results

A new MTA project is created in your workspace. It includes the initial HDB (SAP HANA Database), and Java modules. In the trial environment it also includes a CDS (Core Data Services) module.

Each module has a role in the project:

- The CDS module is used to define the extended data model of the new service.
- The HDB module is used to generate the corresponding data objects in the SAP HANA database.
- The Java module is used to implement the service interface.

**Task overview:** [Developing SAP S/4HANA Service Extensions \[page 380\]](#)

**Next task:** [Create an Extended Data Model \[page 382\]](#)

## Related Information

[Create an OData V4 Service That Exposes Data From Multiple Data Sources](#)

### 11.2.6.2 Create an Extended Data Model

You define an extended data model for your service, and create the corresponding data objects.

#### Context

The database objects (tables) and service metadata will be created from this model.

#### Procedure

1. Define your data model. It should include the entities exposed by the source service, and new entities that extend it. The actual steps to perform depend on your SAP Web IDE environment.

##### In the productive environment:

1. Create an `.hdbc` table definition file under `hdb > src`. In this file, define only the new extending data entities.
2. Create a `<service name>.xml` file under `java > src > main > resources > edmx`. In this file, define the service metadata according to the OData V4 protocol.

##### In the trial environment:

1. In the `cds` folder of your project, create the `src` subfolder. In this subfolder, create a `.cds` file, open it in the editor and define the data entities for your service.
2. From the `cds` module context menu, choose *Build*. As a result, the `.hdbc` table definition file and the service metadata `<service name>.xml` file are generated.

2. From the `hdb` module context menu, choose *Build*.

The new tables are generated in your SAP HANA database according to the `.hdbc` file.

**Task overview:** [Developing SAP S/4HANA Service Extensions \[page 380\]](#)

**Previous task:** [Create an SAP S/4HANA Service Extension Project \[page 381\]](#)

**Next task:** [Configure Access to the Source Service \[page 383\]](#)

## 11.2.6.3 Configure Access to the Source Service

To enable your application to access the source SAP S/4HANA service, configure the destination in the MTA descriptor.

### Context

You need to specify the URL and user credentials for the service in the `destinations` property of the `java` module in the MTA descriptor (`mta.yaml` file).

### Procedure

Open `mta.yaml` in the MTA or code editor, and enter the URL, user name, and password in the `destinations` property of the `java` module.

For example:

```
name: java
type: java
path: java
..
requires:
- name: hdi_hdb
properties:
  ..
  destinations: '[{"name": "S4System", "url": "https://s4hana.example.com", "username": "DEVUSER", "password": "!@#$%%^^"}]'
```

---

**Task overview:** [Developing SAP S/4HANA Service Extensions \[page 380\]](#)

**Previous task:** [Create an Extended Data Model \[page 382\]](#)

**Next task:** [Implement the Java Module \[page 384\]](#)

## 11.2.6.4 Implement the Java Module

In the Java module you expose the OData V4 service interface and implement the CRUD operations on the exposed data entities.

### Context

### Procedure

1. Expand the `src` subfolder of the `java` module, and create one or more Java classes under `java`.
2. Implement the required interface and operations with the help of the SAP Cloud Platform SDK for service development.
3. From the Java module context menu, choose **Run As > Java Application**. Once the application opens in a new browser tab, run some OData queries by adding the suffix `/odata/v4/<service-name>/` to the application URL. Test that the service works as expected.

**Task overview:** [Developing SAP S/4HANA Service Extensions \[page 380\]](#)

**Previous task:** [Configure Access to the Source Service \[page 383\]](#)

**Next task:** [Build and Deploy the Service to Cloud Foundry \[page 385\]](#)

### Related Information

[Define the Service Class](#)

[Run and Test Java Modules \[page 370\]](#)

## 11.2.6.5 Build and Deploy the Service to Cloud Foundry

You can build and deploy your service to Cloud Foundry directly from SAP Web IDE.

### Procedure

1. From the project context menu, choose *Build*.  
An `.MTAR` (MTA archive) file is created under `mta_archives/`.
2. From the archive context menu, choose *Deploy* *Deploy to Cloud Foundry*.
3. Once the service is deployed, you can test it in the SAP Cloud Platform cockpit.

**Task overview:** [Developing SAP S/4HANA Service Extensions \[page 380\]](#)

**Previous task:** [Implement the Java Module \[page 384\]](#)

### Related Information

[Packaging and Deploying Applications to Production Systems \[page 385\]](#)

[Test the OData V4 Service](#)

## 11.2.7 Packaging and Deploying Applications to Production Systems

At the last stage of multi-target application (MTA) development, you need to package your application and deploy it to a target production system.

### Prerequisites

You have selected a space to build your application. For more information, see [Setting Up Application Projects \[page 359\]](#) -> *Selecting a Space for a Project*.

### Context

If your target production system is SAP Cloud Platform, Cloud Foundry environment, perform the following steps:

## Procedure

1. From your project context menu, choose *Build*.
2. Expand the *mta\_archives* folder to locate your project's build archive (<project\_name-version>.mtar file).
3. From the archive context menu, choose ► *Deploy* ► *Deploy to Cloud Foundry* ▾.
4. In the *Deploy to Cloud Foundry* dialog box that opens, select the space in the target production system where you want to deploy your application, and click *Deploy*.

### 11.2.8 Developing Applications Using the SAP Cloud Platform Programming Model

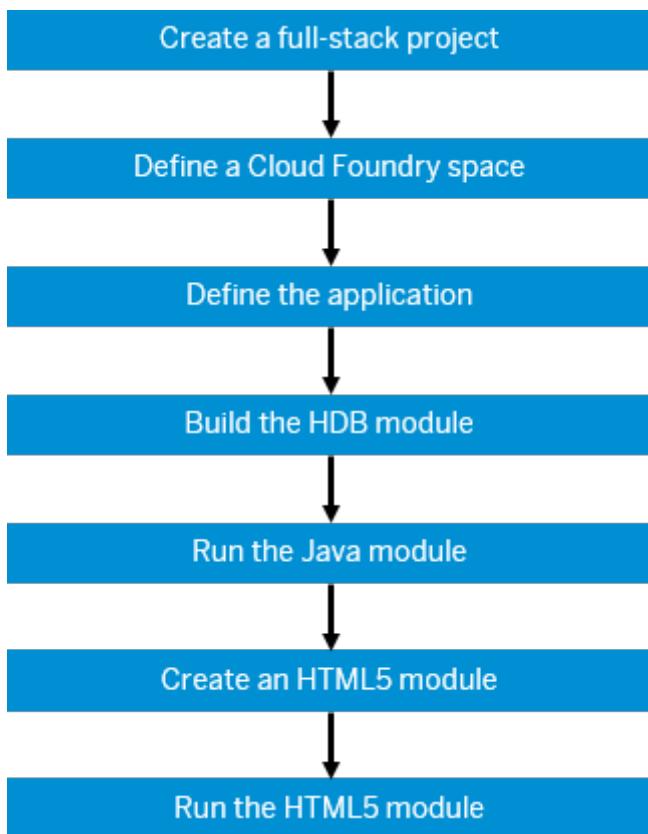
You can create a full-stack application with CRUD capabilities in SAP Web IDE using the SAP Cloud Platform programming model (experimental).

#### i Note

This is an experimental feature.

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 at an early point of time allowing customers and partners to influence the future product accordingly. To provide feedback, use the [SAP Community](#). By providing your feedback, you accept that Intellectual Property rights of the contributions or derivative works shall remain the exclusive property of SAP.



