

2017-06-21

SAP Web IDE Hybrid App Toolkit Add-on



Content

1	SAP® Web IDE Hybrid Application Toolkit	5
2	What's New in Hybrid App Toolkit 1.28.	8
3	Developing Apps through SAP Cloud Platform mobile service for SAP Fiori (Fiori Mobile)	9
3.1	Prerequisites.....	9
3.2	Creating a New Project..... Choosing a Project Template..... Creating a New Extension Project Using the Hybrid Mobile Enablement Feature..... Visual Mobile Quality Development.....	11 11 14 15
3.3	Testing..... Testing Functionality in a Browser with Cordova Facade..... Running a Simple Preview with Hybrid App Project Templates..... Testing Your Hybrid Mobile Apps Using the Device Test Cloud Service..... Using QR Codes to Load and Test Apps on Devices..... Using preview URL to load and test the apps on device.....	15 16 26 26 28 29
3.4	Developer Companion for Fiori Mobile Apps..... Building Developer Companion for Fiori Mobile Apps.....	29 30
3.5	Deploying Apps to SAP Cloud Platform and Registering to SAP Fiori Launchpad.....	32
3.6	Configuring Apps for Fiori Mobile Build..... Creating Signing Profiles..... Understanding the Verbose Logs.....	32 34 36
3.7	Push Notifications for Fiori Mobile Apps.....	37
3.8	Customizing the Theme for Fiori Mobile Apps.....	40
3.9	Things to Remember.....	41
4	Developing Apps through Hybrid App Toolkit Local Add-on.....	44
4.1	Upgrading the Hybrid App Toolkit Add-on for Cloud SAP Web IDE.....	44
4.2	Installing and Setting Up..... Preparing the Cordova Development Environment..... Downloading and Extracting the Windows or Macintosh Installer..... Enabling the SAP Web IDE Plugin for Hybrid App Toolkit..... Using Custom Cordova Plugins.....	45 45 53 57 58
4.3	Getting Started..... Starting Hybrid App Toolkit Connector..... Strengthening Connection Security..... Creating a New Project.....	60 60 62 62

Configuring Device Properties	74
4.4 Testing	76
Testing Functionality with Native Device APIs Using the Hybrid App Toolkit Companion	76
4.5 Building and Deploying Apps	77
Building and Packaging Apps for Installation	77
Deploying Apps to SAP Cloud Platform	79
Deploying Apps to SAP Cloud Platform mobile service for development and operations	80
Deploying to SAP Mobile Platform (SMP)	83
5 Consuming Internet of Things Services	85
5.1 Enabling Internet of Things Services	85
5.2 Assigning Roles for Internet of Things Services	85
5.3 Configuring Destinations for the Message Management Service and Remote Device Management Service	86
5.4 Configuring OAuth Authentication	88
5.5 Managing the Internet of Things Services	91
Creating Device Types	91
Deleting Device Types	92
Creating Message Types	92
Deleting Message Types	93
Creating Devices	93
Deleting Devices	94
5.6 Inject the Internet of Things library into your project	94
5.7 Consuming the Internet of Things APIs	95
6 Troubleshooting	97
6.1 Working Around Known Issues	97
Error Occurs When Cordova Camera Plugin is Added to the Application	100
Issue with Choosing Tabs on the <i>Select Cordova Plugins</i> Screen	100
Issue with Creating an Project Using SAPUI5 1.28	100
Issue with Previewing an iOS App Using Xcode version 7	100
Issues with Running Apps Using Xcode version 8 on iOS Devices	100
Issue with accessing a Fiori Mobile App in the Fiori Launchpad	101
Issue with Displaying a List Report Application Built Using Fiori Mobile Build	102
App Update Fails on iOS Device	102
Issue with Snippet Pane in Safari Browser	102
Issue with Smart Template on iOS Platform	102
Mobile-Enabled Fiori Master Detail Project Not Responding	102
Hybrid App Toolkit Companion Preview Fails for Web Extension Project on iOS Device or Simulator	103
Hybrid App Toolkit Companion Throws Exception When Home Button is Pressed	103
Voice Recorder Plugin Issue on Windows Platform	103

Access File Plugin Issue.	103
Issue with Previewing an Application on Chrome.	103
Deployment Issue with Internet Explorer Versions 10 and 11.	104
Fiori App Issue.	104
Issue with Fiori apps created using Shop or Manage Products as sample applications.	104
VoiceRecording Plugin Issue.	104
Compatibility Issue Occurs in Some Projects.	105
Search Field Issue.	105
SMP Deployment Issue.	106
Hybrid App Toolkit Companion's Back Button on iOS Simulator.	106
Kapsel Calender Plugin.	106
Project Directory Locks Unexpectedly on Android.	107
Android Emulator Fails to Start.	107
Issue with Barcode Scanner UI in Android Devices.	108
Issue with Passcode Page Format.	108
Issue with Downloading .ipa File in Safari Browser.	108
Kapsel Push Plugin.	108
Issue with Kapsel Logon Manager Plugin in iOS Simulator.	109
6.2 Resolving Usage or Configuration Errors.	109
Hybrid App Toolkit Connector Fails Without Error Message.	109
Cordova Development Tooling Connections Fail.	110
6.3 Reporting an Unknown Issue.	110
7 Previous Releases.	112
7.1 Hybrid App Toolkit 1.27.	112
7.2 Hybrid App Toolkit 1.26.	113
7.3 Hybrid App Toolkit 1.25.	114

1 SAP® Web IDE Hybrid Application Toolkit

SAP Web IDE Hybrid App Toolkit Add-on is an optional plugin for SAP Web IDE which provides the Fiori Mobile developer experience. It comprises three components which enable developers to create and deploy Apache Cordova-based hybrid apps.

A subscription for SAP Fiori Cloud, premium edition is required for each user who wants the Fiori Mobile developer experience to build hybrid applications.

Table 1: Hybrid App Toolkit Components

Component	Description	Location
SAP Web IDE plugin	<p>Adds the Fiori Mobile developer experience to SAP Web IDE:</p> <ul style="list-style-type: none">Automatic code completion and documentation for Cordova and Kapsel plugin APIs.Hybrid project templates and code snippets.Previews on Hybrid App Toolkit Companion, and browsers.Support for building apps through Fiori Mobile buildSupport for downloading and installing these apps from SAP Mobile Place.Device configuration (applicable only for Hybrid App Toolkit local build).Deploy to local, SAP Cloud Platform mobile service for development and operations and on-premise SAP Mobile Platform environments (applicable only for Hybrid App Toolkit local build).	A plugin for SAP Web IDE.

Component	Description	Location
SAP Hybrid App Toolkit Companion	<p>A native mobile application that runs on a mobile device or device emulator. It enables a live preview of a web application created with SAP Web IDE.</p> <p>Following are the benefits of using the Hybrid App Toolkit Companion:</p> <ul style="list-style-type: none"> • It contains all the Cordova and Kapsel plugins supported by Hybrid App Toolkit. So any plugins used in the application will work in the Hybrid App Toolkit Companion. • It allows you to instantly preview the web application in the Hybrid App Toolkit Companion without packaging the application. • It contains the Kapsel Barcode Scanner plugin that allows you to scan the application's barcode displayed in SAP Web IDE and instantly load the application. <p>See Using QR Codes to Load and Test Apps on Devices [page 28].</p>	<p>Included with the download add-on package for Hybrid App Toolkit. Download this package from the SAP Store.</p> <p>The Hybrid App Toolkit Companion is also available for download from the Google play store and Apple app store. For more information, see the document provided with the app.</p>
Hybrid App Toolkit Connector (Applicable only for Hybrid App Toolkit local build)	A local server process that enables SAP Web IDE to connect to the local system's Cordova development environment, and allow developers to create and manage a local Cordova project.	<p>Included with the download add-on package for Hybrid App Toolkit. Download this package from the SAP Store.</p>

Read this document to learn how to set up the toolkit, use the add-on to develop Fiori Mobile apps in SAP Web IDE, trigger a Fiori Mobile build and download and install these apps from SAP Mobile Place. You can also deploy these apps to the SAP Mobile Platform or SAP Cloud Platform mobile service for development and operations.

Related Information

[Installing and Setting Up \[page 45\]](#)

[Getting Started \[page 60\]](#)

[Testing \[page 15\]](#)

[Building and Deploying Apps \[page 77\]](#)

2 What's New in Hybrid App Toolkit 1.28

Learn about the changes to Hybrid App Toolkit.

New Features

- Session timeout support for apps:
You will receive a warning message on your mobile device if the app is idle for 20 minutes.
- Custom UI5 Library for apps based on the template **SAP Fiori Worklist Application**:
If you create apps based on this template, you should use the SAPUI5 library version 1.46.7.

Experimental Features

Experimental features are those that are currently still in development. They are available only in trial SAP Web IDE landscapes.

There are no new experimental features in this release.

Changed Requirements

Table 2:

Development Environment Dependency	Previous Requirement	Current Requirement
Cordova environment	6.3.1	6.5
SMP Hybrid SDK	SP14	SP15

Resolved Issues

For a list of open issues, see [Troubleshooting \[page 97\]](#).

Note

To understand the changes introduced in the previous releases, see [Previous Releases \[page 112\]](#).

3 Developing Apps through SAP Cloud Platform mobile service for SAP Fiori (Fiori Mobile)

SAP Web IDE and the Hybrid App Toolkit plugin now support creating and building hybrid apps through the SAP Cloud Platform mobile service for SAP Fiori (Fiori Mobile). You can now develop your apps in SAP Web IDE, trigger a Fiori Mobile build and download and install these apps from SAP Mobile Place.

Following are the steps:

1. Getting a subscription for SAP Fiori Cloud, premium edition.
2. Completing the prerequisites
See [Prerequisites \[page 9\]](#).
3. Creating a hybrid mobile-enabled app
See [Creating a New Project \[page 11\]](#).
4. Modifying the resource files
5. Testing the app
See [Testing \[page 15\]](#).
6. Deploying apps to SAP Cloud Platform and Registering to SAP Fiori Launchpad
See [Deploying Apps to SAP Cloud Platform and Registering to SAP Fiori Launchpad \[page 32\]](#).
7. Configuring apps for Fiori Mobile build
See [Configuring Apps for Fiori Mobile Build \[page 32\]](#).

3.1 Prerequisites

Ensure that you complete the prerequisites mentioned here before you proceed to create and build hybrid apps through Fiori Mobile build.

Prerequisites

- In SAP Web IDE, go to **Tools** **Preferences**, choose **Plugins**, and enable the Hybrid App Toolkit plugin
- In the SAP Cloud Platform cockpit, go to **Services** and enable **Fiori Mobile** (for factory accounts contact your SAP channel to provision the Fiori Mobile service), **Development and Operations**, and **Portal Service**.

In these services, the app developer must have the following roles:

Table 3: SAP Cloud Platform Roles

SAP Cloud Platform Services	Roles
Fiori Mobile	Account Admin, App Catalog Admin, App Catalog Publisher, Mobile Place User
Mobile Packager (available when you subscribe to Fiori Mobile)	Administrator
Development and Operations	Administrator
Development and Operations	Notification User (if your apps require push notification)
Portal Service	TENANT_ADMIN

i Note

Trial account users automatically have these roles except the Notification User role. Production account users must be manually assigned these roles.

- In the SAP Cloud Platform cockpit, go to *Services* and choose *Portal Service*. Create a new SAP Fiori Launchpad site using the available templates, publish the site, and set it as the default site.
 1. To create a new launchpad site, see https://help.hana.ondemand.com/cloud_portal/frameset.html?c8ccc454aa8444d6b0aab6e269b0fb83.html.
 2. After you create a new launchpad site, hover over the site tile, click the drop-down arrow and select *Publish*. Click *Publish* in the dialog box that appears.
 3. Hover over the site tile, click the drop-down arrow, and select *Set as default*. Click *Ok* in the dialog box that appears.
- SAP Web IDE requires that a webidetesting destination exists in the SAP Web IDE account to make sure all OData services effectively function in the Developer Companion. You must manually add this destination. This destination needs to be added only for the trial accounts where Fiori Mobile was enabled before March 29, 2017.
 1. In the SAP Cloud Platform cockpit, go to **Services** **Fiori Mobile** **Configure Mobile Packager** **Destinations** .
 2. Add a new destination with the following settings:

Table 4: Web IDE Testing Destination

Parameter	Value
Name	webidetesting
Type	HTTP
Description	Web IDE Testing

Parameter	Value
URL	For trial accounts: https://webidetesting-{account name}.dispatcher.hana-trial.ondemand.com
Proxy Type	Internet
Authentication	AppToAppSSO

3.2 Creating a New Project

There are different project creation methods available from SAP Web IDE for creating hybrid apps.

Following are the different methods:

- Create a new project by choosing one of the mobile-enabled templates provided by SAP Web IDE
- Create a new project using the Hybrid Mobile Enablement feature.

3.2.1 Choosing a Project Template

There are different project templates available from SAP Web IDE.

Context

You can create hybrid apps using one of the following templates:

- SAP Fiori Master-Detail Application
- SAP Fiori Worklist Application

To create apps based on this template, you should use the SAPUI5 library version 1.46.7. You can download and use it as a custom library in your device configuration.

To download the SAPUI5 library 1.46.7 version, go to <https://mdocs-sapui5.hanatrial.ondemand.com/mcm/public/v1/open?sh=8175S80tjCafVbwpvBw0VQtqLCvm9t7Ik5q6E0cYdQ>.

- CRUD Master-Detail Application
- SAPUI5 Application
- List Report Application

These templates are verified templates that are mobile compatible.

3.2.1.1 Creating a Hybrid Mobile-Enabled Project

To create a hybrid mobile-enabled project, choose the SAP Fiori Master-Detail Application, SAP Fiori Worklist Application or CRUD Master-Detail Application template.

Procedure

1. In SAP Web IDE, create the hybrid app project by selecting *File* > *New* > *Project from Template*.
2. In the *Category* drop-down, go to *SAP Fiori Application* and choose *SAP Fiori Master-Detail Application*, *SAP Fiori Worklist Application*, or *CRUD Master-Detail Application*.
3. In the *SAPUI5 Version* drop-down, choose *SAPUI5 1.38* or above.
4. In *Basic Information*, set the project name; the domain is optional.

Tip

Use lowercase for the project name. This convention keeps the name consistent across several deployment options.

5. In *Data Connection*, choose a service from the list of available sources, then choose a corresponding service.

Each source has a different set of configuration requirements:

Option	Description
<i>Service Catalog</i>	<ol style="list-style-type: none">1. Select the system that holds the catalog.2. If the system requires authentication, use the appropriate login credentials.3. On successful authentication, select the desired catalog name.
<i>Workspace</i>	Expand the top-level folder to browse for the service on your SAP Web IDE workspace.
<i>Filesystem</i>	Click <i>Browse</i> to select the location of the service on your local machine.
<i>Service URL</i>	Select the service that is available, paste the required URL for it, then select the adjacent arrow button.

6. In *Template Customization*, enter the *Initial View Details* as required, then click *Finish*.

3.2.1.2 Creating a SAPUI5 Application Project

Use the *SAPUI5 Application* template to create a basic project for developing hybrid apps

Procedure

1. In SAP Web IDE, create the hybrid app project by selecting *File* > *New* > *Project from Template*.
2. In the *Category* drop-down, go to *SAP Fiori Application* and choose *SAPUI5 Application*.

3. In the *SAPUI5 Version* drop-down, choose *SAPUI5 1.38* or above.
4. In *Basic Information*, set the project name; the domain is optional.

➔ Tip

Use lowercase for the project name. This convention keeps the name consistent across several deployment options.

5. Confirm the details and click *Finish*.

3.2.1.3 Creating a List Report Application

Use the *List Report Application* template to create a SAP Fiori application that is based on predefined templates and controllers using OData services and annotation files.

Procedure

1. In SAP Web IDE, create the hybrid app project by selecting ➔ *File* ➔ *New* ➔ *Project from Template* ➔
2. In the *Category* drop-down, go to *SAP Fiori Elements* and choose *List Report Application*.
3. In the *SAPUI5 Version* drop-down, choose *SAPUI5 1.38* or above.
4. In *Basic Information*, set the project name; the title, namespace and domain are optional.

➔ Tip

Use lowercase for the project name. This convention keeps the name consistent across several deployment options.

5. In *Data Connection*, choose a service from the list of available sources, then choose a corresponding service.

Each source has a different set of configuration requirements:

Option	Description
<i>Service Catalog</i>	<ol style="list-style-type: none">1. Select the system that holds the catalog2. If the system requires authentication, use the appropriate login credentials.3. On successful authentication, select the desired catalog name.
<i>Workspace</i>	Expand the top-level folder to browse for the service on your SAP Web IDE workspace.
<i>Filesystem</i>	Click <i>Browse</i> to select the location of the service on your local machine.
<i>Service URL</i>	Select the service that is available, paste the required URL for it, then select the adjacent arrow button.

6. In *Annotation Selection*, choose the annotation files from a remote location or add the files from your local system.

To add the files from your local system, choose *+Add Annotation Files* and click *File System*.

7. In *Template Customization*, choose *OData Collection* and *OData Navigation*, then click *Finish*.

3.2.2 Creating a New Extension Project Using the Hybrid Mobile Enablement Feature

When you are developing your application through SAP Cloud Platform mobile service for SAP Fiori (Fiori Mobile), the Hybrid Mobile Enablement feature enables you to create a new hybrid project from a existing source.

The existing application is considered as the parent application. This parent application can reside in the remote repository (SAP Cloud Platform).

3.2.2.1 Creating a New Hybrid Project from a Remote Source

To create a Hybrid Mobile Project for a Fiori application that resides on SAP Cloud Platform, use the *Extension Project* wizard present in SAP Web IDE.

Procedure

1. Click  *File* > *New* > *Extension Project*.
2. Click *Select Application* and choose the server type as *SAP Cloud Platform*.

You may be prompted to configure the *Username* and *Password*.

On successful connection, the server's project repository appears.

The SAP Cloud Platform can be used as a remote source only for apps that have an existing reference library. New reference libraries are not supported.

3. Browse and select a parent application. If the list is long, use the search window to reduce the list.
4. Configure the *Project Configuration* properties:
 - *Original Application*: Use the application name present in the remote repository.
 - *Extension Project*: Retain the default name, which is the *Original Application* name with a *Extension* suffix, or choose a new name.
 - (Optional) *Application Domain*: Use the domain name.
5. Click *Next*.
6. On the *Confirm* page, click *Finish*.

Results

The new project is added to the SAP Web IDE workspace.

3.2.3 Visual Mobile Quality Development

To improve the efficiency of developing hybrid mobile apps, app developers can add mobile qualities to the application and preview them at runtime.

This method offers the following benefits:

- Reduces the gap between design time and runtime when adding mobile qualities to the application, as the mobile qualities can be injected and previewed at runtime itself.
- App developers can easily select the mobile qualities and add them to the application without having to write the code snippets. The code snippets will be generated automatically and injected into the project.