- [#unique\\_306/unique\\_306\\_Connect\\_42\\_subsection-im1 \[page 387\]](#)
- [#unique\\_306/unique\\_306\\_Connect\\_42\\_subsection-im2 \[page 388\]](#)
- [#unique\\_306/unique\\_306\\_Connect\\_42\\_subsection-im3 \[page 388\]](#)
- [#unique\\_306/unique\\_306\\_Connect\\_42\\_subsection-im4 \[page 388\]](#)
- [#unique\\_306/unique\\_306\\_Connect\\_42\\_subsection-im5 \[page 388\]](#)
- [#unique\\_306/unique\\_306\\_Connect\\_42\\_subsection-im6 \[page 389\]](#)
- [#unique\\_306/unique\\_306\\_Connect\\_42\\_subsection-im7 \[page 389\]](#)

Click the steps for more information.

### Create a Full-Stack Project

1. Enable the *Tools for SAP Cloud Platform App Development (Experimental)* feature. See [Enable Additional Features \[page 392\]](#).
2. Select *File* > *New* > *Project from Template* .
3. Select the *Full-Stack Application for Cloud Foundry* template and click *Next*.
4. Follow the wizard and fill in the required fields.

Your new project is created in your Workspace. It contains the following modules:

- CDS
- HDB

- Java

These modules contain no Java code or SAP HANA database (HDB) artifacts.

## Define a Cloud Foundry Space

In order to build and run your application, you need to define a Cloud Foundry space for your project.

1. In your workspace, right-click your project folder and select *Project Settings*.
2. In the *Space* settings, provide your Cloud Foundry organization and space details. You will be able to use this space for development needs such as running the Java service. The organization and space are also required to install a builder.
3. If there is no builder in your space, click *Install Builder*.
4. Save your changes.

## Define the Application

In the CDS module, you can define the application models. The application models consist of the data model, service model, and if needed, the UI model.

You can define the application models manually using the CDS editor. In this example, we are using the models from the *Manage Products for Cloud Foundry* sample application.

1. Go to the *cds* module in your project and create new CDS artifacts.
2. Build the *cds* module.

The CDS build adds the following artifacts to the project:

- In HDB module, the hdbcds artifacts were created.
- In the Java module, the service metadata files and the csn.json artifacts were created.

## Build the HDB Module

You need to build the HDB module to create the SAP HANA DB schema.

1. Right-click the *hdb* module in your application and select *Build*. The schema and tables are created in the SAP HANA database.
2. (Optional) You can open the SAP HANA database explorer to view information about your database's catalog objects and execute queries.



1. Open the DBX perspective (  ) from the left-side pane to see the created schemas in SAP HANA.
2. Select your db schema and tables.
3. Check that the tables were added, and click *Open Data* to see the data.
4. Go back to the development perspective to continue developing your app.

3. For more information, see [SAP HANA Database Explorer and SQL Analyzer](#).

## Run the Java Module

The Java module is currently empty except for the *edmx* folder, which contains the service metadata that was generated from the build of the CDS module. You need to build, run, and test the Java module to expose the OData Service.

1. Right-click the Java module and select   .
2. Once the application is running, go to the Run console.
3. Click the application URL and add /odata/v2 to the URL to see the OData service.

## Create an HTML5 Module

Create a new HTML5 module in your project. For more information about HTML5 modules, see [Developing HTML5 Modules \[page 375\]](#).

1. Right-click your project and select **New > HTML5 Module**.
2. Select the desired template and fill in the required fields in the wizard.
3. In the *Data Connection* step of the wizard, in the *Sources* list, select *Current Project* so that you can connect between the Java module and the HTML5 module. In the *Service* list in the table, select the service you want to use for the module.
4. Continue with the wizard according to the template you selected.
5. Choose *Finish*.

A new HTML5 module is added to your project.

## Run the HTML5 Module

1. Select the HTML5 module folder and click *Run*.
2. In the *Choose the file to Run* window, select the file you want to run.
3. Because the preview is done on the Neo environment, you need to configure a destination to consume the service that you just created.

### Note

If you do not have a destination, you will be prompted for your Neo credentials to create one from within SAP Web IDE.

- The SAP Fiori launchpad sandbox opens.
4. Select your app tile.  
The app data is displayed.
  5. If needed, you can edit the data.

# 12 Additional Features

You can enable additional features in SAP Web IDE to extend the existing functionality.

## i Note

This is not available in SAP Web IDE personal edition.

Features are the building blocks of SAP Web IDE. They are used to group functionality into small units.

Features can expose services to provide public APIs. Feature code may include any SAP Web IDE component: a new command, template, editor, pane, or any other contribution to the SAP Web IDE application.

The functionality provided by these additional features can be enabled or disabled from the Preferences perspective. For more information, see [Enable Additional Features \[page 392\]](#).

The list below shows a selection of our most used features.

### SAP Web IDE Additional Features

| Feature                           | 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 <a href="#">SAP Web IDE Hybrid App Toolkit Add-on for Cloud Deployments</a>.</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 <a href="#">Extend Prototypes with SAP Web IDE</a></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 <a href="#">SAP Cloud Platform Portal service</a>.</p>   |
| IoT Application Enablement        | <p>You can build new IoT-related applications and customize them by using predefined components and templates.</p> <p><b>i Note</b></p> <p>In order to use the IoT project templates, you must first subscribe to the IoT Application Enablement Services.</p> <p>See <i>IoT Application Enablement Reuse Components and Templates</i> in the <a href="#">IoT Application Enablement</a> page.</p> |

| Feature                             | Description   |
|-------------------------------------|---|
| SAP Cloud Platform Workflow         | <p>You can model and deploy workflows that help in automating process steps.</p> <p><b>i Note</b></p> <p>The Workflow editor is available only in regions where the SAP Cloud Platform Workflow service is offered.</p> <p>See <a href="#">SAP Cloud Platform Workflow</a>.</p> |
| SAP Enterprise App Modeler          | <p>You can develop next generation App Modeler applications.</p> <p>See <a href="#">SAP Enterprise App Modeler</a></p>  |
| SAP HANA Database Explorer          | <p>The database explorer is integrated into SAP Web IDE and allows you to execute SQL statements and database procedures, query information about the database, and view information about database catalog objects.</p> <p>See <a href="#">SAP HANA Database Explorer</a></p>  |
| SAP Fiori Launchpad Extensibility   | <p>You can extend the shell of the SAP Fiori launchpad by creating your own SAP Fiori launchpad plugins.</p> <p>See <a href="#">SAP Fiori Launchpad Extensibility</a></p>   |
| Visualization Extension (Vizpacker) | <p>You can use the Visualization Extension plugin (Vizpacker) in SAP Web IDE to create chart extension packages that can be used within SAP Lumira and other products.</p> <p>See <a href="#">Visualization Extension (VizPacker) Plugin for SAP Web IDE</a></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><b>i Note</b></p> <p>This feature has been deprecated.</p> <p>See <a href="#">Fact Sheet Editor</a>.</p>  |

### i Note

If you want to use the same class for several features, you must:

1. Define the desired class as an SAP Web IDE service for the first feature.
2. Consume the class as a service in the other features.

## 12.1 Enable Additional Features

You can enable additional SAP Web IDE features to use in application development.

### Context

SAP Web IDE includes features that are not enabled by default. If you would like to use additional features, perform the following steps:

### Procedure

1. To open the Preferences perspective, in the left sidebar, choose  (Preferences).
2. Choose *Features*.
3. Select the toggle button for the feature you want to enable.
4. Choose *Save*.
5. Refresh your browser.

### Results

You can use the enabled features in your projects.

# 13 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](#).

# 14 SAP Web IDE Personal Edition

Customers using SAP Web IDE on SAP Cloud Platform 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 offline development (not on a server).

You can use it in one of the following ways:

- Trial - for test and evaluation purposes for anyone.
- Productive - for all customers who have a license to productively use SAP Web IDE on SAP Cloud Platform.

## Note

The personal edition is updated periodically and may not include features of SAP Web IDE on SAP Cloud Platform.

The personal edition 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 (or from the personal edition to the cloud edition), you can use Git or you can export the relevant project and import it in the other edition.

The following features are **not** available if you are using SAP Web IDE personal edition:

- SAP Cloud features:
  - Deploying to SAP 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

## 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
- Grunt build
- Translation Hub

## 14.1 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 \[page 395\]](#)

[Start SAP Web IDE Personal Edition \[page 399\]](#)

[Upgrade SAP Web IDE Personal Edition \[page 401\]](#)

### 14.1.1 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 higher 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<sup>®</sup>) or Terminal window (Mac OS<sup>®</sup>).

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)
```

## 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 <sup>®</sup>   | Mac OS <sup>®</sup>  |
|--|--|
| Extract the zipped files to C:\\SAPWebIDE.<br><br><b>i</b> Note <ul style="list-style-type: none"><li>○ 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.</li><li>○ Your workspace will be saved in the serverworkspace folder created. Be sure to back up this folder regularly.</li></ul> | Extract the zipped files to /Applications/<br>SAPWebIDE.<br><br><b>i</b> Note <ul style="list-style-type: none"><li>If you have installed Mac OS™ Sierra, you need to perform some post-installation steps. For more information, see <a href="#">Post Installation Troubleshooting for Mac Sierra Users [page 397]</a>.</li></ul> |

5. Start SAP Web IDE personal edition. For more information, see [Start SAP Web IDE Personal Edition \[page 399\]](#).

#### [Post Installation Troubleshooting for Mac Sierra Users \[page 397\]](#)

Steps required to run SAP Web IDE personal edition on Mac Sierra systems.

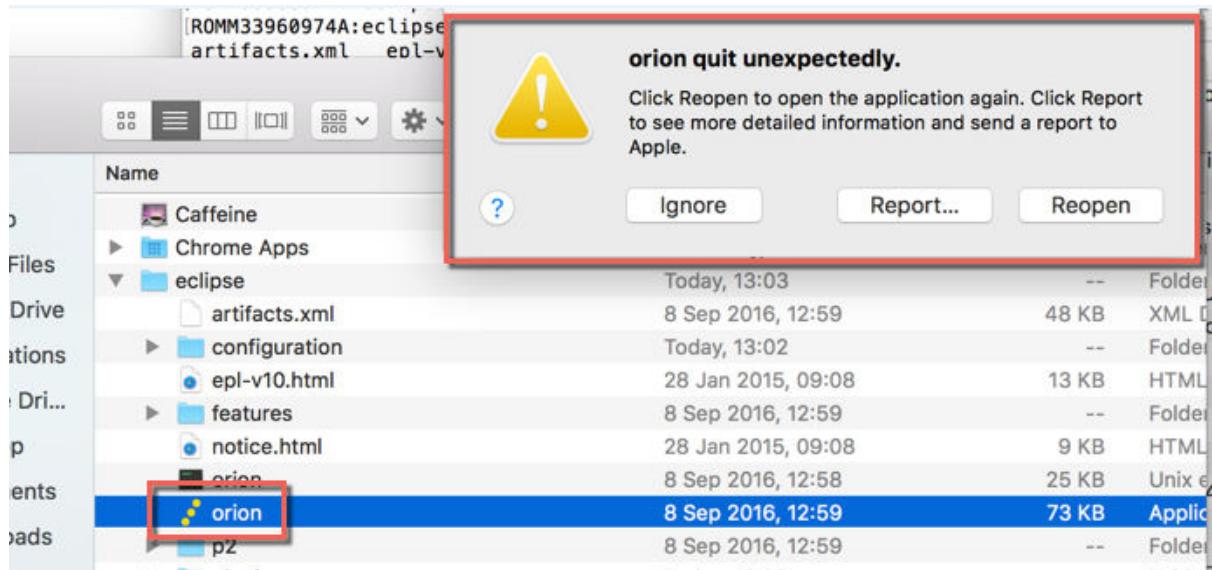
## 14.1.1.1 Post Installation Troubleshooting for Mac Sierra Users

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](#).

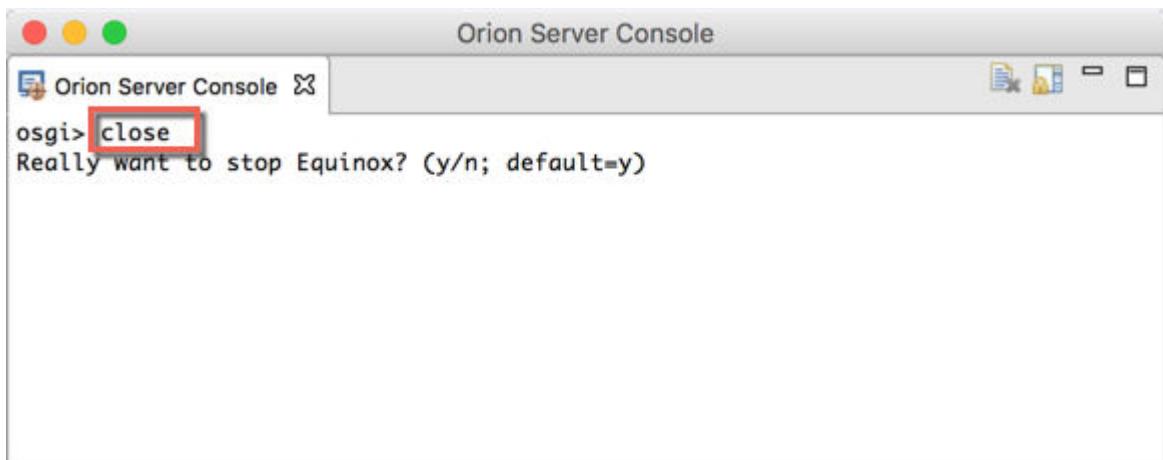
### Solution for closing using the command line

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 \[page 395\]](#)

## 14.1.2 Start SAP Web IDE Personal Edition

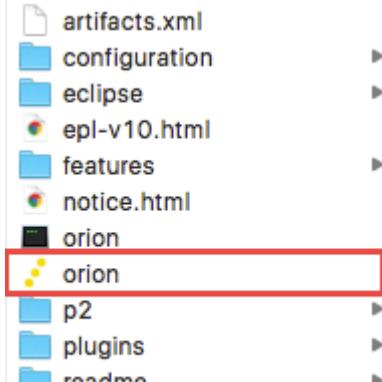
Follow the procedure below to open SAP Web IDE personal edition.

### 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 \[page 403\]](#).

| Microsoft Windows®   | Mac OS®   |
|--|---|
| <p>1. Go to C:\SAPWebIDE and open the eclipse folder.<br/>2. Double-click the <code>orion.exe</code> file.</p> <p>A command shell opens.</p> | <p>1. In the Finder, go to /Applications/SAPWebIDE/eclipse<br/>2. Hold the <code>Ctrl</code> key and click the <code>orion</code> file.</p>  <p>3. From the context menu, select <i>Open</i>.</p> <p><b>Note</b><br/>If prompted, you should confirm the use of Orion.</p> <p><b>Note</b><br/>If you have installed macOS™ Sierra, you need to perform some post-installation steps. For more information, see the blog post: <a href="#">Post-Installation tips for SAP Web IDE Personal Edition on MAC</a>.</p> |

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.

#### **Note**

When you start the Orion Application server for the first time (after initial installation or upgrade), you have to create a new account for it. (In the future this might be done automatically.)

1. Choose [Create a new account](#).
2. Enter a user name and password.

SAP Web IDE personal edition does not support more than one account.

3. On the Orion Server Application logon page, enter the user and password that you defined.

## 14.1.3 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:
  - o `orion.ini` file: This file holds proxy information and has to be recreated during each installation.
  - o `orion.conf` file: This file is used when the Orion configuration is changed. For example, when setting up SSL or changing the port.
  - o `destinations` folder: This folder contains the destination you have set up.
  - o `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 \[page 395\]](#).

**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 \[page 397\]](#).

4. Restore the following files and folders:
  - o `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.
  - o `orion.conf` file.
  - o `destinations` folder. Restore the folder to the following path: `config_master/service.destinations/destinations`.
  - o `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 \[page 403\]](#)
- [Connect Remote Systems in SAP Web IDE Personal Edition \[page 405\]](#)
- [Import the Git Server Certificate into the JVM \[page 402\]](#)

## 14.1.4 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.

## 14.2 Import the Git Server Certificate into the JVM

If the server certificate is based on a company internal root certificate, you have to import the root certificate into the JVM.