To add mobile qualities to a hybrid mobile application at runtime:

1. Create a new hybrid mobile extension project or Fiori project.
See [Creating a Hybrid Mobile-Enabled Project \[page 12\]](#) and [Creating a New Extension Project Using the Hybrid Mobile Enablement Feature \[page 14\]](#).
2. Go to [Tools](#) > [Mobile Development Pane](#) .
You can see the preview window along with the following panes:
 - Outline area: Lists the event handlers and view elements
 - Plugin area: Lists the customizations added for the Cordova and Kapsel plugins included with the app
 - Snippet area: Displays the code snippets generated based on the event handlers/plugins that are selectedIn the preview window, when you select any control, the [Outline](#) area displays the related controllers.
3. Choose the required controller. The related code snippets will be displayed in the [Snippet](#) area.
4. In the [Plugin](#) area, select the required plugin and customize it by setting the plugin display name, functionality, and so on. Based on the settings the code snippets are generated and displayed in the [Snippet](#) area.

Note

You can also manually edit the generated code snippets in the [Snippet](#) area.

5. Choose [Inject](#) to inject the generated snippets into the application and preview it.
6. After previewing the application, if you are satisfied with the application, choose [Save](#).

3.3 Testing

Eliminate development errors or functional issues by regularly running and previewing your hybrid app. There are different methods to consider, depending on the functional robustness you require.

Context

- **Enablement of Cordova Facade:** If you enable Cordova Facade previews, applications fail gracefully for functions that are not available, and provide alternative implementations for native functionality. To enable Facade and use browser previews for testing, see [Testing Functionality in a Browser with Cordova Facade](#)

[page 16]. Otherwise, to use SAP Web IDE template previews, see [Running a Simple Preview with Hybrid App Project Templates \[page 26\]](#).

- **Testing on an actual device:** The best hybrid app testing takes place on an actual device. See [Testing Your Hybrid Mobile Apps Using the Device Test Cloud Service \[page 26\]](#) and [Using QR Codes to Load and Test Apps on Devices \[page 28\]](#).

3.3.1 Testing Functionality in a Browser with Cordova Facade

To quickly validate hybrid app functionality without deploying the app to an emulator or device, enable previews with Cordova Facade.

Context

Cordova Facade previews take advantage of simulations when functionality is not inherently part of the browser. Simulated functionality is limited to QR code encoding and scanning, camera use, contact lookup, device information collection, obtaining geolocation coordinates, and displaying visual notifications through dialogs.

Restriction

While the Kapsel barcode scanner supports multiple barcode types on different platforms, Cordova Facade preview currently supports QR code and one-dimensional barcode encoding and scanning.

Procedure

1. If you enabled Cordova Facade preview, select any command from the  menu.

In the preview screen, you can use the  button to manage the Cordova Facade preview. Click this button to access the following options:

- Enable/disable Cordova Facade preview
- Go to    Manage the plugins included with the application and configure values for the plugins, as required.

Note

The changes made here are temporary and are applicable only while previewing the application using Cordova Facade preview.

For mobile apps built using Fiori Mobile, only plugins that are included in the app will be displayed in the [Cordova Facade Configuration](#) page. This functionality is currently supported for the Device Orientation plugin.

The application opens in a new tab, and the Cordova plugins are automatically injected into the `index.html` file of the hybrid app. Test the functionality of the plugins; some work similarly as on the emulator or device,

but will simulate functionality using desktop cameras or browser data stores. Other plugins are currently not supported, and you receive a message similar to <plugin-name> has not been implemented yet! both on screen and in the browser console.

2. To evaluate how well the interfaces scales, adjust the browser size (Large, Medium, Small, or Custom), as appropriate.

3.3.1.1 Supported Cordova and Kapsel Plugins

The Cordova Facade currently supports some Cordova and Kapsel plugins.

Context

The supported Cordova plugins are listed here:

Table 5:

Plugin	Description
Camera	Captures photos using the device's camera. For information on Camera API classes and methods, see https://github.com/apache/cordova-plugin-camera .
Contacts	Simulates the device contacts database using the local browser storage. For information on Contacts API classes and methods, see https://github.com/apache/cordova-plugin-contacts .
Device Information	Gathers device-specific information. For information on Device Information API classes and methods, see https://github.com/apache/cordova-plugin-device .
Dialogs	Provides visual device notifications like beep, and vibration. Also displays alert, prompt, and confirmation messages. For information on Dialogs API classes and methods, see https://github.com/apache/cordova-plugin-dialogs .

Plugin	Description
Geolocation	<p>Provides the current location coordinates. Also supports adding predefined locations to the map.</p> <p>To add a location on the map:</p> <ol style="list-style-type: none"> 1. Right-click a project and navigate to  Project Settings  Hybrid App Toolkit  Cordova Facade Configuration. 2. Select a location on the map or enter the location coordinates. 3. Click Add Location. 4. Click Save. <p>To view predefined locations on the map, open the Cordova Facade Configuration window from the application.</p> <p>Choose from the predefined locations or click on a new location on the map to change the location coordinates.</p> <p>To remove a predefined from the map, right-click the location.</p> <p>For information on Geolocation API classes and methods, see .</p>
Device Motion	<p>Provides access to the device's accelerometer.</p> <p>The accelerometer is a motion sensor that detects the change in movement with respect to the current device orientation. It detects the change in three dimensions along the x, y, and z axis.</p> <p>For information on Device Motion API classes and methods, see .</p>
Device Orientation	<p>Provides access to the device's compass which detects the direction in which the device is pointed.</p> <p>For information on Device Orientation API classes and methods, see .</p>

Plugin	Description
MediaCapture	<p>Captures audio, video, and images using the device's media capture capabilities.</p> <p>i Note</p> <p>The Cordova Facade preview supports only the CaptureAudio app.</p> <p>For information on Media Capture API classes and methods, see https://github.com/apache/cordova-plugin-media-capture.</p>
File	<p>Provides read/write access for files residing on the device.</p> <p>This plugin supports the following data paths:</p> <ul style="list-style-type: none"> • applicationDirectory - uses xhr to get local files that are packaged with the app • dataDirectory - saves persistent app-specific data files. • cacheDirectory - caches files that should survive app restarts. Apps should not rely on the OS to delete files present here. <p>It has the following limitations:</p> <ul style="list-style-type: none"> • Does not fail when removing non-empty directories • Does not support metadata for directories • The copyTo and moveTo methods do not support directories
Printer	<p>Enables printing of html docs from Android, iOS and Windows Universal apps.</p> <p>For information on Printer API classes and methods, see https://github.com/katzer/cordova-plugin-printer.</p>

The supported Kapsel plugins are listed here:

Table 6:

Plugin	Description
Barcode Scanner	<p>Encodes and decodes QRCode and one-dimensional barcodes.</p> <p>The Barcode Scanner API Reference provides usage information for Barcode Scanner API classes and methods. See http://help.sap.com/disclaimer?site=https://github.com/phonegap/phonegap-plugin-barcodescanner.</p>

Plugin	Description
Calender	<p>Supports finding, creating, and deleting calender entries.</p> <p>The Calendar API Reference provides usage information for Calendar API classes and methods. See http://help.sap.com/disclaimer?site=https://github.com/EddyVerbruggen/Calendar-PhoneGap-Plugin.</p>
Attachment Viewer	<p>Supports viewing an attachment in the web view without having to download the attachment and open it manually.</p> <p>The plugin temporarily stores a file when you view it, and deletes all the downloaded files when you exit the application. It also ensures the application is not reloaded when you return from the viewer.</p>
EncryptedStorage	<p>Supports storing data locally and securely on the device, so that you do not have to retrieve the data from the server every time the application is opened.</p> <p>In Cordova Façade preview, all data is stored in the browser's local storage. When you clear the browser data, all the data stored previously is removed. Avoid storing sensitive information to ensure it is not removed accidentally.</p> <p>The EncryptedStorage API Reference provides usage information for EncryptedStorage API classes and methods. See http://help.sap.com/Download/Multimedia/html-smp3011sdk/jsdoc/sap.EncryptedStorage.html.</p>
VoiceRecording	<p>Supports recording audio using the native audio recorder application.</p> <p>The plugin includes a native audio recorder application that allows you to record and store audio information in an encrypted format. The maximum quota of the browser's local storage is 5 MB. Once this size is exceeded, you must clear the local storage to proceed with the audio recording.</p> <p>The VoiceRecording API Reference provides usage information for VoiceRecording API classes and methods. See http://help.sap.com/Download/Multimedia/html-smp3011sdk/jsdoc/sap.VoiceRecording.html.</p>
AppUpdate	<p>Supports updating an app to the latest version.</p> <p>Updates to the app are checked when you start the app, resume the app, or explicitly check for updates to the app.</p>

Plugin	Description
Offline OData	<p>Provides offline OData support to Kapsel applications.</p> <p>The Offline OData API Reference provides usage information for Offline OData API classes and methods. See http://help.sap.com/Download/Multimedia/html-smp3011sdk/jsdoc/sap.OData.html.</p>

3.3.1.2 Supportability Matrix for Cordova and Kapsel Plugins

The Hybrid App Toolkit supports several Cordova and Kapsel plugins that can be used while creating a hybrid app.

Table 7: Supportability Matrix for Cordova Plugins Used in Android Apps

Plugin	Build Packaged App Supported? (Applicable only for Hybrid App Toolkit local build)	Hybrid App Toolkit Companion Preview Supported?	Facade Preview Supported?
Device Information	Yes	Yes	Yes
Compass	Yes	Yes	Yes
Media Playback	Yes	Yes	No
Vibration Notification	Yes	Yes	Yes
InApp Browser	Yes	Yes	No
Network Connection	Yes	Yes	No
Geolocation	Yes	Yes	Yes
Access File	Yes	Yes	Yes
Contacts	Yes	Yes	Yes
Debug Console	Yes	Yes	No
Battery Status	Yes	Yes	No
Camera	Yes	Yes (not supported from gallery)	Yes
File Transfer	Yes (file download works, upload requires a remote php server)	Yes (file download works, upload requires a remote php server)	No
Globalization	Yes	Yes	No

Plugin	Build Packaged App Supported? (Applicable only for Hybrid App Toolkit local build)	Hybrid App Toolkit Companion Preview Supported?	Facade Preview Supported?
Accelerometer	Yes	Yes	Yes
Capture	Yes	Yes	Yes (audio only)
Dialog Notification	Yes	Yes	Yes
Splash Screen	Yes	Yes	No

Table 8: Supportability Matrix for Kapsel Plugins Used in Android Apps

Plugin	Build Packaged App Supported?	Hybrid App Toolkit Companion Preview Supported?	Facade Preview Supported?
Logon Manager	Yes	No	No
Encrypted Storage	Yes	Yes	Yes
Barcode Scanner	Yes	Yes	No
Calender	Yes	Yes	Yes
App Update	Yes	No	No
Logger	Yes	No	No
Push	Yes	No	No
Settings	Yes	No	No
Offline OData	Yes	No	No
Voice Recording	Yes	Yes	Yes

Table 9: Supportability Matrix for Cordova Plugins Used in iOS Apps

Plugin	Build Packaged App Supported?	Hybrid App Toolkit Companion Preview Supported?	Facade Preview Supported?
Device Information	Yes	Yes	Yes
Compass	Yes	Yes	Yes
Media Playback	Yes	Yes	No
Vibration Notification	Yes	Yes	Yes
InApp Browser	Yes	Yes	No

Plugin	Build Packaged App Supported?	Hybrid App Toolkit Companion Preview Supported?	Facade Preview Supported?
Network Connection	Yes	Yes	No
Geolocation	Yes	Yes	Yes
Access File	Yes	Yes	Yes
Contacts	Yes	Yes	Yes
Debug Console	Yes	Yes	No
Battery Status	Yes	Yes	No
Camera	Yes	Yes	No
File Transfer	Yes (file download works, upload requires a remote php server)	Yes (file download works, upload requires a remote php server)	No
Globalization	Yes	Yes	No
Accelerometer	Yes	Yes	Yes
Capture	Yes	Yes	No
Dialog Notification	Yes	Yes	Yes
Splash Screen	Yes	Yes	No

Table 10: Supportability Matrix for Kapsel Plugins Used in iOS Apps

Plugin	Build Packaged App Supported?	Hybrid App Toolkit Companion Preview Supported?	Facade Preview Supported?
Logon Manager	Yes	No	No
Encrypted Storage	Yes	Yes	Yes
Barcode Scanner	Yes	Yes	Yes
Calender	Yes	Yes	Yes
App Update	Yes	No	No
Logger	Yes	No	No
Push	Yes	No	No
Settings	Yes	No	No
Offline OData	Yes	No	No

Plugin	Build Packaged App Supported?	Hybrid App Toolkit Companion Preview Supported?	Facade Preview Supported?
Voice Recording	Yes	Yes	Yes

Table 11: Experimental Feature: Supportability Matrix for Cordova Plugins Used in Windows Apps (Applicable only for Hybrid App Toolkit local build)

Plugin	Build Packaged App Supported?	Facade Preview Supported?
Accelerometer	Yes	Yes
BatteryStatus	Only on Windows Phone 8.1	Only on Windows Phone 8.1
Camera	Yes	Yes
Capture	Yes	Yes
Compass	Yes	Yes
Connection	Yes	Yes
Contacts	Partially	Partially
Device	Yes	Yes
Events	Yes	Yes
File	Yes	Yes
File Transfer	No	No
Geolocation	Yes	Yes
InAppBrowser	No	No
Media	Yes	Yes
Notification	Yes	Yes
Splashscreen	Yes	Yes
Status Bar	Only on Windows Phone 8.1	Only on Windows Phone 8.1
Storage	Yes	Yes
Vibration	Only on Windows Phone 8.1	Only on Windows Phone 8.1

Table 12: Experimental Feature: Supportability Matrix for Kapsel Plugins Used in Windows Apps (Applicable only for Hybrid App Toolkit local build)

Plugin	Windows 8.1	Windows 10	Windows Phone 8.1
Logon	Yes	Yes	Yes

Plugin	Windows 8.1	Windows 10	Windows Phone 8.1
AppUpdate	No	No	No
Application Preferences	Yes	Yes	Yes
Attachment Viewer	Yes	Yes	Yes
AuthProxy	Yes	Yes	Yes
Barcode Scanner	Yes	Yes	Yes
Cache Manager	No	No	No
Calender	No	No	No
End-to-end Tracing	Yes	Yes	Yes
Encrypted Storage	Yes	Yes	Yes
Logger	Yes	Yes	Yes
Offline OData	Yes	Yes	No
Online Application	No	No	No
Printer	No	No	No
Push	Yes	Yes	Yes
Settings	Yes	Yes	Yes
Toolbar	Yes	Yes	Yes
Usage	Yes	Yes	Yes
Voice Recording	No	No	No

3.3.2 Running a Simple Preview with Hybrid App Project Templates

When you have included enough functionality to test the user experience of the hybrid app, you can run a simple preview in SAP Web IDE. Hybrid app project templates are available only if you have not enabled Cordova Facade.

Context

Tip

The default hybrid app project templates that are provided with Hybrid App Toolkit contain only a minimum of screen elements, but you can add more as required.

Procedure

1. In the hybrid app project, select an appropriate HTML file (for example, `index.html`), then select the *Run* icon from the menu bar.
2. In the new browser tab, preview the application in several form factors to check the responsive design.
3. Change the device orientation to check how the application responds when changing between landscape and portrait modes.

3.3.3 Testing Your Hybrid Mobile Apps Using the Device Test Cloud Service

Test your hybrid mobile apps on actual mobile devices made available by Device Test Cloud service providers.

Using the Device Test Cloud service you can:

- Enable Device Test Cloud features by selecting the default provider, that is, Perfecto Mobile.
- Select the app to be tested and launch it using the configured Device Test Cloud service.

3.3.3.1 Configuring the Device Test Cloud Service

Use the Device Test Cloud service to test iOS and Android apps on actual mobile devices made available by Device Test Cloud service providers.

Procedure

1. Select ► *Account* ▶ *Device Test Cloud* ▶.

The *Device Test Cloud* page is displayed.

2. Select your service from the *Service provider* drop-down.

Your options are:

Perfecto Mobile Continuous Quality Lab The default setting.

None Use to disable access to the Device Test Cloud service.

3. If you select *Perfecto Mobile Continuous Quality Lab*, the remaining fields on the page are automatically filled in with the required information to use the Perfecto Mobile Continuous Quality Lab service.

4. Click *Save*.

The Device Test Cloud Service is configured.

3.3.3.2 Using the Device Test Cloud Service

Test your iOS or Android apps using a configured Device Test Cloud service.

Prerequisites

Configure your Device Test Cloud service. See *Configuring the Device Test Cloud Service*.

Note

SAP is not responsible for issues encountered while using third-party service providers, including Perfecto Mobile. Please use the support mechanisms provided by your service provider to resolve any issues you encounter with their service. We also recommend that you review the terms and conditions of use for any service provider.

Procedure

1. Build your Cordova App.
2. Click  [Fiori Mobile](#)  [Launch on Device Cloud \(android\)](#)  or [Launch on Device Cloud \(ios\)](#).

The Perfecto Mobile testing page is launched.

Note

The [Launch on Device Cloud](#) menu item is only available if you have already configured the Device Cloud Test service.

3. Choose [Launch Mobile Device And Open Your App](#).

Your application is transferred to Perfecto Mobile and you can then follow the instructions on that site to select devices and test your application.

3.3.4 Using QR Codes to Load and Test Apps on Devices

For convenience, you can generate and use a QR code to load the corresponding preview onto the mobile device. On-device testing is always the most reliable method of validating app features.

Prerequisites

Install the Hybrid App Toolkit Companion on the device.

Note

The iOS version of the Hybrid App Toolkit Companion available on the Apple app store does not support the QR code reader.

Procedure

1. Generate the QR code in SAP Web IDE by clicking the [QR Code](#) button on the top-right corner of the Preview perspective (either Facade-enabled or template previews).
2. When the QR code is generated, open the Hybrid App Toolkit Companion on the device and scan it by:
 - a. Tapping anywhere in the Hybrid App Toolkit Companion to activate the toolbar.
 - b. To scan the generated QR code, from the toolbar, tap [Scan](#). If the QR code scan is successful, a notification sounds and the page is loaded.

3.3.5 Using preview URL to load and test the apps on device

For convenience, you can access the preview URL to load the corresponding preview onto the mobile device. On-device testing is always the most reliable method of validating app features.

Prerequisites

Install the Hybrid App Toolkit Companion on the device.

Procedure

1. In the SAP WebIDE, open the required project.
2. Go to *Run > Show Preview URL*, copy the preview URL.
3. Paste the URL in an email and Send to the device.
4. On the device, enter the Preview URL in the Hybrid app toolkit companion to load and test the application.