### Context

If you are connecting to a Git server in your company via HTTPS, the Java Virtual Machine (JVM) on which the Orion installation is running has to trust the server certificate. If the server certificate is not issued by a public agency, but is based on a company internal root certificate, you have to import the root certificate into the JVM. Otherwise the verification of the Git server's certificate fails and prevents you from performing Git operations.

To import your certificate into the JVM, fetch the certificate \*.cer file from your company's IT department and store it on your machine, as follows:

| Microsoft Windows®  | Mac OS®   |
|---|---|
| <ol style="list-style-type: none"><li>In the command shell, enter the following command:<br/><code>cd C:\Program Files\Java\jre&lt;VERSION&gt;\lib\security</code><br/>Example:<br/><code>C:\Program Files\Java\jre7\lib\security</code></li><li>Next, enter the following command:<br/><code>keytool -import -file &lt;PATH OF THE .cer FILE&gt; -keystore cacerts -alias &lt;ANY ALIAS&gt;</code></li></ol> | <ol style="list-style-type: none"><li>In the Terminal window, enter the following command:<br/><code>cd /Library/Java/JavaVirtualMachines/jdk&lt;VERSION&gt;.jdk/Contents/Home/jre/lib/security</code><br/>Example:<br/><code>/Library/Java/JavaVirtualMachines/jdk1.7.0_65.jdk/Contents/Home/jre/lib/security</code></li><li>Next, enter the following command:<br/><code>sudo keytool -import -file &lt;PATH OF THE .cer FILE&gt; -keystore cacerts -alias &lt;ANY ALIAS&gt;</code></li></ol> |

### i Note

If the keystore is protected by a password, ask your company's IT department for the password.

## 14.3 Configure the Orion Application Server

You may wish to configure the Orion Application Server for your system landscape.

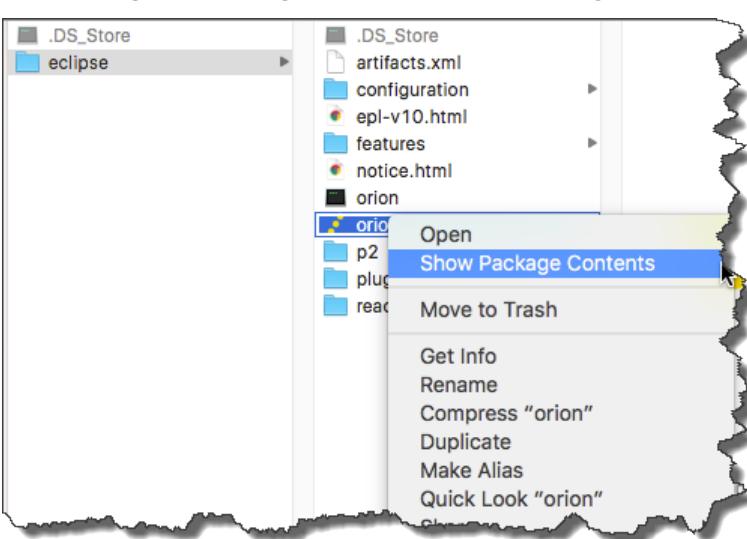
### Context

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

- Open the `orion.ini` file:

| Microsoft Windows®                                       | Mac OS®   |
|--|---|
| Open the <code>orion.ini</code> file with a text editor. | <p>1. In the Finder go to <code>orion</code>, right-click, and select <i>Show Package Contents</i>.</p>  <p>2. Go to the terminal and enter <code>cd /Applications/SAPWebIDE/eclipse/orion.app/Contents/MacOS</code></p> <p>3. Right-click <code>orion.ini</code> and open it with a text editor.</p> |

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.

## 14.4 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 \[page 31\]](#).

### 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®   | Mac OS®   |
|--|---|
| C:\SAPWebIDE\eclipse<br>\config_master<br>\service.destinations<br>\destinations | <ol style="list-style-type: none"><li>1. Go to the terminal and enter cd /Applications/SAPWebIDE/eclipse/orion.app/Contents/MacOS/config_master/service.destinations/destinations/</li><li>2. Create a new file by running the following command touch &lt;your system name&gt;</li></ol> |

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>
WebIDEEabled=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"> <li>○ odata_abap:<br/>For the OData functionality of Gateway (corresponds to URL path /sap/opu/odata)</li> <li>○ odata_gen:<br/>For generic OData functionality (service URL must be provided manually in the New Project wizard)</li> <li>○ ui5_execute_abap:<br/>For executing SAPUI5 applications from the SAPUI5 ABAP repository (corresponds to URL path /sap/bc/ui5_ui5)</li> <li>○ dev_abap:<br/>For extensibility scenarios and developing or deploying to the SAPUI5 ABAP repository (corresponds to URL path /sap/bc/adt)</li> <li>○ bsp_execute_abap:<br/>For working with fact sheets (corresponds to URL path /sap/bc/bsp)</li> <li>○ odata_xs:<br/>For SAP HANA XS OData services (corresponds to URL path /sap/hba)</li> </ul> <p><b>i Note</b></p> <p>When you enter multiple usages for a destination, separate them by commas without spaces (for example, <b>odata_abap.ui5_execute_abap</b>).</p> |

- 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 \[page 403\]](#)

## 14.5 Connect to an External Git Repository

Configure SAP Web IDE personal edition to access an external Git repository through a proxy.

### Context

Modify the configuration file `orion.ini` to allow SAP Web IDE personal edition requests to pass through a proxy. Then, configure the account settings for the SAP Web IDE personal edition Git client.

### Procedure

1. Make sure that the Eclipse Orion console is closed.
2. In the `C:\SAPWebIDE\eclipse` folder, locate and open the `orion.ini` file in the editor.
3. Insert the following entries after the `-vmargs` row:

```
-Dhttp.proxyHost=<proxy>
-Dhttp.proxyPort=<port>
-Dhttps.proxyHost=<proxy>
-Dhttps.proxyPort=<port>
-Dhttps.nonProxyHosts=<hosts to be excluded from proxy>
```

For example:

```
-Dhttp.proxyHost=myproxy.mycompany
-Dhttp.proxyPort=8080
-Dhttps.proxyHost=myproxy.mycompany
-Dhttps.proxyPort=8080
-Dhttps.nonProxyHosts=*.mycompany|localhost
```

4. Save the `orion.ini` file, and restart SAP Web IDE personal edition.



5. From the left sidebar, choose (Preferences) and select *Git Settings*.
6. Enter your e-mail and user name, and choose *Update*.

### Results

You have configured your SAP Web IDE personal edition Git client to use the proxy. You can test this by cloning an external Git repository.

## Related Information

[Clone Repositories \[page 291\]](#)

## 14.6 Run Applications with Multiple SAPUI5 Versions

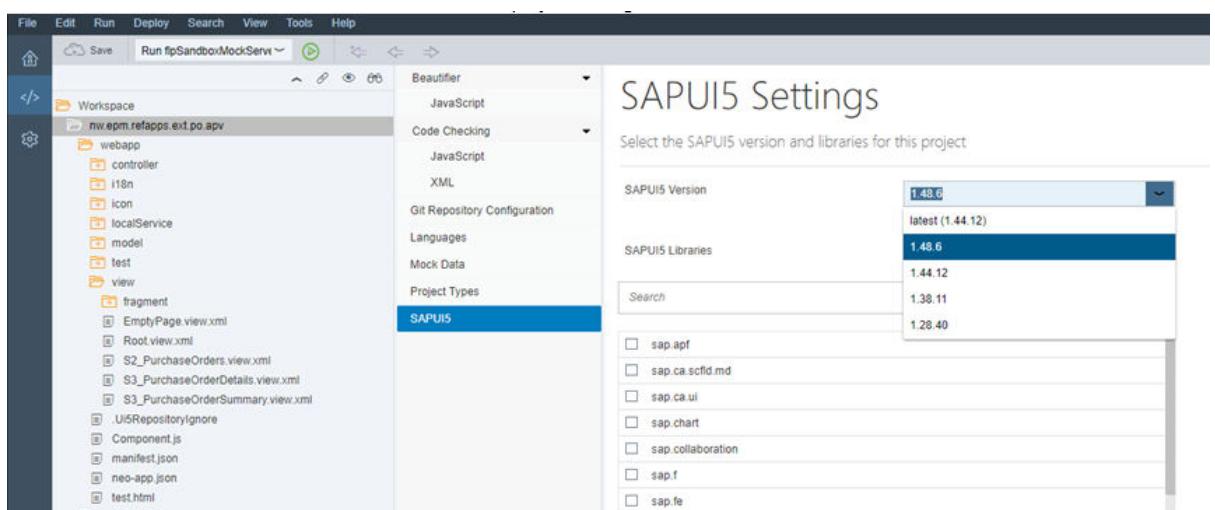
SAP Web IDE personal edition comes with three default SAPUI5 versions you can choose from. If you want to run your application using a different SAPUI5 version you must make the version available in SAP Web IDE personal edition.

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.



# 15 Security

When using SAP Web IDE, make sure that your data and processes support your business needs and prevent unauthorized access to critical information.

Errors due to the application users' actions, negligence, or any attempted malicious operation on your system should not result in loss of information or processing time. You must guarantee and comply with the legal regulations regarding protection of users' personal data.

In addition to using this documentation, refer to the other security documentation below:

- [Security Information for SAPUI5](#)
- [SAP HANA Security Guide](#)
- [Identity Service](#)
- [Cloud Connector](#)
- [Cloud Connector Operation Guide](#)
- [Principal Propagation \[page 419\]](#)
- [Destinations](#)
- [HTML5 Applications Development](#)

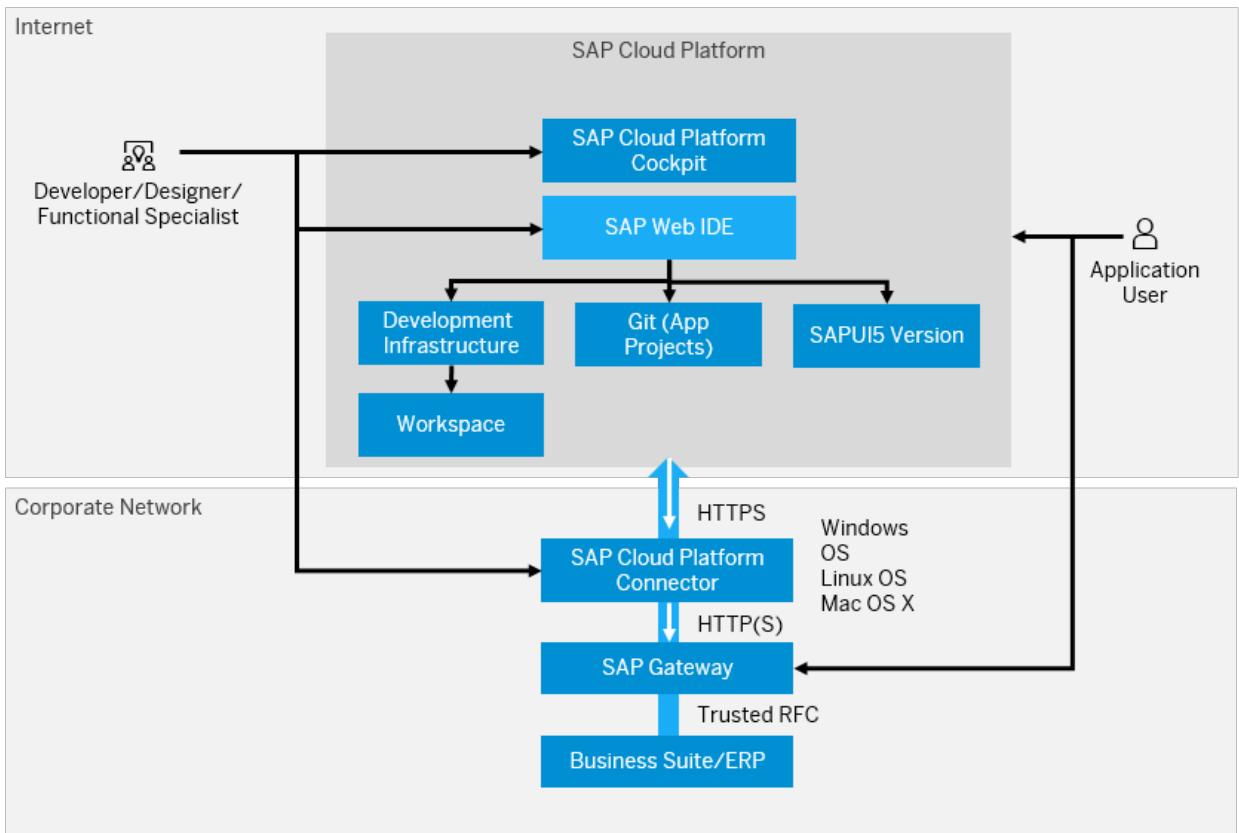
## 15.1 Architectural Overview

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

SAP Web IDE is integrated with the dispatcher, a mechanism that manages access to the application and various services, and enables you to connect to your SAP systems through REST services.

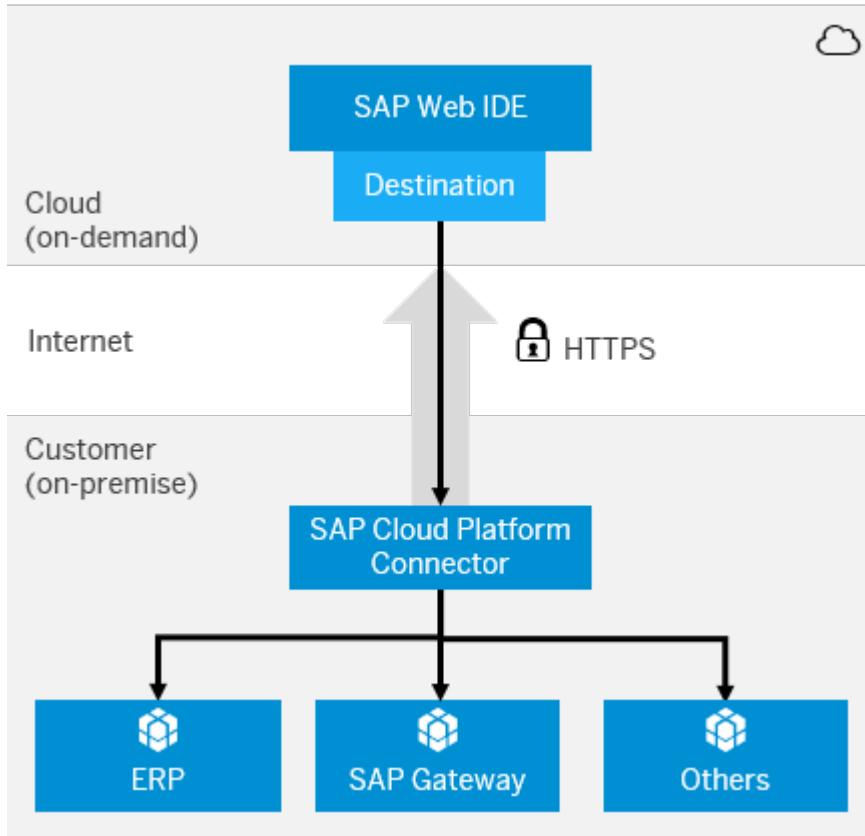
The Development Infrastructure (DI) is a server-side set of development infrastructure services, such as workspace management, build, run, and so on. The services are exposed as a rich, standard REST API, based on Eclipse Che Open-Source Software (OSS).

The image below illustrates the high level typical architecture for SAP Web IDE Full-Stack. For more information about the dispatcher, see [HTML5: Development](#).



SAP Web IDE implements the following features of SAP 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.



## 15.2 User Authentication and Authorization

User authentication and authorization processes in SAP Web IDE ensure that users can access only the resources to which they have the required permissions.

When developing in SAP Web IDE, you need permissions for the following software components:

- Git: Manages revisions and provides source code control. You must provide user credentials in the SAP ID service and an account to log on to the cloud.
- DI (Development Infrastructure): Used for development infrastructure, workspace management, build, run, and so on. DI leverages SSO as part of the user logon to SAP Web IDE.
- Dispatcher (on SAP 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.

## Related Information

[Authentication \[page 413\]](#)

[Assign Users Permission for SAP Web IDE \[page 413\]](#)

### 15.2.1 Authentication

Authentication and user propagation in SAP Web IDE occurs in the Dispatcher and consists of two authentication processes: IdP (Identity Provider) configuration and Git authentication.

#### IdP Configuration

IdP configuration is required to access SAP Web IDE.

You must configure an SAP IdP or a custom IdP as the identity provider for SSO.

**i Note**

When using SAP Cloud Identity (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

When you push changes to the Git server, the SAP ID service requests your SCN credentials.

## Related Information

[Identity and Access Management \(SAP Cloud Platform documentation\)](#)

[Principal Propagation \[page 419\]](#)

[Register, Create and Manage Your Profile \(SCN document\)](#)

[SAP Cloud Platform Connector \(SAP Cloud Platform documentation\)](#)

### 15.2.2 Assign Users Permission for SAP Web IDE

To enable working with SAP Web IDE Full-Stack, the `DiDeveloper` role needs to be assigned.

**i Note**

If you are a member of an account with the `AccountDeveloper` role, you are already assigned to the `DiDeveloper` role by default. You may proceed to work with SAP Web IDE.

## Individual User

The `DiDeveloper` role can be assigned to an individual user as follows:

1. In the SAP Cloud Platform cockpit, choose `Services` `SAP Web IDE Full-Stack` `Configure Service`.
2. On the `Configure Service` tab, under `New Role`, select `DiDeveloper`.
3. In the `Individual Users` area, choose `Assign`, then in the popup window, enter the user ID of the user to whom you want to assign the `DiDeveloper` role, and choose `Assign`.

## User Group

### Prerequisite

You have created one or more user groups in SAP Cloud Platform cockpit. For more information, see [Managing Roles](#).

The `DiDeveloper` role can be assigned to a user group as follows:

1. In the SAP Cloud Platform cockpit, choose `Services` `SAP Web IDE Full-Stack` `Configure Service`.
2. On the `Configure Service` tab, under `New Role`, select `DiDeveloper`.
3. In the `Groups` area underneath, choose `Assign`, then in the popup window, enter the group to which you want to assign the `DiDeveloper` role, and choose `Assign`.

## Related Information

[Account Member Roles](#)

[Managing Roles and Permissions](#)

[Managing Roles](#)

[ID Federation with the Corporate Identity Provider](#)

### 15.2.2.1 Maintain an IdP Mapping Rule for a User Group

How to maintain an IdP mapping rule for a large user base.

### Prerequisites

You have configured a corporate SAML identity provider (IdP) for your account. For more information, see [ID Federation with the Corporate Identity Provider](#).

## Context

This approach requires minimum ongoing effort to maintain an IdP mapping rule, especially where there is a large user base. When this IdP mapping rule is assigned, the required SAP Web IDE authorization is automatically derived from the IdP, making it unnecessary to assign it to each user separately.

## Procedure

1. In SAP Cloud Platform cockpit, choose [Security](#) [Authorizations](#), and select the *Groups* tab. Define a new group, for example, **DiDeveloperGroup**.
2. Choose [Security](#) [Trust](#), and in the *Application Identity Provider* tab, click the link with the relevant IdP name.
3. In the *Trusted Identity Provider* dialog box that opens, select the *Groups* tab and click the link, [Add Assertion-Based Group](#).
4. In the *Group* dropdown list, select the group created in the first step and define a mapping rule based on the SAML attribute issued by the custom IdP that will automatically assign users to the group if the mapping rule is matched.
5. In SAP Cloud Platform cockpit, again choose [Security](#) [Authorizations](#) and select the *Groups* tab. Select the group you created in the first step and assign it with the **DiDeveloper** role.

## Related Information

[Managing Roles](#)

[Account Member Roles](#)

### 15.2.3 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.

## 15.2.4 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.

## 15.2.4.1 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.<br><br><b>⚠ Caution</b><br>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](#)

## 15.3 Data Protection and Privacy

### Overview

SAP Web IDE is governed by the [SAP Cloud Platform Privacy Statement](#).