3.4 Developer Companion for Fiori Mobile Apps

Hybrid App Toolkit now supports developers to build project-specific companion app in SAP Web IDE for each hybrid app. Application developer needs to trigger a Fiori Mobile build once to build the apk/ipa file for the hybrid app and install the apk/ipa file to the mobile device/simulator/emulator only once.

In the Developer Companion, instead of packaging all the app content, Hybrid App Toolkit will embed the project URL in the SAP Web IDE workspace. Every time you make some changes in the hybrid project, you just need to trigger a reloading in the Developer Companion on the mobile device/simulator/emulator. By doing this, you can quickly check the new runtime behavior.

Everytime when changes are made to the hybrid project in SAP Web IDE, Developer Companion enables you to skip rebuilding and reinstalling the hybrid app. Unlike building a packaged app, you need not deploy the project changes to a Developer Companion to SAP Cloud Platform or register it to a SAP Fiori Launchpad.

i Note

When you build Developer Companion for your packaged app make sure that you build it in the same SAP Web IDE workspace that was used to build the packaged app.

3.4.1 Building Developer Companion for Fiori Mobile Apps

Prerequisites

- Create a hybrid project in SAP Web IDE.
See [Creating a New Project \[page 11\]](#).
- iOS users must create a custom plugin for **cordova-disable-http-cache** on the Fiori Mobile Service Cockpit before building a Developer Companion. If you do not add this custom plugin, you cannot reload the Developer Companion to see the new runtime behaviour.

After creating the custom plugin, do these steps:

1. Choose your hybrid app project, right-click and choose ► *Fiori Mobile* ► *Select Cordova Plugins* ▶.
2. In the *Search Plugin* text box, type the plugin name to search.
3. Select the plugin. In the *Actions* column, select *Add*.
4. In the dialog box that opens, select the *Enable For* check box for iOS. Click *OK*.
5. Click *Save*.

Procedure

1. In SAP Web IDE, choose your hybrid app project, right-click and choose ► *Fiori Mobile* ► *Build* ► *Build Developer Companion* ▶.
2. In the *Configure Fiori Mobile Build Settings* wizard, enter the description in the *App Info* page and choose *Next*.
3. In the *Platform* page, do the steps that follow:

- a. Choose the required mobile platforms

Choose a minimum version for the selected operating system from the Minimum OS Version drop-down list.

Choosing a *Minimum OS Version* helps you to build applications for the specified operating system versions.

➔ Remember

Always use a device/emulator of which the OS version is not lower than the selected Minimum OS Version to install the built app later. Otherwise, the app does not show up in the Mobile Place loaded from the device/emulator.

The recommended minimum operating system version is 5.0 for Android.

- b. Select a valid signing profile for each platform from the drop-down list.

The signing profile used for the previous build is pre-filled. You can use the same signing profile or choose a new one.

1. To create a new signing profile, choose *Create New Signing Profile* from the drop down list
2. Click *OK*. The new signing profile will be added to the drop down list.
3. In the new signing profile dialog box, fill-up all the entries. To understand more about the entries in the dialog box, see [Creating Signing Profiles \[page 34\]](#).

- c. (Optional) Check the *Enable verbose logging* check box.

Checking this box enables the backend build logs to have additional information for both successful and failed builds.

You can review these logs to fine tune your applications and make them perform better. For more information, see [Understanding the Verbose Logs \[page 36\]](#).

- d. (Optional) Check the *Build debug-enabled binaries* check box.

Checking this box enables you to debug your applications from a desktop browser. Android applications can be debugged using Chrome and iOS applications by using Safari.

For debug-enabled Android applications, an internal debug signing profile is used.

After you finish debugging the application, uncheck this option and do a rebuild in order to prepare the application to production.

- e. (Optional) Check the *Disable Privacy Screen* check box.

Privacy Screen feature hides the app content in the app switcher. Checking this box disables you from hiding the app content in the app switcher by which sensitive information can be easily seen.

- f. Click *Next*.

4. (Optional) In the *Push Notification* page, check the box *Enable receiving push notification* if you want your app to support push notification.

If you enable this option, you need to register for the platform for which you develop the app.

For more information on registering push notification messages for Android and iOS, see [Push Notifications for Fiori Mobile Apps \[page 37\]](#).

5. In the *Confirmation* page, verify your selections and choose *Build*.

This will start the build process and the status will be displayed in the SAP Web IDE console.

Once the build completes successfully, a QR code is displayed.

6. Depending on the mobile platform selected for the app, use an Android or iOS phone and scan the QR code to download and install the Developer Companion from SAP Mobile Place.

- o Alternatively, in the QR code screen, you can download the .apk/.ipa file to your local desktop to install Developer Companion to an emulator.

7. Run the Developer Companion in your device.

Content of your hybrid project will be dynamically loaded from the SAP Web IDE workspace.

8. To see the runtime behaviour of the latest changes made to the hybrid project, tap the reload icon in the Fiori Launchpad toolbar.

Fiori mobile app will be reloaded from the SAP Web IDE workspace.

3.5 Deploying Apps to SAP Cloud Platform and Registering to SAP Fiori Launchpad

To build a packaged app using Fiori Mobile build, you must deploy the project to the SAP Cloud Platform and register it on the SAP Fiori Launchpad.

Prerequisites

Create a new project and test it.

See [Creating a New Project \[page 11\]](#).

See [Testing \[page 15\]](#).

Procedure

1. In SAP Web IDE, choose the hybrid app project, right-click and choose ► *Deploy* ► *Deploy to SAP Cloud Platform* ▶.
2. In the *Deploy Application to SAP Cloud Platform* dialog, accept the default values that are displayed and choose *Deploy*.
3. After the app is successfully deployed, choose *Register to SAP Fiori launchpad*.
You need to register the app only once.
4. In the *Register to SAP Fiori Launchpad* dialog, accept the default values displayed in the *General Information* page and choose *Next*.
5. In the *Tile Configuration* page, choose the *Type* as *Static*, enter the *Title* and *Subtitle* for the app, and choose *Next*.
6. In the *Assignment* page, accept the default values that are displayed and choose *Next*.
7. Choose *Finish* to register the app to the Fiori Launchpad.
8. After the app is successfully registered, choose *OK*.
9. If you make changes to the hybrid project, deploy the app again to SAP Cloud Platform, before configuring a Fiori Mobile build. You need not register to SAP Fiori Launchpad again.

3.6 Configuring Apps for Fiori Mobile Build

Prerequisites

Deploy the hybrid app project to the SAP Cloud Platform and register it to the SAP Fiori Launchpad.

See [Deploying Apps to SAP Cloud Platform and Registering to SAP Fiori Launchpad \[page 32\]](#).

Procedure

1. In SAP Web IDE, choose the hybrid app project, right-click and choose ► *Fiori Mobile* ► *Build* ► *Build Packaged App* ▶.
2. In the *Configure Fiori Mobile Build Settings* wizard, enter the description in the *App Info* page and choose *Next*.
3. In the *Platform* page, do the steps that follow:

- a. Choose the required mobile platforms

Choose a minimum version for the selected operating system from the Minimum OS Version drop-down list.

Choosing a *Minimum OS Version* helps you to build applications for the specified operating system versions.

➔ Remember

Always use a device/emulator of which the OS version is not lower than the selected Minimum OS Version to install the built app later. Otherwise, the app does not show up in the Mobile Place loaded from the device/emulator.

The recommended minimum operating system version is 5.0 for Android.

- b. Select a valid signing profile for each platform from the drop-down list.

The signing profile used for the previous build is pre-filled. You can use the same signing profile or choose a new one.

1. To create a new signing profile, choose *Create New Signing Profile* from the drop down list
 2. Click *OK*. The new signing profile will be added to the drop down list.
 3. In the new signing profile dialog box, fill-up all the entries. To understand more about the entries in the dialog box, see [Creating Signing Profiles \[page 34\]](#).
- c. (Optional) Check the *Enable verbose logging* check box.

Checking this box enables the backend build logs to have additional information for both successful and failed builds.

You can review these logs to fine tune your applications and make them perform better. For more information, see [Understanding the Verbose Logs \[page 36\]](#).

- d. (Optional) Check the *Build debug-enabled binaries* check box.

Checking this box enables you to debug your applications from a desktop browser. Android applications can be debugged using Chrome and iOS applications by using Safari.

For debug-enabled Android applications, an internal debug signing profile is used.

After you finish debugging the application, uncheck this option and do a rebuild in order to prepare the application to production.

- e. (Optional) Check the *Disable Privacy Screen* check box.

Privacy Screen feature hides the app content in the app switcher. Checking this box disables you from hiding the app content in the app switcher by which sensitive information can be easily seen.

- f. Click [Next](#).
4. (Optional) In the *Push Notification* page, check the box [Enable receiving push notification](#) if you want your app to support push notification.

If you enable this option, you need to register for the platform for which you develop the app.

For more information on registering push notification messages for Android and iOS, see [Push Notifications for Fiori Mobile Apps \[page 37\]](#).

5. (Optional) In the *Packaging* page, enable the check box [Include additional Fiori applications in this packaged app](#).

By default, this option is disabled to build a standalone app. If you enable this option, you build a multiple tile app.

- a. For a multiple tile app, move the additional tile(s) from the [Available Applications](#) table to the [Selected Applications](#) table. You can do this either by performing a double-click or a drag and drop.

The [Available Applications](#) table lists down the packageable applications that are deployed and registered to the Fiori Launchpad.

6. In the *Confirmation* page, verify your selections and choose [Build](#).

This will start the build process and the status will be displayed in the SAP Web IDE console.

Once the build completes successfully, a QR code is displayed.

7. Depending on the mobile platform selected for the app, use an Android or iOS phone and scan the QR code to install the app from SAP Mobile Place.

After you successfully download and install the app, you can run it on your device. After entering your SAP Cloud Platform credentials on the device, you will see standalone or multiple-tile app with a launchpad bar on the top.

3.6.1 Creating Signing Profiles

You can create new Android and iOS signing profiles that can be used while configuring apps for Fiori Mobile build.

For Android:

You can create a new Android signing profile in one of the following ways:

- Upload a keystore file to create a Android signing profile
- Automatically generate a new Android signing profile (esaier way)

Creating a New Android Signing Profile by Uploading a KeyStore File

To create a new Android signing profile, you need to have the following:

- KeyStore File (in .keystore or .jks format)
- KeyStore Password
- KeyChain Alias

You can generate a keystore file using the Java Keytool command-line utility. To do this, enter the following command at the command-line and follow the instructions.

```
keytool -genkey -keyalg RSA -alias <keychain-aliasname> -keystore <keystore-name>.keystore -storepass <keystore-password> -validity 360 -keysize 2048
```

Generating a New Android signing profile

In this method, the keystore file is automatically generated by Fiori Mobile.

1. Provide values for the mandatory fields *Profile Name*, *Validity* and *Common Name*.
2. Click *Generate*.

For iOS:

To create a new iOS signing profile, you must have the following:

- Signing Certificate (in .p12 format)
- Private Key Pass-phrase
- Provisioning Profile (in .mobileprovisioning format)

To generate a signing certificate in .p12 format, obtain an Apple developer certificate (in .cer format) and export it to the p12 keystore format.

To do this on a Mac system:

1. Open the Keychain Access application in the ► *Applications* ► *Utilities* folder.
2. If you have not already added the certificate to Keychain, go to ► *File* ► *Import* and navigate to the certificate file (.cer file) you obtained from Apple.
3. Select the *Keys* category in Keychain Access.
4. Choose the private key associated with your iPhone Development Certificate. The private key is identified by the iPhone Developer: public certificate that is paired with it.
5. Right-click the iPhone Developer certificate and choose *Export "iPhone Developer: Name..."*.
6. Save your keystore in the Personal Information Exchange (.p12) file format.
7. You will be prompted to create a password that is required when you use the keystore to sign applications or transfer the key and certificate in this keystore to another keystore.

To do this on a Windows system:

1. Download and install OpenSSL.
2. Run the following command-line statement from the OpenSSL bin directory:

```
openssl x509 -in <developer_identity>.cer -inform DER -out <developer_identity>.pem  
-outform PEM
```
3. If you are using a private key from the keychain on a Mac computer, convert it into a PEM key:

```
openssl pkcs12 -nocerts -in <mykey>.p12 -out <mykey>.pem
```
4. You can now generate a valid p12 file, based on the private key and the PEM version of the iOS developer certificate:

```
openssl pkcs12 -export -inkey <mykey>.pem -in <developer_identity>.pem -out  
<iOS_dev>.p12
```

3.6.2 Understanding the Verbose Logs

You can access the verbose logs to review what happened during the build. With this information, you can narrow down to the reasons for failure and make necessary changes to your applications.

During any application build, you can choose to enable verbose logging. Verbose logs can be downloaded for failed builds.

If you do not enable this option, a build log is generated for a failed build. In this build log, you get to know the information like what command was executed and whether the execution was successful or not.

If you enable this option, a build log is generated with more insight on what happened during the command execution.

For example, shown below are samples of a build log and a verbose log for the same application.

Verbose log not enabled:

Sample Code

```
[2017-03-16 07:10:45.29629] [IT] [58ca39a696dcd::i309885@int-w39953f4e] [Finished  
Executing Command: cordova compile] [] [CBS\Utils\Utility::execCommand]  
[2017-03-16 07:10:45.29639] [IT] [58ca39a696dcd::i309885@int-w39953f4e] [Cordova  
Compile Failed] [] [Cordova=>build]  
[2017-03-16 07:10:45.29646] [IT] [58ca39a696dcd::i309885@int-w39953f4e] [Cordova  
Compile End] [] [Cordova=>build]  
[2017-03-16 07:10:45.29655] [IT] [58ca39a696dcd::i309885@int-w39953f4e] [Cordova  
Build End] [] [Cordova=>build]  
[2017-03-16 07:10:45.31582] [ET] [58ca39a696dcd::i309885@int-w39953f4e] [Build  
Failed for [android] [ ErrorCode:20500]] [] [CBS\BuildEntities  
\CordovaBuildEntity::notify]
```

In this case, you can see in the first line of the log, compilation was executed. The build is failed at the end but the reason for failure is not available in this log file.

Verbose log enabled:

Sample Code

```
FAILURE: Build failed with an exception.  
* What went wrong:  
Execution failed for task ':processReleaseManifest'.  
> Manifest merger failed : Attribute application@allowBackup value=(false) from  
AndroidManifest.xml:5:18-45  
    is also present at [:app:WuzeAndroidSDK:] AndroidManifest.xml:16:9-35  
value=(true).  
    Suggestion: add 'tools:replace="android:allowBackup"' to <application>  
element at AndroidManifest.xml:5:5-68:19 to override.  
* Try:  
Run with --stacktrace option to get the stack trace. Run with --info or --debug  
option to get more log output.  
BUILD FAILED
```

In this case, you can see that along with the failure message, the reason for failure is also explained.

3.7 Push Notifications for Fiori Mobile Apps

Push notifications allows your apps to notify a user of new messages or events even when the users are not actively using the apps. To send such push notifications to the users' device, you need to register your app on Google Cloud Messaging (GCM) or Apple Push Notification Service (APNS).

Prerequisites

- You must have Notification User role in SAP Cloud Platform mobile service for development and operations.

Procedure

1. Update your project source codes with appropriate Javascript to register the push success, error and process handlers:
 - a. Refer the following sample code to register the handlers.

Sample Code

```
regSuccess: function(result) {
    jQuery.sap.require("sap.m.MessageBox");
    sap.m.MessageBox.show(
        "Registered For Push Notifications", {
            icon: sap.m.MessageBox.Icon.INFORMATION,
            title: "Success",
            actions: [sap.m.MessageBox.Action.OK]
        });
},
regFailure: function(errorInfo) {
    jQuery.sap.require("sap.m.MessageBox");
    sap.m.MessageBox.show(
        JSON.stringify(errorInfo), {
            icon: sap.m.MessageBox.Icon.INFORMATION,
            title: "Push Registration Failed",
            actions: [sap.m.MessageBox.Action.OK]
        });
},
processNotification: function(notification) {
    if (sap.Push.isPlatformIOS()) {
        var notif_alert = JSON.parse(notification).payload.aps.alert;
        var notif_sound = JSON.parse(notification).payload.aps.sound;
        var notif_badge = JSON.parse(notification).payload.aps.badge;
        var notif_data = JSON.parse(notification).payload.data;
    } else {
        var notif_alert = notification.payload.alert;
        var notif_sound = notification.payload.sound;
        var notif_badge = notification.payload.badge;
        var notif_data = notification.payload.data;
    }
    jQuery.sap.require("sap.m.MessageBox");
    sap.m.MessageBox.show(
        notif_data, {
            icon: sap.m.MessageBox.Icon.INFORMATION,
```

```

        title: notif_alert,
        actions: [sap.m.MessageBox.Action.OK]
    },
),
pushRegistration: function() {
    var nTypes = sap.Push.notificationType.SOUND |
sap.Push.notificationType.ALERT;
    sap.Push.registerForNotificationTypes(nTypes, this.regSuccess,
this.regFailure, this.processNotification);
}

```

- b. Save your changes.
 - c. Deploy your updated project to SAP Cloud Platform. See [Deploying Apps to SAP Cloud Platform and Registering to SAP Fiori Launchpad \[page 32\]](#)
2. For Android:
- To configure push notification messages for an Android app you need to generate a Server API Key and Sender ID with your Google account.
- In the *Configure Fiori Mobile Build Settings* wizard, do the following steps in the *Push Notification* page:
- a. Click the *Copy* button next to the generated Android package name. Then click on the link [Google Developers](#).
 - b. Login with your Google account.
 - c. In the *App name* field, type a desired name for your app.
 - d. In the *Android package name* field, paste the name generated by SAP Web IDE. This name is displayed in the *Push Notification* page.
 - e. Click *Choose and configure services*.
 - f. Click *Cloud Messaging* services.
- Server API Key and Sender ID are generated for your Android app.
- g. Copy and paste the Server API Key and Sender ID in the respective fields in the *Push Notification* page.
 - h. Click *Next*.

3. For iOS:

To configure push notification messages for an iOS app you need generate APNS certificate with Apple Developer account.

- a. **Generating a Certificate Signing Request (.csr):**
 1. In your Mac machine, click Keychain Access application will open up.
 2. Click Request a Certificate From a Certificate Authority
 3. In the *Certificate Information* window, type the email address and name that you used for your Apple Developer account.
 4. Select the option *Saved to disk* and click *Continue*.
 5. Click *Save*. A CSR (Certificate Signing Request) is saved to your local machine.
 6. Click *Done*.
- b. **Registering App ID in Apple Developer Portal:**
 1. Copy the Bundle ID from *Push Notification* page and paste in the field *Bundle ID*.
 2. In the *Push Notification* page of SAP Web IDE, click on the link [Apple Developer](#).