SAP Web IDE stores a user's projects and files as well as his or her preference settings. An administrator can export a user's workspace on request. For information about exporting a user workspace, see [Export Workspace \[page 418\]](#).

Additionally, SAP Web IDE uses SAP Web Analytics. SAP Web Analytics helps to analyze how users use the relevant site. The information generated (including a part of your IP address and a browser-ID) will be transmitted to and stored by SAP on servers in Germany. SAP will use this information only for the purpose of evaluating your use of the Web site and compiling reports on Web site activity for Web site operators - and finally, to improve SAP's Web sites.

By using SAP Web IDE, you consent to the processing of analytics data about you by SAP in the manner and for the purposes set out above. If you would like to opt-out of SAP Web Analytics tracking, in the application, go to ► [Help](#) ➔ [Privacy](#) and in the [SAP Web IDE Privacy Statement](#), deselect the *Allow tracking* checkbox and choose **OK**.

If an application built using SAP Web IDE includes person-related data, the application must comply with the data protection laws of its target countries. This includes the usage of proper authentication, authorization, and encryption, such as SSO and usage of HTTPS, as well as properly securing and logging access to person-related data.

## Git

SAP Web IDE is integrated with Git for source and version control. To work with Git, users need to provide their user name and e-mail in the SAP Web IDE [Preferences](#) perspective and in the [Project Settings](#) of each project. These settings are optional in SAP Web IDE, but if you don't provide them, you will not be able to work with Git.

### **i** Note

If you choose to provide your user name and e-mail, these settings are sent to and stored on the remote Git server of your choice. These settings cannot be deleted.

The settings that are configured in the [Preferences](#) perspective and the [Git Settings](#) area of the [Project Settings](#) are copied to each new project and stored in the Git repository configuration. A user can change these settings for each project because the credentials used for Git might be different for each project.

## Related Information

[Using Source Control \(Git\) \[page 283\]](#)

[Setting User Preferences \[page 34\]](#)

### 15.3.1 Export Workspace

An administrator can export a `.zip` file containing projects from a user's SAP Web IDE workspace.

#### Prerequisites

In order to export a workspace, you need to have the following:

- The administrator role for the user account.
- 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 to URL format.

- The SAP Web IDE service URL.

**i** Note

To get the SAP Web IDE service URL, follow this path:

► *SAP Cloud Platform Cockpit* ► *account* ► *Services* ► *SAP Web IDE Full-Stack tile* ► *Go to Service link* ▶

In the browser address bar, enter:

`https://{{SAP Web IDE service URL}}/di/workspace/export-all/{{userid}}`

**Result**

A .zip file containing the contents of the user workspaces is downloaded to your computer.

## 15.4 Principal Propagation

In order to support the SSO solution of SAP Web IDE Full-Stack, you need to configure your account to allow principal propagation.

**i** Note

The principal propagation setting is configured per account, which means that once it is enabled, applications propagate principal information to each other. If you do not want to enable application-to-application single sign-on, set this option to *Disabled*.

To enable the principal propagation setting, in the SAP Cloud Platform cockpit, choose ► *Security* ► *Trust Management* ▶ and set *Principal Propagation* to *Enabled*.

## Trust Management

Local Service Provider Trusted Identity Provider

 Manage Local Provider Settings  
for sapwebide

Configuration Type: Default

Principal Propagation: Enabled

Force Authentication: Default

Edit



For more information about configuring the principal propagation settings, see:

- [Application-to-Application SSO Authentication](#)
- [ID Federation with the Corporate Identity Provider](#)

## 15.5 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.

# 16 Troubleshooting

Here are some common troubleshooting issues in SAP Web IDE.

## [Archive Import Troubleshooting \[page 421\]](#)

If you are having trouble importing an archive (.zip file) into SAP Web IDE Full-Stack, you can try the following possible solutions.

## [Deployment to SAPUI5 ABAP Repository Troubleshooting \[page 422\]](#)

Steps you can take if you have trouble deploying your app to the SAPUI5 ABAP Repository.

## [Deployment to SAP Cloud Platform Troubleshooting \[page 424\]](#)

Steps you can take if you have trouble deploying your application to SAP Cloud Platform.

## [Git Troubleshooting \[page 426\]](#)

Steps you can take if you have trouble using Git.

## 16.1 Archive Import Troubleshooting

If you are having trouble importing an archive (.zip file) into SAP Web IDE Full-Stack, you can try the following possible solutions.

- **UTF-8 Encoding**

Make sure that the archive files and folder names in your project contain only characters that are encoded with UTF-8 encoding. In other words, these names must not contain any special characters such as a question mark (?) or an ampersand (&). In addition, make sure to use only Latin characters. Non-Latin characters such as Chinese, Japanese, or Hebrew cannot be used.

- **Check File Size**

The .zip file you want to import must be less than 20 MB.

- **Import .zip Files Only**

Make sure that the archive you want to import is a .zip file. Archives with other extensions, such as .rar, cannot be imported.

## Multitarget Application (MTA) Project Troubleshooting

- Check the correctness of the mta.yaml file.

For more information, see [Inside an MTA Descriptor \[page 355\]](#).

- Make sure that the path element of each module in the mta.yaml file points to a folder that actually exists in the archive.

**Parent topic:** [Troubleshooting \[page 421\]](#)

## Related Information

[Deployment to SAPUI5 ABAP Repository Troubleshooting \[page 422\]](#)

[Deployment to SAP Cloud Platform Troubleshooting \[page 424\]](#)

[Git Troubleshooting \[page 426\]](#)

## 16.2 Deployment to SAPUI5 ABAP Repository Troubleshooting

Steps you can take if you have trouble deploying your app to the SAPUI5 ABAP Repository.

The following list includes error messages you may receive when deploying, possible causes of the error, and possible solutions.

### Note

The technical messages are displayed in the language selected for your back-end system. Here we provide the English version of these errors.

### HTTP Status 504 - An internal application error occurred

This can happen if the application contains a rather large file, and the HTML5 dispatcher (which is on the route from SAP Web IDE to the SAP system) gets a timeout when trying to dispatch it.

Sometimes, the HTML5 dispatcher has a shorter timeout (30s) than the ABAP system itself (5m). When this happens, the ABAP system is actually able to handle the upload of the large file, but the HTML5 dispatcher raises a timeout in the process.

The solution is to increase the timeout of the HTML5 Dispatcher. To do this, follow the instructions under *Destination Properties* in [Accessing REST Services](#) and increase the timeout to the maximum of 300s.

If you're still experiencing this error after raising the timeout, you may need to perform an ICM hard shutdown. ICM (Internet Communication Manager) handles all inbound HTTP/HTTPS/SMTP connections .

Use `SMICM` transaction to view the status of these connections. The hard shutdown may be necessary if you are pushing an app from SAP Web IDE, as the connection may be held open for some reason. The action is triggered from the menu by selecting  `Administration`  `ICM`.

If the problem persists, locate the large file and upload it manually via the `SE80` transaction.

## Namespace Errors

There are a variety of namespace errors:

- **Test objects cannot be created in foreign namespaces**