3. Login with your Apple Developer account.
 4. Click on the *Certificates, Identifiers & Profiles* tab.
 5. Click *Identifiers* *App IDs* .
 6. Click on the plus sign to create a new App ID.
 7. Type a name for your app.
 8. In the *App ID Suffix* section, select the option *Explicit App ID*.
 9. In the *App Services* section, enable the check box *Push Notifications* and click *Continue*.
 10. Click *Register* and then *Done*.
 11. Search for the App ID that you created, select it and click *Edit*.
 12. Select *Push Notifications*. In the *Development SSL Certificate* section, click *Create Certificate*.
 13. Click *Continue* in the window that appears.
 14. Click *Choose File* to upload the CSR file that you generated.
 15. Click *Download*. A *aps_development.cer* is downloaded to your local machine.
 16. Click *Done*.
- c. **Generating a Sandbox Push Certificate file:**
1. Double click the *aps_development.cer* to open it.
The certificate is installed in your Keychain application. You can check the certificate in the section *My Certificates*.
 2. Right click on the certificate and choose *Export*.
 3. In the *Save As* field, type a desired name and choose the file format as Personal Information Exchange (.p12). Click *Save*.
 4. Type a password to encrypt your certificate. A new certificate is saved to your local machine.
- d. **Generating Provisioning Profile for Push Application:**
1. On the Apple Developers portal, click *Provisioning Profiles* *All* .
 2. Click on the plus sign to create a new provisioning profile.
 3. In the *Add iOS Provisioning Profiles* wizard, choose *iOS App Development* in the *Select Type* page.
 4. In the *Configure* page, do the steps that follow:
 1. Choose your app from the *App ID* spin box and click *Continue*.
 2. Select the certificate that you want to include and click *Continue*.
 3. Select the devices on which you want to run the application and click *Continue*.
 5. In the *Generate* page, type a name for your profile and click *Continue*.
 6. Download the provisioning profile and click *Done*. A *.mobileprovision* file is downloaded in your local desktop.
- e. **Using Provisioning Profile & Push Certificate in SAP Web IDE:**
1. In the *Configure Fiori Mobile Build Settings* wizard, select *Platform* page.
 2. Enable the check box *iOS* and click *Create Signing Profile* in the drop-down list.
 3. In the *Create iOS Signing Profile* dialog box, type a profile name and upload the signing certificate and provisioning profile. Also, type the password which you used to encrypt the signing certificate.
 4. Click *OK*. Your provisioning profile is added to the list of signing profiles.
 5. In the *Push Notifications* page, for iOS, select *Development (Sandbox)*.
 6. Click *Upload*. In the *Select a Push Certificate* dialog box, upload the push certificate and type the password used for encryption.
 7. Click *Upload*. In the *Push Notifications* page, *Push Certificate Name* and *Expiration Date* fields are populated now.
 8. Continue in the *Configure Fiori Mobile Build Settings* wizard to build the app.

4. Testing Push Notifications

After a successful build with push notification configuration, you can send a test push notification to your app.

- a. Start your app and make sure the code to register the notification handlers (success/error/process) is executed.
- b. Right click on your project in SAP Web IDE workspace and choose ► *Fiori Mobile* ► *Test Push Notification* ▾.
- c. In the dialog box that appears, click *Send*.

Push notification messages are sent to your app. If you see a browser authentication pop-up, enter your SAP Cloud Platform credential.

3.8 Customizing the Theme for Fiori Mobile Apps

You can create your own custom theme package for Fiori Mobile apps.

Procedure

1. In SAP Web IDE, go to ► *Tools* ► *SAP Cloud Platform Cockpit* ▾.
2. In the left pane, go to ► *Services* ► *Portal* ► *Go to Service* ▾.
3. In the left pane, click *Site Directory*.
4. Click *Edit* on the default Fiori launchpad site.

The Fiori Configuration Cockpit opens up.

5. In the left pane of the cockpit, click *Services and Tools*.
6. Open the *UI Theme Designer*.
7. Select a base theme and open the editing tool.
8. Click *SAPUI5 Application Previews*.
9. Enable *SAP Fiori launchpad*.
10. In the customization wizard, edit the features of the theme.
11. Choose ► *Theme* ► *Export* ▾. Save the .zip file in your local desktop.
12. In the Fiori Configuration Cockpit, open the *Theme Manager*.
13. Click the plus symbol at the bottom of the themes list to import the customized theme.
14. Upload the .zip file and click *Save*.
15. Choose the customized theme from the themes list. At the bottom of the screen, choose *Assign to Site*.

3.9 Things to Remember

This section provides you some key points to remember while you build a hybrid mobile app using a Fiori Mobile build.

- For creating template-based Fiori apps, use one of the following templates. Other templates may not support all the expected mobile capabilities.
 - SAP Fiori Master-Detail Application
 - SAP Fiori Worklist Application
 - CRUD Master-Detail Application
 - SAPUI5 Application
 - List Report Application
- For Fiori extension apps, the parent apps must already be deployed to SAP Cloud Platform and use a valid manifest.json file.
- Support apps must be registered with a Fiori Launchpad site that is published and marked as the default site.
- For apps that have Fiori reuse libraries, you must first deploy the libraries to SAP Cloud Platform and verify if you can preview the apps in SAP Web IDE with the libraries loaded.
- Support app localization following UI5 practice.
- The Android apps (minimum OS version lower than 5.0) and iOS apps produced with the Fiori Mobile build cannot be installed and run on the Android emulator or iOS simulator. For testing and debugging your application on the Android emulator or iOS simulator you can preview the app in Hybrid Application Toolkit Companion, or build and run the app through the hybrid app local add-on.
- Refer to the following table to understand the plugins supported in code completion, Cordova Façade preview, Mobile Development Pane and Hybrid App Toolkit Companion.

Table 13:

Code Completion	Cordova Facade Preview	Mobile Development Pane	Hybrid App Toolkit Companion
Battery Status	Device Information	Device Information	Battery Status
Camera	Device Orientation	Device Orientation	Camera
Console	Vibration	Geolocation	Console
Contacts	Geolocation	Contacts	Contacts
Device Information	File	Printer	Device Information
Device Motion	Contacts	Barcode Scanner (Kapsel plugin)	Device Motion
Device Orientation	Camera		Device Orientation
Dialogs	Device Motion		Dialogs
File	Media Capture		File

Code Completion	Cordova Facade Preview	Mobile Development Pane	Hybrid App Toolkit Companion
File Transfer	Dialogs		File Transfer
Printer	Printer		Geolocation
Geolocation	Encrypted Storage (Kapsel plugin)		Globalization
Globalization	Calender (Kapsel plugin)		InAppBrowser
InAppBrowser	Voice Recording (Kapsel plugin)		Media
Media	Barcode Scanner (Kapsel plugin)		Media Capture
Media Capture	Toolbar (Kapsel plugin)		Network Information
Network Information			Splash Screen
Splash Screen			Vibration
Vibration			Barcode Scanner (Kapsel plugin)
Printer			Calender (Kapsel plugin)
Barcode Scanner (Kapsel plugin)			Encrypted Storage (Kapsel plugin)
Calender (Kapsel plugin)			Voice Recording (Kapsel plugin)
AppPreferences (Kapsel plugin)			
AppSettings (Kapsel plugin)			
AppUpdate (Kapsel plugin)			
AttachmentViewer (Kapsel plugin)			
AuthProxy (Kapsel plugin)			
CacheManager (Kapsel plugin)			
E2ETrace (Kapsel plugin)			

Code Completion	Cordova Facade Preview	Mobile Development Pane	Hybrid App Toolkit Companion
Encrypted Storage (Kapsel plugin)			
Logger (Kapsel plugin)			
Logon (Kapsel plugin)			
Odata (Kapsel plugin)			
OfflineStore (Kapsel plugin)			
Push (Kapsel plugin)			
Settings (Kapsel plugin)			
Toolbar (Kapsel plugin)			
Usage (Kapsel plugin)			
Voice Recording (Kapsel plugin)			

4 Developing Apps through Hybrid App Toolkit Local Add-on

The hybrid apps that you develop using SAP Web IDE can be built and deployed through Hybrid App Toolkit local build. To enable the optional local build related features, go to Tools > Preferences > Hybrid Application Toolkit and select the *Enable Local Add-On Features* check box. Click Save.

4.1 Upgrading the Hybrid App Toolkit Add-on for Cloud SAP Web IDE

If you have used a version of Hybrid App Toolkit in the cloud version of SAP Web IDE, and your SAP Web IDE and Hybrid App Toolkit plugin have been upgraded by SAP, you must also upgrade the locally installed add-on files for Hybrid App Toolkit.

Context

The add-on you install locally must match the version of the plugin you have enabled in SAP Web IDE.

Procedure

1. If you have upgraded SAP Web IDE, and enabled the newer version of the Hybrid App Toolkit plugin, upgrade the locally installed components to the same version as the Hybrid App Toolkit plugin.

To upgrade, download and extract the appropriate Windows or Macintosh installer. See [Downloading and Extracting the Windows or Macintosh Installer \[page 53\]](#)

2. If you have upgraded to a new version of a required component (for example, Kapsel), and the new files are installed in a different folder, update all corresponding environment variables (for example `KAPSEL_HOME` and `PATH`) to reflect the new version.

If you upgrade the Kapsel component, you must also update the Kapsel CLI:

```
npm install -g <kapsel folder>/cli
```

4.2 Installing and Setting Up

To use Hybrid App Toolkit as part of the SAP Web IDE environment, validate the prerequisites, then locally install supporting toolkit resources.

1. [Preparing the Cordova Development Environment \[page 45\]](#)

Before you install any Hybrid App Toolkit resources, review all hybrid app development prerequisites.

2. [Downloading and Extracting the Windows or Macintosh Installer \[page 53\]](#)

Run the Hybrid App Toolkit installation assistant to install, set up, and validate SAP Web IDE version 1.21 and Hybrid App Toolkit version 1.13 in your current Cordova development environment.

3. [Enabling the SAP Web IDE Plugin for Hybrid App Toolkit \[page 57\]](#)

Enable the Hybrid App Toolkit optional plugin, so you can use hybrid app customizations for SAP Web IDE. Enabling the plugin makes it part of the SAP Web IDE interface.

4. [Using Custom Cordova Plugins \[page 58\]](#)

Hybrid App Toolkit supports using external custom Cordova plugins in a hybrid application.

4.2.1 Preparing the Cordova Development Environment

Before you install any Hybrid App Toolkit resources, review all hybrid app development prerequisites.

Context

These tools are platform-independent and required for all platforms. Complete these steps before continuing platform-specific preparations.

Procedure

1. Validate environment and connectivity prerequisites.

- Proxy-based connections: If your workstation is connected to the Web via a proxy server, ensure that all development tools in your environment (for example, Git, npm, emulators, and so on) are configured to correctly use that proxy.
- Network configurations: SAP recommends that you use the same network configuration while installing all tools.

2. (Optional) Hybrid App Toolkit has some specific capabilities that leverage capabilities provided by the SAP Mobile Platform Hybrid SDK (Kapsel). Before you can take advantage of those capabilities, you must first install the SMP Hybrid SDK on the system that is running Hybrid App Toolkit.

The manner in which you download the product depends on whether or not you already have a license to download SMP Hybrid SDK version 3.0 SP15:

License	Procedure
Have license	<ol style="list-style-type: none"> 1. Go to https://support.sap.com/swdc. 2. Navigate to Installations and Upgrades > A-Z Alphabetical list > M > SAP Mobile Platform SDK > SAP Mobile Platform SDK 3.0 > Installation. 3. Select the download for the OS your development system is running. The SDK is updated periodically, so you must first download and install the latest SDK release (indicated by a SP## at the end of the version string) then download and install the latest patch release (indicated by the PL## at the end of the version string). 4. Use the installer to install the SDK. 5. Navigate to Installations and Upgrades > A-Z Alphabetical list > M > SAP Mobile Platform SDK > SAP Mobile Platform SDK 3.0 > Support Packages > SAP Mobile Platform SDK 3.0 > select the platform you are working on > select 3.0 SP15. 6. Use the installer to install the SDK patch.
No license	<ol style="list-style-type: none"> 1. Obtain a trial version from the SAP Store at https://store.sap.com/sap/cp/ui/resources/store/html/SolutionDetails.html?pid=0000013098&catID=&pcontry=US&sap-language=EN&cp_id=id-1409756206625-0. 2. Register to receive an email that contains a download link. 3. On the SAP Mobile Platform SDK downloads page, select the download for the OS your development system is running. The SDK is updated periodically, so you must first download and install the latest SDK release (indicated by a SP## at the end of the version string) then download and install the latest patch release (indicated by the PL## at the end of the version string). 4. Use the installer to install the SDK. 5. Select the SAP Mobile Platform SDK SP15 installer for your platform type. 6. Use the installer to install the SDK patch.

→ Tip

For more information about the SDK and how to install it, see https://help.sap.com/viewer/product/SAP_MOBILE_PLATFORM_SDK/3.0.15/en-US.

3. Download and install v5.4.1 of Node.js from <https://nodejs.org/en/blog/release/v5.4.1/>.

- a. Run the installer and follow the prompts in the Install wizard.
- b. On completion, verify that the Node.js folder is in your system environment path.

Open a command prompt, and enter `npm` to verify the Node.js package manager is available. After you issue this command, the terminal window displays the help file for `npm`. If you do not see this, and see an error message instead, it indicates the node installation did not complete successfully. You may need to reboot the system to see the changes, or you may need to reinstall Node.js.

- c. If you will be operating this software in an environment where network access is controlled by a proxy server, you may need to add proxy settings for `npm`. Ask your administrator if you are not sure. If necessary, configure the HTTP and HTTPS proxy for Node.js:

```
npm config set proxy <proxy server:port>
npm config set https-proxy <proxy server:port>
```

Substitute the correct proxy server address and port number for PROXY_ADDRESS[:port] in the commands shown. For example, if your environment directs outbound traffic to a host called proxy using port 8080, you would use these commands:

```
npm config set proxy http://proxy:8080  
npm config set https-proxy http://proxy:8080
```

4. Download and install the latest Java Development Toolkit (JDK) from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. It is required for generating certificates during installation, which allow a secure connection to be setup between your browser and the local tooling environment.

Note

The Android development tools require that you install the entire Java SDK. Installing just the Java Runtime Environment is insufficient.

- a. Follow the installer to install the package.
- b. Create a `JAVA_HOME` environment variable in the `.bash_profile` with this line (for Mac OSX 10.5 or later):

```
export JAVA_HOME=$(/usr/libexec/java_home)
```

For an older Mac OSX, `/usr/libexec/java_home` does not exist. You should set the `JAVA_HOME` to a fixed path such as:

```
export JAVA_HOME=/System/Library/Java/JavaVirtualMachines/1.8.0.jdk/Contents/Home
```

Next Steps

Continue the setup by following the platform specific recommendations that follow: [Preparing Windows for Android Development \[page 48\]](#) or [Preparing Macintosh OS X for iOS and Android Development \(New Installation\) \[page 50\]](#).

Task overview: [Installing and Setting Up \[page 45\]](#)

Next task: [Downloading and Extracting the Windows or Macintosh Installer \[page 53\]](#)

4.2.1.1 Preparing Windows for Android Development

You can build Android applications on Windows, Macintosh, or Linux machines. However, on Windows, you can build only Android apps using the Hybrid App Toolkit.

Procedure

1. Use npm to install Cordova version 6.5.

```
npm install -g cordova@6.5
```

i Note

Do not substitute versions of Cordova; they may trigger compatibility issues.

2. If your workstation is connected to the Web via a proxy server, create a file named `c:\users\<your user>\.plugman\config` and add the same proxy settings you assigned using the `npm` command. (See [Preparing the Hybrid Development Environment \[page 45\]](#).)

The entries must use this syntax:

```
proxy = <proxy server:port>
https-proxy = <proxy server:port>
```

For example:

```
proxy = http://proxy:8080
https-proxy = http://proxy:8080
```

3. Set up the command line Git client.

- a. Download and install the Git client, and ensure it can be used from a Windows command prompt.

SAP recommends: <http://git-scm.com/>.

- b. If you are in an environment where network access is controlled by a proxy server, configure the Git client to use the same proxy configured in previous steps:

```
git config --global http.proxy <proxy server: port>
```

For example:

```
git config --global http.proxy http://proxy.company.com:8080
```

4. (Optional) If you will be leveraging the capabilities of the SMP Hybrid SDK (Kapsel) with Hybrid App Toolkit, create an environment variable called `KAPSEL_HOME` and set the location to the folder that contains the Kapsel plugins.

For example, `c:\SAP\MobileSDK3\KapselSDK`.

5. (Optional) If you are performing Kapsel development, install the Kapsel command-line interface as an administrator:

```
npm install -g C:\SAP\MobileSDK3\KapselSDK\cli
```

6. Restart your command line (`cmd.exe`) to ensure that the latest changes are used.
7. Install required Android tools:

Android Tooling Guidelines	Steps
Apache Ant 1.8 or later	<ol style="list-style-type: none"> 1. Download the package from http://ant.apache.org ↗. 2. Unzip the package to your disk, and follow the instructions in the file. 3. Add the unzipped Ant <code>bin</code> directory path to your Windows PATH environment. 4. Ensure that the <code>ANT_HOME</code> system environment variable has been set to the ANT extraction location.
Android SDK 6.0 (API Level 23)	<ol style="list-style-type: none"> 1. To download the SDK installation package, download the Android Studio from https://developer.android.com/studio ↗. The Android Studio package includes Android SDK and installer. The installer successfully installs Android SDK.
(Optional) Google USB Driver Required when debugging or testing an application over a USB connection to a Google Nexus device.	<p>Either:</p> <ul style="list-style-type: none"> ◦ Install the USB driver directly from the SDK Manager, or, ◦ Download the USB driver from http://developer.android.com/sdk/win-usb.html ↗.

4.2.1.2 Preparing Windows for Android Development (Reinstallation)

For Windows systems that already have Hybrid App Toolkit installed, if you want to upgrade to the latest version of Hybrid App Toolkit or reinstall it, clear the existing files before proceeding to ensure a clean installation.

Procedure

1. Go to the folder where Hybrid App Toolkit is installed. To open a terminal, go to the bottom-left corner of the screen, right-click the Windows icon and choose *Command Prompt*.
2. Turn off the proxy settings:

```
proxy.cmd off
```

3. Clear the bower and cache:

```
npm uninstall -g bower
```

```
npm cache clean
```

If these commands fail, open a new terminal and try again.

4. Remove the node modules and bower components that were already downloaded:

```
delete folder: setup/node_modules  
delete folder: setup/app/bower_components  
delete file: setup/.bowerrc
```

To proceed with the installation, see [Preparing Windows for Android Development \[page 48\]](#).

4.2.1.3 Preparing Macintosh OS X for iOS and Android Development (New Installation)

Use Macintosh OS X machines to develop both iOS and Android applications.

Procedure

1. Use `sudo npm` to install Cordova version 6.5.

```
sudo npm install -g cordova@6.5
```

i Note

Do not substitute versions of Cordova; they may trigger compatibility issues.

2. Create a file named `~/.plugman/config` and add the same proxy settings you assigned using the `npm` command. (See [Preparing the Hybrid/Kapsel Development Environment \[page 45\]](#).)

The entries must use this syntax:

```
proxy = proxy server:port  
https-proxy = proxy server:port
```

For example:

```
proxy = http://proxy:8080  
https-proxy = http://proxy:8080
```

3. (Optional) If you will be leveraging the capabilities of the SMP Hybrid SDK (Kapsel) with Hybrid App Toolkit, create a `KAPSEL_HOME` environment variable in the `.bash_profile` file of your home directory, and set the location to the folder that contains the Kapsel plugins.

For example:

```
export KAPSEL_HOME=/Users/<yourusername>/SAP/MobileSDK3/KapselSDK
```

4. If you are using Kapsel in your development, install the Kapsel command-line interface as an administrator:

```
sudo npm install -g /Users/<yourusername>/SAP/MobileSDK3/KapselSDK/cli
```

5. Install required iOS tools:

iOS Tooling Prerequisite	Steps
Xcode and utilities	<ol style="list-style-type: none"> 1. Install Xcode from either the App Store or Developer Downloads: <ul style="list-style-type: none"> ◦ https://itunes.apple.com/us/app/xcode/id497799835?mt=12 ↗ ◦ https://developer.apple.com/downloads/index.action ↗ 2. Start Xcode. 3. If the command line tools are not yet installed, click ► Preferences ➤ Downloads ↗, then click Install next to Command Line Tools.
(Optional) iOS sim tool, if you want to start the simulator from the command line.	Run: <code>sudo npm install -g ios-sim</code>
(Optional) iOS deploy tool, if you want to deploy the app onto a connected iOS device with USB.	Run: <code>sudo npm install -g ios-deploy</code>
(Optional) iOS signing profile, if you want to build apps for your personal device for testing.	<ol style="list-style-type: none"> 1. Download an iOS developer certificate by following these instructions in the <i>App Distribution Guide</i>: Launching Your App on Devices ↗ 2. Validate the profile setup, by creating a simple Xcode application and running it on your connected device.

6. Install required Android tools:

Android Tooling Prerequisite	Steps
Apache Ant 1.8 or later	<ol style="list-style-type: none"> 1. Download the installation package from http://ant.apache.org ↗. 2. Unzip the package, and follow the instructions in the <code>INSTALL</code> file. 3. Add the unzipped Ant <code>bin</code> directory path to your <code>PATH</code> environment.
Android SDK 6.0 (API level 23)	<ol style="list-style-type: none"> 1. To download the SDK installation package, download the Android Studio from https://developer.android.com/studio ↗. The Android Studio package includes Android SDK and installer. The installer successfully installs Android SDK.

4.2.1.4 Preparing Macintosh OS X for iOS and Android Development (Reinstallation)

For Macintosh OS X systems that already have Hybrid App Toolkit installed, if you want to upgrade to the latest version of Hybrid App Toolkit or reinstall it, clear the existing files before proceeding to ensure a clean installation.

Procedure

1. Go to the folder where Hybrid App Toolkit is installed. Open a terminal by clicking ► *Applications* ► *Utilities* ► *Terminal*.
2. Turn off the proxy settings:

```
sh proxy.sh off
```

3. Delete the HTTP and HTTPS proxy settings for Node.js:

```
npm config delete proxy
```

```
npm config delete https-proxy
```

4. Clear the bower and cache:

```
sudo npm uninstall -g bower
```

```
sudo npm cache clean
```

5. Remove the node modules and bower components that were already downloaded:

```
delete folder: setup/node_modules
```

```
delete folder: setup/app/bower_components
```

```
delete file: setup/.bowerrc
```

To proceed with the installation, see [Running the Hybrid App Toolkit Installer for Macintosh \[page 55\]](#)

4.2.2 Downloading and Extracting the Windows or Macintosh Installer

Run the Hybrid App Toolkit installation assistant to install, set up, and validate SAP Web IDE version 1.21 and Hybrid App Toolkit version 1.13 in your current Cordova development environment.

Procedure

1. Go to https://store.sap.com/sap/cp/ui/resources/store/html/SolutionDetails.html?pid=0000013586&catID=&pcntry=US&sap-language=EN&_cp_id=id-1417532741302-0 and follow the download instructions for latest `SAP_HAT_local.zip` file.
2. Run the installer for your platform type. For details, see either: [Running the Hybrid App Toolkit Installer for Windows \[page 53\]](#) or [Running the Hybrid App Toolkit Installer for Macintosh \[page 55\]](#).

Task overview: [Installing and Setting Up \[page 45\]](#)

Previous task: [Preparing the Cordova Development Environment \[page 45\]](#)

Next task: [Enabling the SAP Web IDE Plugin for Hybrid App Toolkit \[page 57\]](#)

4.2.2.1 Running the Hybrid App Toolkit Installer for Windows

Extract the contents of the downloaded archive and run the installer for this platform. Windows environments are limited to creating Android hybrid apps.

Context

Prerequisites:

➔ Remember

In either of the following scenarios, the installation gives an error and does not continue:

- You are logged on to Windows using an administrator's account
- You are logged on to Windows using a regular user account that has administrator's privileges but the [User Account Control Settings](#) is disabled

To handle these scenarios, ensure the following prerequisites are met:

- You are logged on to Windows using a regular user account that has administrator's privileges
- You have enabled the [User Account Control Settings](#) to notify you when programs try to make changes to the system

Procedure

1. Extract the contents of your downloaded archive to a local folder, and double-click `setup.cmd`.

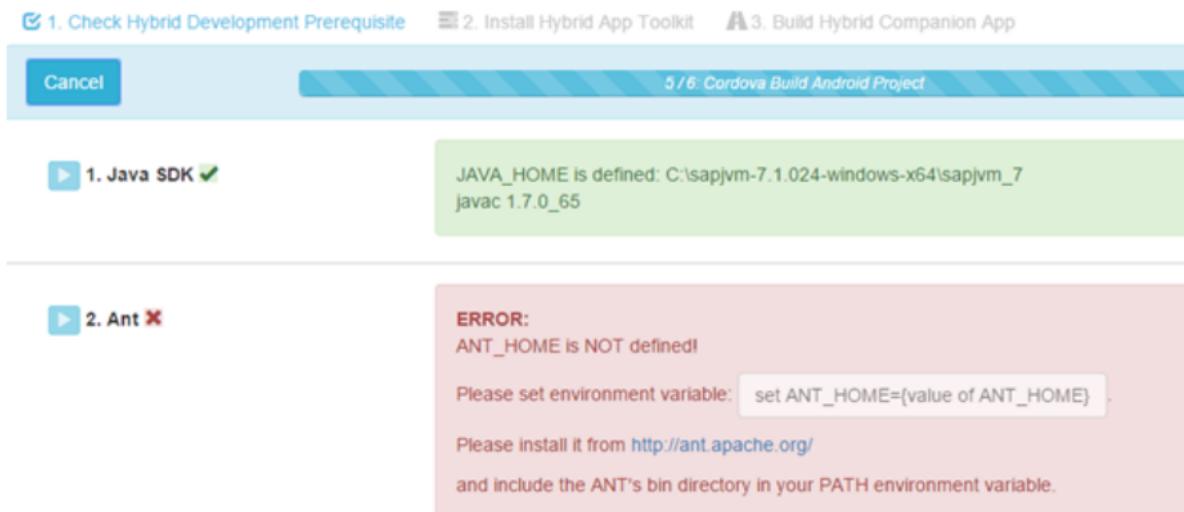
The installer begins to extract and load required packages, before opening in your default browser and prompting you to check installation prerequisites.

The *Important Notes* pop-up window lists the required installation prerequisites. Complete these prerequisites before proceeding with the installation.

2. Choose the platform. If the system is running Windows 8.1/10 operating systems, choose *All*.

3. Click *Check All* in the *Check Hybrid Development Prerequisite* tab.

All prerequisites are itemized and marked as passed or failed. For any unmet prerequisites, issues are color-coded in red as an error and the resolution path is clearly identified.



Use the information provided in a red error box to correct these issues. For example, the figure indicates the `ANT_HOME` environment variable is not defined. Define the variable and restart the system.

Starting with Android 5.1, gradle is used to perform the build. So, you must configure the HTTP(s) proxy for gradle correctly. If the *Cordova build Android project error!* error appears, click *Set proxy setting for /Users/xxxxxx/.gradle/gradle.properties* to set the proxy settings correctly for gradle.

You can accept the recommended settings displayed in the pop-up window or provide new settings and click *SET*.

Repeat this step until all requirements are met and the GUI utility reports *Status: Passed* for *Check Hybrid Development Prerequisites*.

i Note

You cannot proceed to the installation step unless you meet all requirements.

4. Go to *Install Hybrid App Toolkit* and click *Install*.
5. When prompted, enter the SAP IDE URL to be used with the Hybrid App Toolkit Connector, for example, <https://webide-username.dispatcher.neo.ondemand.com/>.

Hybrid App Toolkit Connector uses this URL to set up a secure connection with the SAP Web IDE that is running in your browser.

6. Click *Update*.
7. Enter the password required for the Hybrid App Toolkit Connector to generate a certificate. Confirm the password by entering it again.

Hybrid App Toolkit Connector uses this certificate to connect to SAP Web IDE. You must enter this password each time you start Hybrid App Toolkit Connector.

8. Click *Generate*. When Windows prompts you to allow it to change your computer, select *Yes*.

The installer generates the server certificate and installs it on your machine.

9. Go to *Build Hybrid Companion App* and click *Build*.

Once the build completes, you can use the Hybrid App Toolkit Companion to preview in-development applications that consume native mobile APIs, without building a full Cordova application every time.

The companion container includes a range of popular Cordova plugins and Kapsel plugins that you would commonly use in a hybrid app. See [Testing Functionality with Native Device APIs Using the Hybrid App Toolkit Companion \[page 76\]](#). You can also enable custom Cordova plugins. See [Configuring Custom Plugins for Previewing with SAP Hybrid App Toolkit Companion \[page 59\]](#).

10. For the Hybrid App Toolkit Companion build configuration, retain the default values or modify them as required. Click *Save*.

Once the build completes, the installer generates the required Android .apk file in the <CHAT_HOME>\WebIDECompanion\platforms\Android folder.

11. Click *Exit*.

i Note

You can now rebuild the Hybrid App Toolkit Companion without completely reinstalling the HAT add-on.

4.2.2.2 Running the Hybrid App Toolkit Installer for Macintosh

Extract the contents of the downloaded archive and run the installer for this platform. Macintosh environments can create iOS and Android hybrid apps.

Procedure

1. Start the Terminal application: click .
2. Change to the download folder of the Hybrid App Toolkit archive, and extract the contents of this file.
3. Change to the root folder of the extracted archive contents.
4. Ensure the scripts can be executed by running this command:

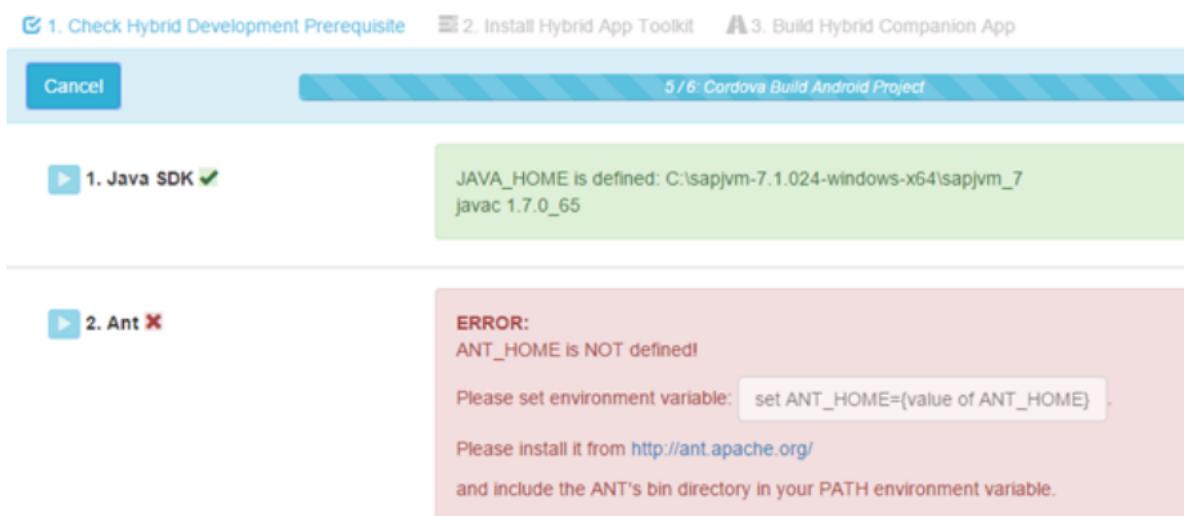
```
chmod +x *.sh
```

5. Run `./setup.sh`.

The installer begins to extract and load required packages, then opens in your default browser.

6. The *Important Notes* pop-up window lists the required installation prerequisites. Start the installation process by first checking prerequisites for each target platform:
 - Choose the device platform for which you want to check prerequisites (*Android*, *iOS* or *All*).
 - Select *Check All*.

All prerequisites are itemized and marked as passed or failed. For any unmet prerequisites, issues are color-coded in red as an error and the resolution path clearly identified.



Use the information provided in a red error box to correct these issues and try again. For example, in the screen shot, click in the `ANT_HOME={value of ANT_HOME}` string and set a new environment variable.

Starting with Android 5.1, gradle is used to perform the build. So, you must configure the HTTP(s) proxy for gradle correctly. If the *Cordova build Android project error* error appears, click *Set proxy setting for /Users/xxxxxx/.gradle/gradle.properties* to set the proxy settings correctly for gradle.

You can accept the recommended settings displayed in the pop-up window or provide new settings and click *SET*.

Repeat this step until all requirements receive a passed status.

i Note

You cannot proceed to the installation step unless you meet all requirements.

7. After you have met all requirements, install the plugin: click the *Install Hybrid App Toolkit* step, then click *Install*.

The installation begins, and each successful step in the process is indicated with a pass status. No action is needed until the installer needs Hybrid App Toolkit Connector configuration properties.

8. When the installer reaches the SAP Web IDE URL definition step, enter the URL for your SAP Web IDE instance (for example, `https://webide-username.dispatcher.neo.ondemand.com/`), then click *Update*.

Hybrid App Toolkit Connector uses this URL to set up a secure connection with the SAP Web IDE that is running in your browser.

- When the installer reaches the Hybrid App Toolkit Connector certificate generation step, type a certificate password and choose *Confirm Password*.

Hybrid App Toolkit Connector uses this certificate to connect to SAP Web IDE. You must enter this password each time you start Hybrid App Toolkit Connector.

- Choose *Generate*, and when Macintosh prompts you to allow it to change your computer, select *Yes*.

The installer generates the server certificate and installs it on your machine.

- Click the *Build Hybrid Companion App* step, and choose *Build*.

Once built, you can use the Hybrid App Toolkit Companion to preview in-development applications that consume native mobile APIs, without building a full Cordova application each time.

The companion container includes a range of popular Cordova plugins and Kapsel plugins that you would commonly use in a hybrid app. See [Testing Functionality with Native Device APIs Using the Hybrid App Toolkit Companion \[page 76\]](#). You can also enable custom Cordova plugins. See [Configuring Custom Plugins for Previewing with SAP Hybrid App Toolkit Companion \[page 59\]](#).

- Either accept the installer defaults or modify the default values as needed, then choose *Save*.

Once the build completes, the installer generates the required device-specific files in the appropriate `<HAT_HOME>\WebIDECompanion\platforms\<type>` folder.

- To exit the installer, in the banner, click *Exit*.

i Note

You can now rebuild the Hybrid App Toolkit Companion without completely reinstalling the HAT add on.

4.2.3 Enabling the SAP Web IDE Plugin for Hybrid App Toolkit

Enable the Hybrid App Toolkit optional plugin, so you can use hybrid app customizations for SAP Web IDE. Enabling the plugin makes it part of the SAP Web IDE interface.

Procedure

- Load the instance of SAP Web IDE you are using:

- For the cloud version, enter the connection URL for either the trial or factory landscape.
- For a locally installed version, use the IP address to load SAP Web IDE in your browser. For example:
`https://<myIPaddress:8080/webide/index.html`.

Once you enable this plugin, always connect to a local SAP Web IDE instance with your IP address over HTTPS.

i Note

To avoid port conflicts, use a port number other than 8080. The location of the port configuration property varies:

- To change the port number for SSL, edit and save `orion.conf`. By default, when you enable SSL for local installed version, the port number is 8443.
Otherwise, the default port is 8080. Edit and save the `orion.ini` file.
2. In SAP Web IDE, select *Tools* > *Preferences* > *Plugins* > *Optional Plugins* .
 3. From the list of optional plugins, locate `com.sap.webide.hybrid` and select *Enabled*.
 4. Select *OK*.
 5. To load the plugin into SAP Web IDE, select the reload button in your browser.

Task overview: [Installing and Setting Up \[page 45\]](#)

Previous task: [Downloading and Extracting the Windows or Macintosh Installer \[page 53\]](#)

Next: [Using Custom Cordova Plugins \[page 58\]](#)

4.2.4 Using Custom Cordova Plugins

Hybrid App Toolkit supports using external custom Cordova plugins in a hybrid application.

Select the custom Cordova plugins from one of the following locations:

- Cordova plugin registry (<https://cordova.apache.org/plugins>
- Local plugins folder

SAP recommends you to select custom plugins from the Cordova plugin registry.

You can build and package an application with custom plugins enabled or preview the custom plugins using SAP Hybrid App Toolkit Companion app.

Note

Cordova Façade preview does not support custom Cordova plugins.

Parent topic: [Installing and Setting Up \[page 45\]](#)

Previous task: [Enabling the SAP Web IDE Plugin for Hybrid App Toolkit \[page 57\]](#)

4.2.4.1 Configuring Custom Plugins for Previewing with SAP Hybrid App Toolkit Companion

To enable custom Cordova plugins, you must configure them while running the Hybrid App Toolkit installer for Windows or Macintosh.

Prerequisites

Copy the custom plugins related files to a local folder on your system.

Procedure

1. When the Hybrid App Toolkit installer is running, in the *Install hybrid App Toolkit* tab, enter a valid path in the *Configure Path for Custom Plugins* step.
2. In the *Build Hybrid Companion App* tab, for adding registry plugins, select plugins from the *Registry* section. For adding local plugins, select plugins from the *Local* section.
Both registry and local plugins can be added together, as required.
3. Click *Save* and continue with the remaining installation.

4.2.4.2 Selecting Custom Cordova Plugins for Building and Packaging an Application

Select the required custom Cordova plugins that must be part of the hybrid application.