- **Remote creation in customer namespace not possible in SAP systems**
- **SAP object <X> cannot be assigned to package <Y>**

These errors are caused by the namespace you provided.  
The target system runs in either SAP or Customer mode.  
If it is running in SAP mode, you can only use the SAP namespace in the given application name.  
If it is running in Customer mode, you can only 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. 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.

**Parent topic:** [Troubleshooting \[page 421\]](#)

## Related Information

[Archive Import Troubleshooting \[page 421\]](#)

[Deployment to SAP Cloud Platform Troubleshooting \[page 424\]](#)

[Git Troubleshooting \[page 426\]](#)

## 16.3 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.

**Parent topic:** [Troubleshooting \[page 421\]](#)

## Related Information

[Archive Import Troubleshooting \[page 421\]](#)

[Deployment to SAPUI5 ABAP Repository Troubleshooting \[page 422\]](#)

[Git Troubleshooting \[page 426\]](#)

## 16.4 Git Troubleshooting

Steps you can take if you have trouble using Git.

The following lists error messages you may receive when using Git, possible causes of the error, and possible solutions.

### Invalid committer

The e-mail listed in the Git repository is not the same as the e-mail assigned to you in Gerrit.

1. Right-click on your project, select  [Project Settings](#)  [Git Repository Configuration](#) and change the e-mail address in the `user.email` field.
2. Open the [Preferences](#) perspective, choose [Git Settings](#), and verify that the [Git Email Address](#) value is correct.
3. Commit your changes again, this time by selecting the [Amend](#) checkbox and then selecting [Commit](#). You can then try to push your changes again.

### NON FAST FORWARD

Someone else pushed new changes to the remote repository.

Sync your repository (either [Fetch](#) and then [Rebase](#), or [Pull](#)).

### Prohibited by Gerrit

If you did not select the [Add configuration for Gerrit](#) checkbox when cloning your project, and the Git repository is connected to a Gerrit server, your remote repository will not work.

Right-click on your project, select  [Project Settings](#)  [Git Repository Configuration](#), and add an entry with the key `gerrit.createchangeid` and set the value to `true`.

### Cannot upload review

You do not have permission to push changes to Gerrit. Request from your administrator permissions on Gerrit.

### Checkout failed

One of the files listed in the `.gitignore` file is preventing you from checking out a different branch.

1. Open `.gitignore` file (original branch).
2. Remove lines containing files blocking checkout.
3. Click *Discard* for the file if it appears in the staging table in the Git pane.
4. Stage the `.gitignore` file and commit it.
5. Check out your branch.
6. Create `.gitignore` file and save it. (If the file already exists, then make a small change and save it.)
7. In staging table, right-click the files blocking checkout and select *Untrack and Ignore*.
8. Stage, commit, and push your changes.
9. Merge your change in Gerrit.
10. Check out the original branch.
11. Select *Reset*.

## Not authorized

You are not authorized in the Git system. This error may occur simply because the password was incorrect, for one of the following reasons:

- You entered the wrong password.
- The wrong password was cached in the browser. Clear the browser cache of passwords.

## 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 \[page 289\]](#).
- **502 Bad Gateway:** You are using a Git system within your corporate network. The channel to the Git system 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 \[page 289\]](#).

## 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.

**Parent topic:** [Troubleshooting \[page 421\]](#)

## Related Information

[Archive Import Troubleshooting \[page 421\]](#)

[Deployment to SAPUI5 ABAP Repository Troubleshooting \[page 422\]](#)

[Deployment to SAP Cloud Platform Troubleshooting \[page 424\]](#)

# 17 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.

### Note

When you define the cloud connector, there are 2 types of hosts: Internal and Virtual. Make sure the internal and external hosts have the same name.

## SAP Web IDE Freezes while Debugging

Make sure you allow popups in your browser the first time run your app.

## SAPUI5 Elements Might Not Display Correctly in Preview Mode

When you open an application in preview mode, SAPUI5 elements in the iPad or iPhone view might not display as expected.

## Parallel Instances of SAP Web IDE Can Cause Issues

Opening several instances of SAP Web IDE in parallel can cause issues and lead to unhandled exceptions. We recommend that you work on a single SAP Web IDE instance.

## SAP Web IDE Fails to Load

- SAP Web IDE fails to load after an update because of an error originating from one of the open files in the editor.

### Solution

Add the URL parameter `settings=ignore` to the SAP Web IDE URL and refresh your browser. This will force SAP Web IDE to ignore all settings configured, including which files are currently open in the editor. This will force the closing of all open files and allow SAP Web IDE to load.

If this does not work, add the URL parameter `settings=delete` to the SAP Web IDE URL and refresh your browser. SAP Web IDE should now load successfully. Note that this will delete every special configuration made in the settings.

Once SAP Web IDE loads, remove the URL parameter from the SAP Web IDE URL and refresh your browser.

- SAP Web IDE fails to load due to an error originating from a plugin.

### Solution

1. Add the URL parameter `settings=ignore` to the SAP Web IDE URL and refresh your browser. SAP Web IDE should now load successfully.
2. Go to [Preferences](#) [Plugins](#).
3. Clear the checkbox of the problematic plugin and choose [Save](#).
4. Remove the URL parameter `settings=ignore` from the SAP Web IDE URL and refresh your browser.

## Web Authorization Changes Cause 403 Error Message

When an identity provider is specified for your system and you did not change the default authorization settings accordingly, you will receive the following error message:

*HTTP Status 403 - You are not authorized to access this resource*

### Solution

Set (or ask your administrator to set) the proper authorization settings for your system. See [Assign Users Permission for SAP Web IDE \[page 413\]](#).

If no other role is assigned to the `WebIDEPermission` permission, only account members with developer or administrator permissions will have access to the environment. See [Managing Roles and Permissions](#).

## Error Message When Logging On

You may receive an error message *Unhandled Error: Unexpected token* when logging on to SAP Web IDE.

### Solution

Close the message and refresh the browser.

## Issues with Logging On Again to SAP Web IDE

You may have issues logging on again to SAP Web IDE, for example, after logging out.

### Solution

Empty your browser cache and log on again.

## Browser Issues

### Mozilla Firefox Browser Support

SAPUI5 does not support smartphone simulation in the Mozilla Firefox browser using the `fakeos` parameter in the URL (which is what the preview service requires).

When you preview your application in the Mozilla Firefox browser, it will always run as a tablet or desktop.

### Safari Browser Support

- **Basic Authentication**

If you defined a destination to a remote system that requires a *Basic Authentication* popup, this popup may not appear for synchronous requests when running or extending an application with SAP Web IDE in the Safari browser.

For more information on defining a destination to a remote system, see [Connect to ABAP Systems \[page 28\]](#)

### Solution

If you cannot change the application coding, you can use a different authentication type in the destination maintenance, such as *Principle Propagation*.

For your own application, ensure that you load the OData metadata asynchronously:

```
For example new sap.ui.model.odata.ODataModel(sServiceUrl, {json:  
    true, loadMetadataAsync: true});
```

- **Exported Files**

Exported files of projects or folders do not have the .zip extension. (To export a file, choose    *File*  *Export Project or Folder* from the menu bar.)

### Solution

Manually add **.zip** as an extension to the exported file.

- **Layout Editor**

SAP Web IDE layout editor is not supported on Safari.

- **Running with Frame**

The Running with Frame feature does not work properly when using Private Browsing.

### Internet Explorer Issues

When previewing an application in regular preview or via the Extensibility pane, you might receive an error message saying the content was blocked because it was not signed by a valid security certificate.

#### Solution

Add SAP Web IDE as a trusted site as follows:

1. Open your Microsoft Internet Explorer and select *Internet Options*.
2. Choose the *Security* tab.
3. Choose *Trusted sites*.
4. Make sure the *Enable Protected Mode* checkbox is deselected.
5. Choose *Sites*.
6. To add the SAP Web IDE web site to the list of trusted sites, enter the 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.

#### Note

Use the XML code editor to perform operations on the view layout that you cannot achieve with the layout editor.

- **Only ResponsiveGridLayout for SimpleForm Control Supported**

The layout editor only supports the *ResponsiveGridLayout* layout for the *SimpleForm* control. Other layouts might cause issues with the graphical display of the XML view in the canvas.

- **Problems on Safari**

SAP Web IDE layout editor is not supported on Safari.

## Blank Page When Comparing Code in Git

When opening the compare editor in Git, by double-clicking the file in the staging table row, you may find that a blank page opens.

**Solution:**

Close the tab of the file that you want to compare and double-click it again in the staging table row. This time the compare editor opens properly.

## Deletion of the .Project.json File Content

When manually editing a project's `.project.json` file, or if the file has syntax errors, its content might be automatically deleted.

**Solution:**

Create a copy of the `.project.json` file before you edit and revert to it if needed.

## Duplication of Projects

A project might appear multiple times in the SAP Web IDE workspace.

The project is not physically duplicated, the data has one occurrence in the data storage.

## Browser Issue When Running Reference Applications with Mock Data

Running reference applications with mock data will not work on the Microsoft Internet Explorer <sup>®</sup> browser.

**Solution:**

You can try one of the following:

- Reduce length of the mock data files to less than 1,000 lines.
- Use another browser.

## Refactor and Find References of Class Variables within Function

When trying to find references for a class variable (that is, properties of the `this` object) from within a function, only references from within the function are shown in the [References](#) tab of the [Search](#) pane. This also affects refactoring: when refactoring a class variable from within a function, only the references of the class variable within the function are changed.

## Replace Data source enhancements

When using the [Replace with Node](#) option, if the user tries to adjust the mapping via the [Remove mapping](#) context menu option, the mapping might get corrupted.

### Solution:

Avoid using the [Remove mapping](#) context menu option together with the [Replace with Node](#) option. We recommend you adjust the mapping in the mapping pane instead.

## Synonym support on SAP Cloud Platform

There is not sufficient authorization to view the data preview of Synonyms which consumes the Calculation View from other containers.

## QR Code in Preview Frame

When running an application in a frame, long application URLs can cause the application's QR code not to work. Remove long URL parameters and try running the application again.

# 18 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

## Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

## Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

## Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: <https://help.sap.com/viewer/disclaimer>).





[go.sap.com/registration/  
contact.html](https://go.sap.com/registration/contact.html)

© 2018 SAP SE or an SAP affiliate company. All rights reserved.  
No part of this publication may be reproduced or transmitted in any  
form or for any purpose without the express permission of SAP SE  
or an SAP affiliate company. The information contained herein may  
be changed without prior notice.  
Some software products marketed by SAP SE and its distributors  
contain proprietary software components of other software vendors.  
National product specifications may vary.  
These materials are provided by SAP SE or an SAP affiliate company  
for informational purposes only, without representation or warranty  
of any kind, and SAP or its affiliated companies shall not be liable for  
errors or omissions with respect to the materials. The only  
warranties for SAP or SAP affiliate company products and services  
are those that are set forth in the express warranty statements  
accompanying such products and services, if any. Nothing herein  
should be construed as constituting an additional warranty.  
SAP and other SAP products and services mentioned herein as well  
as their respective logos are trademarks or registered trademarks of  
SAP SE (or an SAP affiliate company) in Germany and other  
countries. All other product and service names mentioned are the  
trademarks of their respective companies.  
Please see <https://www.sap.com/corporate/en/legal/copyright.html>  
for additional trademark information and notices.