Procedure

1. In the project workspace, right-click the project and go to ► *Project Settings* ► *Device Configuration* ► *Custom*.
2. For registry plugins, select plugins from the *Registry* section.
For local plugins, select plugins from the *Local* section.
Hover over any plugin to see the description, plugin id, author (applicable for registry plugins), version (applicable for local plugins), and supported platforms. To add or remove registry plugins, click *Add or Remove*.
3. Click *OK*.
4. Click *Save*.

4.3 Getting Started

Before you start developing hybrid apps, start the Hybrid App Toolkit Connector, strengthen the connection security, decide on a suitable project, and configure the device properties.

1. Starting Hybrid App Toolkit Connector [page 60]

To allow SAP Web IDE to interact with Hybrid App Toolkit add-on components on your local machine over HTTPS, start Hybrid App Toolkit Connector.

2. Strengthening Connection Security [page 62]

(Optional) SAP recommends that you strengthen the connection security between Web IDE and Hybrid App Toolkit Connector.

3. Creating a New Project [page 62]

There are different project creation methods available from SAP Web IDE.

4. Configuring Device Properties [page 74]

Configure device properties to enable mobile capabilities via native device APIs. The values you select are later used to build the application.

4.3.1 Starting Hybrid App Toolkit Connector

To allow SAP Web IDE to interact with Hybrid App Toolkit add-on components on your local machine over HTTPS, start Hybrid App Toolkit Connector.

Procedure

1. Start Hybrid App Toolkit Connector. From the `SAP_HAT_local-<version>` folder:

- On Windows, either double-click `run.cmd`, or enter the same command in a terminal window.
- On Macintosh, in a terminal window, run `chmod +x *.sh`, followed by `./run.sh`.

2. At the prompt, enter the keystore password. This password is the same one you entered during the setup procedure.

During setup a keystore was created for the HTTPS server key of the Hybrid App Toolkit Connector. The Hybrid App Toolkit Connector starts and listens for requests from SAP Web IDE on the default port of 9010.

→ Tip

If the output indicates that the default port is occupied by another program, change the `<port>` value in the `config.json` file, then restart the Hybrid App Toolkit Connector. Valid ports are in the range of 1024 to 65535. All changes to the `config.json` file must be replicated in the preferences for Hybrid App Toolkit Connector in SAP Web IDE. Select Tools > Preferences > Hybrid App Toolkit Connector to change the port value.

For information on the various options present in the `config.json` file, see [Configuration File Options \[page 61\]](#).

3. Configure your browser, as needed:

Browser	Configuration
Chrome	Click <i>Show advanced settings...</i> and select ► <i>Privacy</i> ► <i>Content settings...</i> ► <i>Cookies</i> ► <i>Allow local data to be set</i> ▶. Or add https://localhost:9010 to the exception list.
Safari	Always allow cookies for the local host by selecting ► <i>Preferences</i> ► <i>Privacy</i> ► <i>Cookie and website data</i> ► <i>Always allow</i> ▶.
Firefox	<ol style="list-style-type: none"> Enter https://localhost:9010 in the address bar. In the untrusted connection warning page, click <i>Add Exception</i>. In the <i>Add Security Exception</i> dialog, select <i>Permanently store this exception</i>, then click <i>Confirm Security Exception</i>. Select ► <i>Options</i> ► <i>Privacy</i> ► <i>History</i> ► <i>Remember History</i> ▶. If <i>Use custom settings for history</i> is selected, add https://localhost:9010 to the exceptions list.
Internet Explorer (IE)	<ol style="list-style-type: none"> After loading SAP Web IDE in IE, select <i>Internet options</i>, then choose the <i>Security</i> tab. To add the SAP Web IDE host as a trusted site, select <i>Trusted sites</i>, then click <i>Sites</i>, and add the host name that appears to the list. In <i>Trusted sites</i>, click <i>Custom level</i>. Set <i>Access data sources across domains</i> to <i>Disable</i>.

4. In Web IDE, select ► *Tools* ► *Preferences* ► *Hybrid App Toolkit* ► *Test Connection* ▶.

A message indicates success or failure:

- If successful, a message appears with the Hybrid App Toolkit Connector version number the SAP Web IDE has connected to.
- Otherwise, an error pop-up describes the problem. Follow the resolution details in that message.

Task overview: [Getting Started \[page 60\]](#)

Next task: [Strengthening Connection Security \[page 62\]](#)

4.3.1.1 Configuration File Options

The config.json file contains various options that can be set as per the project requirements.

Table 14: Options Present in the config.json File

Option	Description
commandLog	Specify the command log file name
serverLog	Specify the server log file name
webideHosts	Specify WebIDE host list
port	Specify the port of local server

Option	Description
serverDebug	Specify if server is in debug mode
maxWait	Specify the maximum waiting time for the connection, in minutes
zipSizeLimit	Specify the maximum size of zip file, in MB
cordovaDebug	Specify if the debug mode is on for the Cordova command
customPluginLocalDir	Specify the path of the custom plugins
assistError	Specify if error assistance is available

4.3.2 Strengthening Connection Security

(Optional) SAP recommends that you strengthen the connection security between Web IDE and Hybrid App Toolkit Connector.

Procedure

1. Replace the default <apiKey> value in the project config.json file with an alphanumeric string, chosen by you, that is not easily guessed. You can change some characters in the default string.
2. Restart the Hybrid App Toolkit Connector.
3. In SAP Web IDE, select Tools > Preferences > Hybrid App Toolkit Connector > API Key. Replicate the key value chosen by you there.

Task overview: [Getting Started \[page 60\]](#)

Previous task: [Starting Hybrid App Toolkit Connector \[page 60\]](#)

Next: [Creating a New Project \[page 62\]](#)

4.3.3 Creating a New Project

There are different project creation methods available from SAP Web IDE.

Following are the different methods:

- Create a new project by using one of the templates provided by SAP Web IDE.

- Create a new project using the Hybrid Mobile Enablement feature.

Parent topic: [Getting Started \[page 60\]](#)

Previous task: [Strengthening Connection Security \[page 62\]](#)

Next task: [Configuring Device Properties \[page 74\]](#)

4.3.3.1 Choosing a Project Template

There are different project templates available from SAP Web IDE.

You can create hybrid apps using one of the following templates:

- SAP Fiori Master-Detail Application
See [Creating a Hybrid Mobile-Enabled Project \[page 12\]](#).
- SAP Fiori Worklist Application
To create apps based on this template, you should use the SAPUI5 library version 1.46.7. You can download and use it as a custom library in your device configuration.
To download the SAPUI5 library 1.46.7 version, go to <https://mdocs-sapui5.hanatrial.ondemand.com/mcm/public/v1/open?sh=8175S8OjCafBwqvBwOVQtqLCvrm9t7Vk5q6E0cYdQ>.
See [Creating a Hybrid Mobile-Enabled Project \[page 12\]](#).
- CRUD Master-Detail Application
See [Creating a Hybrid Mobile-Enabled Project \[page 12\]](#).
- SAPUI5 Application
See [Creating a SAPUI5 Application Project \[page 12\]](#).

These templates are mobile-enabled.

You can also create a List Report Application.

See [Creating a List Report Application \[page 13\]](#).

Following are the other available templates.

- SAPUI5 Master Detail Kapsel Offline Application
- SAPUI5 Master Detail Kapsel Application
- SAPUI5 Starter Kapsel Application

At this point, these Kapsel templates are marked as deprecated on the SAP Web IDE Orion version. These Kapsel templates are not available on the SAP Web IDE Innovation (DI) version. For more information, see the SAP blog post <https://blogs.sap.com/2017/06/20/deprecation-of-dedicated-kapsel-templates-for-sap-web-ide-hybrid-app-toolkit/>.

Other application types cannot be used as a template for a hybrid app project.

Note

For the Kapsel plugins to function properly, go to SAP mobile services and set up either the SMP or SAP Cloud Platform mobile service for development and operations environment.

4.3.3.1.1 Creating a SAPUI5 Master Detail Project for Online Apps

The SAPUI5 Master Detail Kapsel Application template includes Kapsel plugins for Logon and AppUpdate.

Procedure

1. In SAP Web IDE, create the hybrid app project by selecting **File** **New** **Project from Template**.
2. Select **SAPUI5 Mobile Application** **SAPUI5 Master Detail Kapsel Application**.
3. In **Basic Information**, set the project name; the domain is optional.

→ Tip

Use lowercase for the project name. This convention keeps the name consistent across several deployment options.

4. In **Template Customization**, enter the *Initial View Details* as required, then click **Finish**.
5. Configure device properties to enable mobile capabilities via native device APIs. The values you select are later used to build the application.

See [Configuring Device Properties \[page 74\]](#).

Results

A new project is created in the SAP Web IDE workspace.

4.3.3.1.2 Creating a SAPUI5 Master Detail Project for Offline Apps

To use offline OData applications with master detail views, choose the **SAPUI5 Master Detail Kapsel Offline Application** template.

Context

⚠ Restriction

If you use SMP3 to proxy the OData service, in the Administration cockpit, you must configure the OData endpoint for the application back-end connection by entering **rewrite URL in SMP** in the **Rewrite mode**.

property. See http://help.sap.com/saphelp_smp305svr/helpdata/en/21/a016f86b0d4fdeb3e6fef802df9b55/content.htm.

Procedure

1. In SAP Web IDE, create the hybrid app project by selecting **File** **New** **Project from Template**. Select Device and Network Connection Cordova plugins. Also, select Logon Manager and Offline OData Kapsel plugins.
2. Select **SAPUI5 Mobile Application** **SAPUI5 Master Detail Kapsel Offline Application**. Two versions of the template are now available. SAPUI5 1.32 is the recommended version and it supports the AppDescriptor. If this template version is selected, the application will have a new structure.
3. In **Basic Information**, set the project name; the domain is optional.

→ Tip

Use lowercase for the project name. This convention keeps the name consistent across several deployment options.

4. In **Data Connection**, choose a service from the list of available sources, then choose a corresponding service.

Each source has a different set of configuration requirements:

Option	Description
Service Catalog	1. Select the system that holds the catalog 2. If the system requires authentication, use the appropriate login credentials. 3. On successful authentication, select the desired catalog name.
Workspace	Expand the top-level folder to browse for the service on your SAP Web IDE workspace.
Filesystem	Click Browse to select the location of the service on your local machine.
Service URL	Select the service that is available, paste the required URL for it, then select the adjacent arrow button.

5. Click **Next**, and select the data collections to cache when the device is offline.

Data collections that you select are provisioned onto the device when the application starts for the first time. The device must be online for that OData store to be created. Once the data store exists, collections are resynchronized when you tap **Refresh**. All the CRUD operations are submitted to the local data store irrespective of whether the device is online or offline. When you tap **Refresh**, it flushes all the local data store changes to the server and retrieves the latest collections from the server to the device. The device user is notified via an icon in the upper-right corner of the master detail app every time the user runs the app in offline mode.

6. Customize the template:

Option	Description
<i>Master Section</i>	<ul style="list-style-type: none"> ◦ <i>Project Namespace</i>: (Optional) Select the namespace for the project. If you do not define one, the wizard uses the project name you defined in step 2. ◦ <i>Title</i>: (Optional) Set the title of the master view of the UI. ◦ <i>OData Collection</i>: (Required) Choose the collection to be shown in the master view of the UI. ◦ <i>Search Placeholder</i>: (Optional) If you do not like the default UI placeholder that labels the search field, replace it with an alternate string. ◦ <i>Search Tooltip</i>: (Optional) If you do not like the like the default tool tip, replace it with an alternate string. ◦ <i>Search Field</i>: (Required) Choose the collection in which the search is performed.
<i>Main Data Fields</i>	(Required) Choose the <i>Item Title</i> , <i>Numeric Attribute</i> , and <i>Units Attribute</i> from the available options in each list.
<i>Details Section</i>	<ul style="list-style-type: none"> ◦ <i>Title</i>: (Optional) Set the title of the details view of the UI. ◦ Add other attributes you want to include in the details view.
<i>Information Section</i>	<p>(Optional) Select a navigation property that is related to the selected Master OData Collection. Navigation appears in the tab panel section directly under the Detail page.</p> <ul style="list-style-type: none"> ◦ <i>OData Navigation</i>: Select one of the collections (based on the collection's <i>NavigationProperty</i>) that you would like to show as support information. By default, if a collection already exists, it is selected for you. ◦ <i>Navigation Attribute <n></i>: Select other properties that you would like to show as support information.
<i>Create/Edit Section</i>	<p>Use these properties for any create or edit operations in the application.</p> <ul style="list-style-type: none"> ◦ To generate an input box for each key property in the <i>Create</i> form, select <i>Create Input for Key Properties</i>. If this option is unselected, key properties are assumed to be autogenerated, and no input box appears. <div data-bbox="641 1208 747 1244" style="background-color: #f2e0b7; border-radius: 5px; padding: 5px; margin-bottom: 10px;"> i Note </div> <div data-bbox="631 1262 1342 1327" style="background-color: #f2e0b7; border-radius: 5px; padding: 10px;"> <p>Do not generate Input boxes for <i>Edit</i> forms; it is assumed key properties are not updatable.</p> </div> <p style="margin-top: 20px;">See The Importance of Key Properties [page 68].</p> <ul style="list-style-type: none"> ◦ To generate a property as a dropdown instead of input box, use <i>Generate Dropdown for Referential Constraints</i> properties. Start by selecting a <i>Dependent Property</i> and its associated <i>Principal Collection</i>, <i>Principal Key Property</i>, and <i>Principal Display Property</i> values. <p style="margin-top: 10px;">See Dropdown Definition Logic [page 67].</p>

7. Confirm the details and click *Finish*.
8. Configure device properties to enable mobile capabilities via native device APIs. The values you select are later used to build the application.

See [Configuring Device Properties \[page 74\]](#).

Results

A new project is created in the SAP Web IDE workspace.

4.3.3.1.2.1 Dropdown Definition Logic

To add a dropdown to your offline OData application's Create or Edit forms, map a dependent property in the master OData collection to another principal collection.

If there is a property in the selected master OData collection that is a dependent property (meaning its value comes from a key property in another collection) then users can follow this definition logic and repeat the steps, as needed, for other dependent properties:

1. Assign the *Dependent Property* by choosing an appropriate value.
2. Select the *Principal Collection* property, which is the collection where the dependent property value will be sourced in the next step.
3. For *Principal Key Property*, select the property in the *Principal Collection* you selected. This property provides the actual value for the dependent property you defined in step 1.
4. For *Principal Display Property*, select a property that represents a more readable format to be shown in the generated dropdown.

For example, to use the `SupplierId` property in a master OData collection called `Products` as the dependent property, select `Suppliers` as the principal collection and `Suppliers.SupplierId` as the principal key property that provides the real value. The more readable value is identified with `Suppliers.SupplierName`. Together these settings result in multiple selectable supplier name, like **Becker Berlin**, appearing in the `SupplierID` dropdown.

The screenshot shows a Fiori Create form for a product. At the top, there is a header bar with the text "Flyer for our new product". Below the header, there are several input fields:

- A label "SupplierId:" followed by a dropdown menu containing the value "Becker Berlin".
- A label "Weight:" followed by an input field containing "0.01".
- A label "WeightUnit:" followed by an input field containing "KG".

At the bottom of the form, there is a dark footer bar with two icons: a circle icon on the left and a square icon on the right.

However, If you do not enter the *Dependent Property* and so on, the `SupplierId` is a normal input box:

Flyer for our new product

SupplierId:

100000044

Weight:

0.01

WeightUnit:

KG

4.3.3.1.2.2 The Importance of Key Properties

Key Property is the property that is marked as Key in your collection. In SQL terms, this key is usually the Primary Key or the Composite Keys of your table. Understand how to use these keys effectively to build your master detail offline Kapsel application.

To create an input box for each key property in the master OData collection, select *Create Input for Key Properties*. Selecting this option generates input boxes only for the *Create* form. Input boxes for key properties are not generated in the *Edit* form.

Example with *Create Input for Key Properties Selected*

ProductId is the key property (primary key) of the Products collection. If you select *Create Input for Key Properties*, a ProductId input box is added to the *Create* form:

Price:

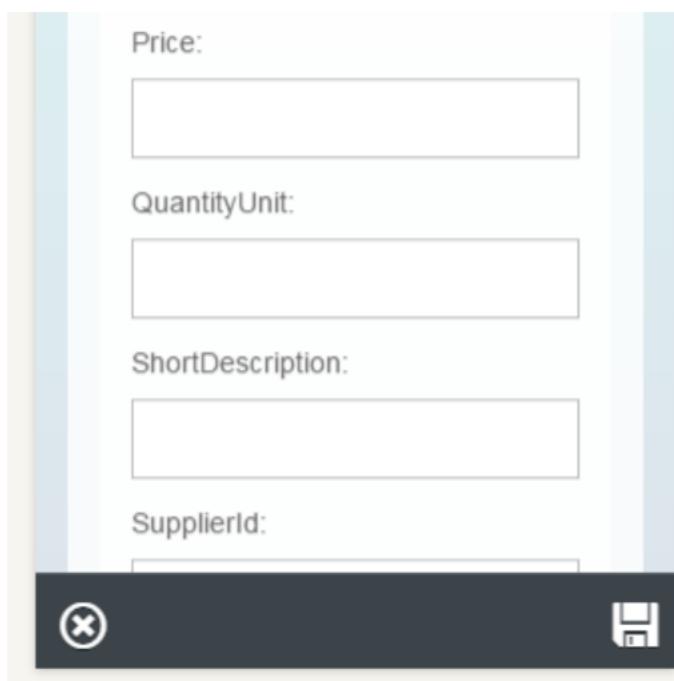
ProductId:

QuantityUnit:

ShortDescription:

Example with *Create Input for Key Properties Not Selected*

If you do not select *Create Input for Key Properties*, *ProductId* input box is not generated:



4.3.3.1.2.3 Error Handling in SAPUI5 Master Detail Kapsel Offline Application

SAPUI5 Master Detail Kapsel Offline application runs with a local offline store on a device. When the device is online, you can synchronize the data between the local offline store and the backend server.

The data synchronization happens in two phases:

1. Data changes from the local offline store are submitted to the backend server
2. Data from the backend server is retrieved and used to refresh the local offline store data

When submitting local data changes to the backend server, it is possible that the local data clashes with server side data. For example, local data might violate server rules, or a local record may no longer exist in the server side.

Following is the workflow for handling synchronization errors:

1. When device is online, open the application in master view and tap *Refresh* to start data synchronization.
First, the locally modified data is pushed by the application to the backend server. After this is completed, the application will read the offline store *ErrorArchive* entity to check if any error has occurred.
If no error has occurred, the application will continue to retrieve data from the backend server and refresh the local offline store.
2. If any error has occurred, data synchronization stops, and the data in the local offline store is not overwritten by server side data.
The `readofflineErrorArchive()` function in `<devlogon.js>` retrieves the synchronization error records from *ErrorArchive* and displays them.
3. Tap any error to view detailed information about the error.

4. Since the data in the local offline store is not overwritten by backend server side data, you can update the local data to fix the synchronization errors. After fixing the errors, tap *Refresh* to synchronize the data again. If there are no further errors, the data synchronization will be successful.
5. If you want to ignore the errors and require the backend server side data to overwrite the local offline store data, tap *Delete* to delete the synchronization errors.
6. Return to the application master view and tap *Refresh* to synchronize the data again and refresh the offline store.

4.3.3.1.3 Creating a SAPUI5 Starter Kapsel Project

Use the `SAPUI5 Starter Kapsel Application` template to create a basic project for developing hybrid apps.

Procedure

1. In SAP Web IDE, create the hybrid app project by selecting *File* *New* *Project from Template*.
2. Select *SAPUI5 Mobile Application* *SAPUI5 Starter Kapsel Application*.
3. In `Basic Information`, set the project name; the domain is optional.

→ Tip

Use lowercase for the project name. This convention keeps the name consistent across several deployment options.

4. Confirm the details and click *Finish*.
5. Configure device properties to enable mobile capabilities via native device APIs. The values you select are later used to build the application.

See [Configuring Device Properties \[page 74\]](#).

Results

A new project is created in the SAP Web IDE workspace.

4.3.3.2 Creating a New Project Using the Hybrid Mobile Enablement Feature

Hybrid Mobile Enablement enables you to create a hybrid mobile application in SAP Web IDE by adapting an existing application as a hybrid mobile application.

The existing application is considered as the parent application. This parent application can reside in your local workspace or in either of the remote repositories (SAPUI5 ABAP System and SAP Cloud Platform).

Hybrid Mobile Enablement provides the following options to create a hybrid project:

- Creating a New Hybrid Project from a Remote Source
See [Creating a New Hybrid Project from a Remote Source \[page 72\]](#).
- Creating a New Hybrid Project from a Local Source
See [Creating a New Hybrid Project from a Local Source \[page 73\]](#).

4.3.3.2.1 Creating a New Hybrid Project from a Remote Source

To create a Hybrid Mobile Project for a Fiori application that resides on a remote SAPUI5 ABAP system or SAP Cloud Platform, use the *Extension Project* wizard present in SAP Web IDE.

Procedure

1. Click  *File*  *New* .
2. Click *Select Application* and choose the server type as *SAPUI5 ABAP System* or *SAP Cloud Platform*.

You may be prompted to configure the *Username* and *Password*.

On successful connection, the server's project repository appears.

The SAP Cloud Platform can be used as a remote source only for apps that have an existing reference library. New reference libraries are not supported.

3. Browse and select a parent application. If the list is long, use the search window to reduce the list.
4. Configure the *Project Configuration* properties:
 - *Original Application*: Use the application name present in the remote repository.
 - *Extension Project*: Retain the default name, which is the *Original Application* name with a **Extension** suffix, or choose a new name.
 - (Optional) *Application Domain*: Use the domain name.
5. Choose *Enable native mobile device functionality*, to create a hybrid mobile-enabled application
6. Click *Next*.
7. On the *Confirm* page, click *Finish*.

Results

The new project is added to the SAP Web IDE workspace.

4.3.3.2.2 Creating a New Hybrid Project from a Local Source

You can create a hybrid project when the parent application is present in the local workspace.

Following are the options:

- Create a new project for a Fiori application
- Change an existing SAPUI5 web project into a hybrid project

4.3.3.2.2.1 Creating a New Hybrid Project for a Fiori Application in Local Workspace

To create a Hybrid Mobile Project for a Fiori application that resides in the local workspace, use *Extension Project* in SAP Web IDE.

Prerequisites

Import a Fiori application to your workspace to be used as the parent application.

Procedure

1. Click  *File* > *New* > *Extension Project*.
2. Click *Workspace*.
3. Browse and select the parent application.
4. Choose *Enable native mobile device functionality*, to create a hybrid mobile-enabled application
5. Click *Next*.
6. On the *Confirm* page, click *Finish*.

Results

The new project is added to the SAP Web IDE workspace.

4.3.3.2.2.2 Changing a SAPUI5 Web Project in Local Workspace into a Hybrid Project

To change an existing SAPUI5 web project that resides in the local workspace into a hybrid project, use *Enable as Mobile App* in SAP Web IDE.

Procedure

1. In the SAP Web IDE workspace, right-click the project and choose *Enable as Mobile App*.
2. Choose *Confirm*.

Results

The existing SAPUI5 web project is changed to a hybrid project.

4.3.4 Configuring Device Properties

Configure device properties to enable mobile capabilities via native device APIs. The values you select are later used to build the application.

Procedure

1. In the project workspace, right-click the project and select ► *Project Settings* ► *Device Configuration* ▾.
2. Select the build mode.
 - For Android apps, go to ► *Settings* ► *Developer Options* ▾ on the Android device, and enable *USB Debugging*. Then type `chrome://inspect` in the Chrome address bar to debug the apps.
For iOS apps, go to ► *Settings* ► *Safari* ► *Advanced* ▾ on the iOS device and enable *Web Inspector*. The device is now available under the *Develop* menu in the Safari browser to debug the apps.
This mode supports debugging of SAPUI5 libraries, as debuggable SAPUI5 library files are packaged with the app.
 - Release mode supports debugging the app on an Android emulator or iOS simulator. This mode supports debugging of SAPUI5 libraries as debuggable SAPUI5 library files are packaged with the app.
Selecting this mode also invokes the SAPUI5 build.
3. (Optional) To provide a custom SAPUI5 library, go to *Build Options* and choose *Use Custom SAPUI5 Library*. In the pop-up, select the required SAPUI5 library and choose *OK*.

i Note

To use a custom SAPUI5 library, first save the library as a zip file and ensure the file name starts with "sapui5-", for example, "sapui5-mobile-custom.zip". Place the file in the local project space folder. Debug mode allows you to debug a built hybrid app directly in the browser by using the browser's debugging capabilities. The app can run either on a device (connected to a PC via a USB cable) or an emulator.

4. Decide whether to use the same icon for Windows, iOS and Android devices. If so, click the ellipsis (...), select the *.PNG file, then click **OK**. Debug mode allows you to debug a built hybrid app directly in the browser. Otherwise, click a specific device tab, browse for and choose a custom icon for each device.
5. (Optional) Decide whether to enable a default or device-specific splash screen for the application:
 - Select **Show Splash Screen** to enable the splash screen display.
 - If you don't need a device-specific splash screen, keep **Use Default Splash Screen** checked. Otherwise, uncheck the box, click each device tab, and configure a unique splash screen for each device.
6. (Optional) For Windows, Android and iOS apps, enable URL scheme binding by providing a value in the **URL Scheme** field. Using this URL scheme, the app can be launched on the corresponding emulator, simulator, or device.
7. Configure application properties:
 - (Required) For the **App Name** and **App ID** properties, enter the exact same values as registered by the administrator on either the on-premise or cloud version of SAP Mobile Platform (SMP).

i Note

The App ID field available in Device Configuration is used for both the application ID and the Cordova package name. Because Cordova packages require a reverse domain syntax (for example, com.sap.demoapp), choose an appropriate value.

- (Optional) Use a **Description**, and set the corresponding **Version** number.
8. Configure the device platforms on which the app will run. First, select the target operating systems (Android, iOS, and Windows) then set other values as appropriate for the application.
 9. Choose the Cordova and Kapsel plugins that must be included with the application during the application build and packaging process.

See *Development Information>Kapsel Development* in <http://help.sap.com/smp3013sdk>.

For adding custom Cordova plugins, see [Selecting Custom Cordova Plugins for Building and Packaging an Application \[page 59\]](#).
 10. (Optional) When you choose the Kapsel Logon Manager plugin, you can customize the Logon UI for your application. Choose **Default Screen** or **Simplified Screen** and customize it as required.
 11. (Optional) If your app uses an image located in SAP Cloud Platform mobile service for development and operations or SMP, enable the **MIME Endpoint** section. Provide the image path and the connection URL for accessing the image.
 12. In SAP mobile services, set up either the SAP Mobile Platform or SAP Cloud Platform mobile service for development and operations environment.
 13. Click **Save**.

Results

All settings are stored in the `.project.json` file.

Task overview: [Getting Started \[page 60\]](#)

Previous: [Creating a New Project \[page 62\]](#)

4.4 Testing

Eliminate development errors or functional issues by regularly running and previewing your hybrid app. There are different methods to consider, depending on the functional robustness you require and which APIs you use.

Following are the different methods:

- [Running a Simple Preview with Hybrid App Project Templates \[page 26\]](#)
- [Testing Functionality in a Browser with Cordova Facade \[page 16\]](#)
- [Testing Functionality with Native Device APIs Using the Hybrid App Toolkit Companion \[page 76\]](#)
- [Using QR Codes to Load and Test Apps on Devices \[page 28\]](#)
- [Using preview URL to load and test the apps on device \[page 29\]](#)

4.4.1 Testing Functionality with Native Device APIs Using the Hybrid App Toolkit Companion

You must configure device settings so the application can properly consume native APIs. To preview the application when it is packaged as a Cordova-based Mobile Web app, use the Hybrid App Toolkit Companion.

Context

Restriction

You cannot use `Run on emulator/device` menu command when testing functionality. They are available only when delivering packages or deploying apps. See [Building and Packaging Apps for Installation \[page 77\]](#).

Procedure

1. In the hybrid app project, select an appropriate HTML file (for example, `index.html`), select  **Run** , then choose from one of the available targets.

If there is more than one device, emulator, or simulator available, select the required target in the *Select Target* window.

The companion container app is installed on the simulator, emulator, or mobile device. On completion, SAP Web IDE passes the URL of the online application to the companion container app to load the online Web assets.

i Note

If the application is using OData sources, they are based on destinations that are defined for the online application that is configured in SAP Cloud Platform.

2. Test the complete functionality of the application.
3. Make changes to your application's code. Save the changes.
4. To display a refresh button, double-tap or click anywhere in the Hybrid App Toolkit Companion to reload the app. To hide the toolbar, double-tap or click again.

If you want a different user to preview the app, click *LOGOUT* and log in again with the new user's credentials.

4.5 Building and Deploying Apps

When you are satisfied with the outcomes of application previews, transfer the application outside the development team for further QA testing cycles or even external release.

[Building and Packaging Apps for Installation \[page 77\]](#)

After you have tested the hybrid app, deploy the app locally, then build and package all assets into a single installable mobile app.

[Deploying Apps to SAP Cloud Platform \[page 79\]](#)

Deploy hybrid apps to SAP Cloud Platform and create a Git repository on the cloud.

[Deploying Apps to SAP Cloud Platform mobile service for development and operations \[page 80\]](#)

Deploy a hybrid app to SAP Cloud Platform mobile service for development and operations so that app updates can be pushed to a registered device with the Kapsel AppUpdate mechanism.

[Deploying to SAP Mobile Platform \(SMP\) \[page 83\]](#)

Deploy hybrid apps to SMP so they can be administered and maintained on premise.

4.5.1 Building and Packaging Apps for Installation

After you have tested the hybrid app, deploy the app locally, then build and package all assets into a single installable mobile app.

Prerequisites:

Packaging a hybrid app for installation can take a significant amount of time. Perform this action only after you have tested and evaluated the app in preview mode in SAP Web IDE, and as a Mobile Web application via the companion container application.

You must also have configured device properties, otherwise no contexts will appear for the *Run on* menu command. Supported contexts are determined by the values you configure. See [Configuring Devices \[page 74\]](#).

For Android Apps:

Android requires all apps to be digitally signed with a certificate before they are installed. Android uses this certificate to identify the author of an app, and the certificate does not need to be signed by a certificate authority. You can sign your app in debug mode during development and in release mode when you are ready to distribute your app. To sign apps in debug mode, the Android SDK generates a certificate. To sign apps in release mode, you can use a test key or generate your own custom key.

1. Open the SAP Web IDE console by selecting ► *View* > *Console* .
When SAP Web IDE successfully completes the deployment, a success popup message appears.
2. Go to ► *Project Setting* > *Device Configuration* , and select *Release Mode*.
3. With the project selected in SAP Web IDE workspace, go to ► *Run* > *Run on* , and choose *Android Emulator* or *Android Device*. If there is more than one emulator or device, select the required target in the *Select Target* window
Review the progress messages that appear. If there are any issues, a popup message appears, which indicates the nature of the problem and corresponding resolution details for it. Correct the issue and try again.
4. Select *Test Key* or *Custom Key*.
5. For test key, enter the following details in pop-up window:
 - Key Store Path: {HAT}/signing/release.keystore
 - Key Store Password: 123456
 - Key Alias: releaseKey
 - Key Alias Password: 123456

For custom key, enter the following command in the command line:

```
keytool -genkey -v -keystore release.keystore -alias releaseKey -keyalg RSA -  
keysize 2048 -validity 20000
```

The custom key is generated and displayed.

Enter the following details in popup window:

- Key Store Path: {HAT}/signing/release.keystore
- Key Store Password: <customer's key store password>
- Key Alias: releaseKey
- Key Alias Password: <customer's key alias password>

6. Click *OK*.

For iOS Apps:

1. Open the SAP Web IDE console, by selecting ► *View* > *Console* .
When SAP Web IDE successfully completes the deployment, you see a "success" message.
2. With the project selected in SAP Web IDE workspace, select ► *Run* > *Run on* , and choose *iOS Simulator* or *iOS Device*. If there is more than one simulator or device, select the required target in the *Select Target* window.
Review the progress messages that appear. If there are any issues, you see a message indicating the nature of the problem and corresponding resolution details for it. Correct the issue and try again.
The project is downloaded to your local disk. Hybrid App Toolkit automatically fetches the specified Cordova plugins to create a Cordova project from the downloaded file.

For Windows Apps:

1. Open the SAP Web IDE console, by selecting  [View](#) > [Console](#) .
- When SAP Web IDE successfully completes the deployment, you see a "success" message.
2. With the project selected in SAP Web IDE workspace, select  [Run](#) > [Run on](#) , and choose [Windows Local](#), [Windows Phone](#) or [Windows Emulator](#). If there is more than one device, select the required target in the [Select Target](#) window.
Review the progress messages that appear. If there are any issues, you see a message indicating the nature of the problem and corresponding resolution details for it. Correct the issue and try again.
The project is downloaded to your local disk. Hybrid App Toolkit automatically fetches the specified Cordova plugins to create a Cordova project from the downloaded file.

Result

Depending on your environment, the [Run on](#) command builds the mobile application for Android and iOS and installs it on simulator, emulator, or mobile device.

4.5.2 Deploying Apps to SAP Cloud Platform

Deploy hybrid apps to SAP Cloud Platform and create a Git repository on the cloud.

Context

Cloud deployment:

- Gives you the option to version (label) your hybrid app, then activate it.
- The activated application's URL (as indicated in the SAP Cloud Platform cockpit) can now be consumed by other mobile applications (such as Fiori Client).

Procedure

1. In the SAP Web IDE workspace, right-click the project, then select  [Deploy](#) > [Deploy to SAP Cloud](#) .
2. Configure the connection properties needed for your environment, and click [OK](#).
You can now see the hybrid app in the SAP Cloud Platform cockpit, and, if you have versioned it, you can activate it. Share the URL of the activated application to any clients that need to consume it.

4.5.3 Deploying Apps to SAP Cloud Platform mobile service for development and operations

Deploy a hybrid app to SAP Cloud Platform mobile service for development and operations so that app updates can be pushed to a registered device with the Kapsel AppUpdate mechanism.

Prerequisites

Table 15: Pre-deployment Checklist for SAP Cloud Platform mobile service for development and operations

Target	Prerequisite	Documented in
SAP Cloud Platform cockpit	<ol style="list-style-type: none">Subscribe to the SAP Cloud Platform mobile service for development and operations service in SAP Cloud Platform cockpit. Ensure you can access the Mobile Service Cockpit.Set up roles:<ul style="list-style-type: none">Choose SAP Cloud Platform mobile service for development and operations from Services, then click the Configure Roles icon.Assign the Administrator role to the user IDs of Web IDE developers.Ensure a mobile services destination with the following properties exists in your account:	<ul style="list-style-type: none">For SAP Cloud Platform account management, see https://help.hana.ondemand.com/hana_cloud_platform_mobile_services/frameset.htm?d2a9afc1681c4e57a4a0f2039274d250.html.For a tutorial on subscribing to a SAP Cloud Platform mobile service for development and operations trial account, see http://scn.sap.com/community/developer-center/mobile-platform/blog/2014/12/18/how-to-enable-hana-cloud-platform-mobile-services-trial

Table 16:

Parameter	Value
Name	mobileservices
Type	HTTP
URL	<Mobile service for development and operations application URL> For example, <a href="https://hcpms-<account>.hanatrial.ondemand.com">https://hcpms-<account>.hanatrial.ondemand.com for trial accounts <a href="https://mobile-<account>-[eu1 us1 ap1].hana.ondemand.com">https://mobile-<account>-[eu1 us1 ap1].hana.ondemand.com for production accounts with Fiori Mobile

Target	Prerequisite	Documented in																						
	<table border="1" data-bbox="425 368 1049 1304"> <thead> <tr> <th data-bbox="425 368 742 422">Parameter</th><th data-bbox="742 368 1049 422">Value</th></tr> </thead> <tbody> <tr> <td data-bbox="425 422 742 615"></td><td data-bbox="742 422 1049 615">https://mobilepreview-<account>.[eu1 us1]ap1].hana.ondemand.com for production accounts with Fiori Mobile Preview</td></tr> <tr> <td data-bbox="425 615 742 691">Proxy Type</td><td data-bbox="742 615 1049 691">Internet</td></tr> <tr> <td data-bbox="425 691 742 745">Authentication</td><td data-bbox="742 691 1049 745">AppToAppSSO</td></tr> <tr> <td colspan="2" data-bbox="425 745 1049 822">Additional Properties</td></tr> <tr> <td data-bbox="425 822 742 929">HTML5.ConnectionTimeoutInSeconds</td><td data-bbox="742 822 1049 929">20</td></tr> <tr> <td data-bbox="425 929 742 1037">HTML5.SocketReadTimeoutInSeconds</td><td data-bbox="742 929 1049 1037">200</td></tr> <tr> <td data-bbox="425 1037 742 1091">WebIDEEnabled</td><td data-bbox="742 1037 1049 1091">true</td></tr> <tr> <td data-bbox="425 1091 742 1145">WebIDEUsage</td><td data-bbox="742 1091 1049 1145">svc_mgmt</td></tr> <tr> <td data-bbox="425 1145 742 1199">HandleRedirects</td><td data-bbox="742 1145 1049 1199">false</td></tr> <tr> <td data-bbox="425 1199 742 1304">SkipSSOTokenGenerationWhenNoUser</td><td data-bbox="742 1199 1049 1304">true</td></tr> </tbody> </table> <p data-bbox="425 1338 1049 1477">The destination is automatically created after you enable Fiori Mobile. You may need to manually add the <code>HTML5.ConnectionTimeoutInSeconds</code>, <code>HTML5.SocketReadTimeoutInSeconds</code>, <code>WebIDEEnabled</code>, and <code>WebIDEUsage</code> properties.</p>	Parameter	Value		https://mobilepreview-<account>.[eu1 us1]ap1].hana.ondemand.com for production accounts with Fiori Mobile Preview	Proxy Type	Internet	Authentication	AppToAppSSO	Additional Properties		HTML5.ConnectionTimeoutInSeconds	20	HTML5.SocketReadTimeoutInSeconds	200	WebIDEEnabled	true	WebIDEUsage	svc_mgmt	HandleRedirects	false	SkipSSOTokenGenerationWhenNoUser	true	
Parameter	Value																							
	https://mobilepreview-<account>.[eu1 us1]ap1].hana.ondemand.com for production accounts with Fiori Mobile Preview																							
Proxy Type	Internet																							
Authentication	AppToAppSSO																							
Additional Properties																								
HTML5.ConnectionTimeoutInSeconds	20																							
HTML5.SocketReadTimeoutInSeconds	200																							
WebIDEEnabled	true																							
WebIDEUsage	svc_mgmt																							
HandleRedirects	false																							
SkipSSOTokenGenerationWhenNoUser	true																							
Mobile Services Cockpit	<p>Application Configuration: In the Mobile Services Cockpit, define and configure an application.</p>	<ul style="list-style-type: none"> For application definitions, see https://help.hana.ondemand.com/hana_cloud_platform_mobile_services/frameset.html?17cff786448442ab2c665ed1cce1505.html. For back-end connections, see https://help.hana.ondemand.com/hana_cloud_platform_mobile_services/frameset.html?21a016f86b0d4fdeb3e6fef802df9b55.html. 																						

Target	Prerequisite	Documented in
Development environment	Kapsel SDK CLI Installation: Ensure you have installed the CLI correctly for your platform type.	Preparing the Cordova Development Environment [page 45] in <i>Installing and Setting Up</i> .
Development environment	Connector Startup: Ensure the Hybrid App Toolkit Connector is running.	Starting Hybrid App Toolkit Connector [page 60] in <i>Getting Started</i> .
SAP Web IDE	Device Configuration: Ensure you have correctly configured the device configuration properties: <ul style="list-style-type: none"> • Set the App ID to the same application ID you have defined for the app in the SAP Cloud Platform mobile service for development and operations cockpit. • Set the version of the app. This version appears as the Development version after the deployment to SAP Cloud Platform mobile service for development and operations succeeds. • Select the mobile platforms on which the built application will run. • Select the required Kapsel plugins. At minimum, you must select <code>Logon Manager</code> and <code>AppUpdate</code>. • Choose SAP Cloud Platform Mobile Services as your mobile service type and input your SAP Cloud Platform mobile service for development and operations host. 	Configuring Device Properties [page 74] in <i>Getting Started</i> .

Context

After you test the hybrid app to ensure that recent code changes run correctly, you can deploy the app to SAP Cloud Platform mobile service for development and operations.

Procedure

1. Right-click the project and select [Deploy](#) [Deploy to HCP Mobile Services](#).
2. Click [Next](#) to package the web assets of project into a `.ZIP` file.
3. Once the packaging succeeds, browse for `packagedKapselApp.zip` in the folder suggested in the dialog.
4. Click [Deploy](#) to deploy the app to the server.

Results

On successful deployment you receive a `Ver x.x.x is deployed` message. If you receive an error message instead, correct the issue and try again.

Next Steps

In the Mobile Services Cockpit, ensure the app version you just deployed is approved and deployed to registered devices. The application update is pushed to the device clients that log on to SAP Cloud Platform mobile service for development and operations by the Kapsel AppUpdate service. To see the new update alert, users may need to press either the Home key (on iOS) or the Back key (on Android) and relaunch the app.

4.5.4 Deploying to SAP Mobile Platform (SMP)

Deploy hybrid apps to SMP so they can be administered and maintained on premise.

Prerequisites

Table 17: Pre-deployment Checklist for SMP

Target	Prerequisite	Documented in
Administration Cockpit	Ensure you define and configure the app in the Administration Cockpit.	<i>Configuring a Hybrid App in the Administration Cockpit</i> on SCN. Go to: http://scn.sap.com/docs/DOC-49592
	<p>Tip</p> <p>The application ID you define must match the application ID you configure in the device configuration of SAP Web IDE.</p>	
Development environment	Kapsel SDK CLI Installation: Ensure you have installed the CLI correctly for your platform type.	Preparing the Cordova Development Environment [page 45] in <i>Installing and Setting Up</i> .
Development environment	Connector Startup: Ensure the Hybrid App Toolkit Connector is running.	Starting Hybrid App Toolkit Connector [page 60] in <i>Getting Started</i> .

Target	Prerequisite	Documented in
SAP Web IDE	<p>Device Configuration: Ensure you have correctly configured the device configuration properties:</p> <ul style="list-style-type: none"> Set the app ID to the same name you have defined for the app in the SMP cockpit. Check the required Kapsel plugins. At minimum, you must select <code>Logon Manager</code> and <code>AppUpdate</code>. Select the mobile platforms on which the built application will run. Set the version of the app. This version appears as the Development version after the deployment to SMP succeeds. Set the mobile platform properties: choose <code>SAP Mobile Platform</code> as your platform and input your host URL. 	Configuring Device Properties [page 74] in Getting Started .

Procedure

- Right-click the project and select ► `Deploy` ► `Deploy to SAP Mobile Platform` ▶.
- Input the appropriate administration account, password, and port (by default, 8083).
- Click `Deploy`.
- Because the deployment to the server can take some time, open the SAP Web IDE console by selecting ► `View` ► `Console` ▶ to evaluate the deployment progress.

Results

On successful deployment, you receive a `Deploy to SAP Mobile Platform is completed` message. The application update is pushed to the device clients that log in to the platform by the Kapsel AppUpdate service. To see the new update alert, users may need to press either the Home key (on iOS) or the Back key (on Android) and relaunch the app.

If you receive an error message in the SAP Web IDE console, correct the issue and try again.

5 Consuming Internet of Things Services

Internet of Things Services enable you to develop, customize, and operate Internet of Things (IOT) business applications in the cloud. You can now consume Internet of Things Services through SAP Web IDE and Hybrid App Toolkit by using a simplified approach to manage different devices, devices types, and message types. SAP Web IDE and Hybrid App Toolkit also support all the IOT APIs.

To consume Internet of Things Services:

1. Enable Internet of Things Services in SAP Cloud Platform cockpit.
2. Assign roles for Internet of Things Services.
3. Configure the destinations for Internet of Things Services Message Management Service and Internet of Things Services Device Management Service.
4. Configure OAuth authentication.
5. Register new devices and define the schema of messages (devices types and message types) that they can send and receive.
6. Inject the IOT library into your project.
7. Consume the IOT APIs.

5.1 Enabling Internet of Things Services

Prerequisites

- You have an SAP user ID and have access to an SAP Cloud Platform developer, customer, or partner account.
- You are an administrator within the account.

See [Account Types](#) of the SAP Cloud Platform documentation.

Procedure

1. Go to the [Services](#) tab in the SAP Cloud Platform cockpit of your account.
2. Select the **Internet of Things Services** tile.
3. Choose [Enable](#) in the detail view of the service.

5.2 Assigning Roles for Internet of Things Services

Internet of Things Services are secured by roles. To use them, you first need to assign the appropriate roles. To grant other users access to the Internet of Things cockpit and the Remote Device Management Service API,

assign the *IoT-User* role to them as described in steps 1-4. In a second necessary procedure, assign the *IoT-User* role for Java Application *iotcockpit* as described in steps 5-8.

Prerequisites

i Note

The user who enabled the Internet of Things Services from the *Services* tab in the SAP Cloud Platform cockpit is assigned the *IoT-User* role automatically.

You are logged on to the SAP Cloud Platform cockpit with your SAP user ID.

Procedure

1. Go to the *Services* tab in your SAP Cloud Platform cockpit.
2. Select the **Internet of Things Services** tile.
3. Choose *Configure Internet of Things Services* in the detail view of the service.
4. Assign the *IoT-User* role to your user.

i Note

For trial accounts, assign the role to your username without the 'trial' suffix.

5. Go to the *Subscriptions* tab in your SAP Cloud Platform cockpit.
6. Select the *iotcockpit* application.
7. Choose the *Roles* tab of the Application details.
8. Assign the *IoT-User* role to your user.

i Note

For trial accounts assign the role to your user name without the 'trial' suffix.

5.3 Configuring Destinations for the Message Management Service and Remote Device Management Service

The Message Management Service (MMS) is the component that is responsible for receiving data from devices and sending messages to devices. The Device Management Service can be used to register new devices, to define the schema of messages (devices types and message types) they can send and receive, as well as to establish the necessary trust relationship needed by devices to interact with MMS.

Create a new destination for MMS using the following values:

Table 18:

Parameter	Value
Name	iotmms
Type	HTTP
Description	Provide a description for your destination
URL	<p><a href="https://<host>/com.sap.iotservices.mms/v1/api/http/">https://<host>/com.sap.iotservices.mms/v1/api/http/</p> <p>To get the host URL, go to Java Applications in the SAP Cloud Platform cockpit, choose the iotrmms application deployed to your SAP Cloud Platform account, and open it. Use the host URL that is displayed.</p>
Proxy Type	Internet
Authentication	NoAuthentication
Additional Properties	
TrustAll	true
WebIDEEnabled	true
WebIDEUsage	svc_mgmt

Create a new destination for the Device Management Service using the following values:

Table 19:

Parameter	Value
Name	iotrdms
Type	HTTP
Description	Provide a description for your destination
URL	<p><a href="https://<host>/com.sap.iotservices.dms/api/">https://<host>/com.sap.iotservices.dms/api/</p> <p>To get the host URL, go to the About dialog present in the top right corner of the Internet of Things Services cockpit next to your username.</p>
Proxy Type	Internet
Authentication	AppToAppSSO
Additional Properties	

Parameter	Value
TrustAll	true
WebIDEEnabled	true
WebIDEUsage	svc_mgnt

5.4 Configuring OAuth Authentication

You can secure the MMS API using OAuth authentication.

Prerequisites

- You have an account with administrator role in SAP Cloud Platform.
- You have developed an OAuth-protected application (resource server).
- You have deployed the application on SAP Cloud Platform.

Context

To use OAuth for authentication, create an OAuth client, define required scopes for MMS and request an OAuth token with the 'push' scope.

Procedure

1. Register a new OAuth client with *Authorization Grant* set to *Client Credentials*.
 - a. In your web browser, log on to the cockpit, and select an account.
 - b. In the *OAuth* section, go to the *Clients* tab.
 - c. Choose *Register new Client*.
 - d. Enter the client data as required (see the table below).

Field	Description
Name	The client name.
Description	A free-text description of the client.

Field	Description
<i>Application</i>	The application for which you are registering this client. To be able to register for a particular application, this account must be subscribed to it.
<i>ID</i>	Required. The ID of the client authorized to access the resource server running on SAP Cloud Platform. If you already have a client with a defined ID at the client device, enter its value here. Otherwise, you can choose <i>Generate ID</i> and <i>Secret</i> to use a system-generated ID, or enter a custom value. In that case, you must provide that value to the user of the client device.
	<p>i Note</p> <p>The client ID must be globally unique within the entire SAP Cloud Platform.</p>
<i>Confidential</i>	If you mark this box, the client ID will be protected with a password. You will need to supply the password here, and provide it to the client.
<i>Secret</i>	A secret (password) that allows the authorization server to authenticate before the client on behalf of the resource owner (user). It will also be needed by the client.
<i>Redirect URI</i>	The application URI to which the authorization server will connect the client with the authorization code.
<i>Token Lifetime</i>	The token lifetime. This value applies to the access token and authorization code.
<i>Refresh Token Lifetime</i>	The refresh token lifetime.
<i>Translations</i>	Optionally, you can provide translations of the client name and description for localization purposes. Choose the <i>Translations</i> button and enter the required language translation there.

- e. Save the new client.
2. Define OAuth scopes for your *iotmms* application using the SAP Cloud Platform cockpit.
- a. In your Web browser, log on to the cockpit, and select an account.
 - b. In the *Java Applications* section, select the OAuth-protected application.
 - c. For the application, go to the *OAuth Scopes* section.
 - d. Choose *New Scope*.
 - e. Enter the scope ID and description.
 - f. Optionally, if you want to provide localization for different languages, choose *Translations* and enter the required data.
 - g. Save the new scope.

The list of scopes depends on the APIs for which you want to enable OAuth access. Currently the following scopes are supported:

- **push** for access to `https://<host>/com.sap.iotservices.mms/v1/api/http/push`
 - **app** for access to `https://<host>/com.sap.iotservices.mms/v1/api/http/app[.svc]`
 - **processing** for access to `https://<host>/com.sap.iotservices.mms/<v1>/api/http/processing`
 - **config** for access to `https://<host>/com.sap.iotservices.mms/<v1>/api/http/config`
3. Request an OAuth token in one of the following ways:
- Use a REST client (See [Enabling OAuth 2.0 Client Credentials Grant](#)) and include the sample code in the REST client. Do an HTTP POST to the token endpoint URL listed for your SAP Cloud Platform account (to get the token endpoint URL, go to the *Branding* section in the *OAuth* tab).

Sample Code

```
POST https://<token endpoint URL>
Content-Type: application/x-www-form-urlencoded
Authorization: Basic <base64-encoded oauth_client_id:oauth_client_secret>

grant_type=client_credentials&
scope=<space separated list of scopes that should be enabled for the
client>
```

- Use the IOT.js library.
In order to use the library to register the OAuth token, go to the *Destination* tab in the SAP Cloud Platform cockpit and configure a destination target to the token endpoint URL.

Table 20:

Parameter	Value
Name	oauth
Type	HTTP
Description	Provide a description for your destination
URL	<code>https://<host>/oauth2/api/</code> To get the token endpoint URL, go to the <i>Branding</i> section in the <i>OAuth</i> tab. You can also add destination in the neo-app.json of your project.
Proxy Type	Internet
Authentication	NoAuthentication

Then, go to the SAP Web IDE landscape and create a simple app that uses the IOT.js library. Include the sample code in the app.

Sample Code

```
IOT.oauth(
  "64a6a808-df74-3e5a-be8e-4753d8a11d2f",    // client id
  "some secret code",                           // client secret
  "app push processing config",                // scopes
```

```
        function(res){
            var token = res.access_token;
            //
        );
    }
);
```

5.5 Managing the Internet of Things Services

Register new devices and define the schema of messages (devices types and message types) they can send and receive.

Go to ► [Tools](#) ► [SAP Cloud Platform IOT Service](#) ▶ to access the Internet of Things Services dialog. All the device types, message types and devices are displayed in a tree format. The top level are device types. Each device type may have some message types and devices under it.

5.5.1 Creating Device Types

Context

A device type specifies a group of devices that share the same specification. In the context of Internet of Things Services a device type is mainly specified by its supported message types (see [Creating Message Types \[page 92\]](#)).

i Note

Please be aware that the number of device types is limited.

Procedure

1. Go to ► [Tools](#) ► [SAP Cloud Platform IOT Service](#) ▶ and choose [Create New Device Type](#).
2. Enter a [Name](#) for the device type.
3. (Optional) Enter a URL to an external application or Web page with further details such as a custom configuration UI or a documentation page. This URL appears as a clickable link in the device type details view.
4. Choose [Save](#).

5.5.2 Deleting Device Types

Context

Device types cannot be deleted if devices of this specific device type still exist. If a device type is deleted, all associated message types are also automatically deleted.

Procedure

1. Go to ► *Tools* > *SAP Cloud Platform IOT Service* and choose the device type you want to delete.
2. Choose the recycle bin button.
3. Choose *OK* to confirm the deletion.

5.5.3 Creating Message Types

Context

Each message type defines a message format including message fields that can be sent or received by a device. Message types are specified for device types.

i Note

You can change the order of message fields by dragging and dropping them with the mouse to the desired position.

Example: A device can send its power consumption. The message can contain 3 fields, the time stamp when the message was created, the ampere and the voltage. In that case, you would create a message type with the name 'Power Consumption' with the direction 'From Device' and would add 3 fields (one time stamp, two double fields for ampere and voltage).

Procedure

1. Go to ► *Tools* > *SAP Cloud Platform IOT Service* and choose a *Device Type*.
2. Choose *New Message Type*.
3. Enter a *Name* for the message type.
4. Select a *Direction* from the dropdown list.
5. Choose *Add Field* and enter the field name, to add fields.

i Note

For type "string" an optional settings field appears where you can set the maximum length of strings for the corresponding message field. The maximum length must be a number between 1 and 5,000. The default value is 255.

6. Choose *Save*.

5.5.4 Deleting Message Types

Context

Message types can be deleted without restrictions

i Note

If a device type is deleted, all associated message types are also automatically deleted.

Procedure

1. Go to  *Tools*  *SAP Cloud Platform IOT Service* and choose the message type you want to delete.
2. Choose the recycle bin button.
3. Choose *OK* to confirm the deletion.

5.5.5 Creating Devices

Prerequisites

Registering a device for Internet of Things services always requires a device type. See [Creating Device Types \[page 91\]](#).

Context

A device is a physical object that is able to send or receive messages.

i Note

Please be aware that the number of devices is limited.

Procedure

1. Go to ► *Tools* ► *SAP Cloud Platform IOT Service* ▾ and choose a *Device Type*.
2. Choose *New Device*.
3. Enter a *Name* for the device.
4. Choose *Save*.

 **Note**

The system displays a pop-up window OAuth Access Token, which shows the token ID generated for the new device. Copy the generated device token since it is needed for secure communication on the device.

5.5.6 Deleting Devices

Context

Devices can be deleted without restrictions.

Procedure

1. Go to ► *Tools* ► *SAP Cloud Platform IOT Service* ▾ and choose the device you want to delete.
2. Choose the recycle bin button.
3. Choose *OK* to confirm the deletion.

5.6 Inject the Internet of Things library into your project

Include the Internet of Things library in your project in order to access the APIs provided by the Internet of Things library.

Prerequisites

Enable the IOT services related features in Hybrid App Toolkit.

To do this, go to ► *Tools* ► *Preferences* ▾ and select the *Enable Internet of Things* checkbox.

Procedure

1. In the SAP Web IDE workspace, choose your project, right-click and choose [Inject Internet of Things API library](#). A new file named IOT.js will be created under your project.
2. Add the following two destinations for the IOT service APIs in the root section of the `neo-app.json` file.

Sample Code

```
"routes": [
  {
    "path": "/iotrdms",
    "target": {
      "type": "destination",
      "name": "iotrdms"
    },
    "description": "IOT Device Management API"
  },
  {
    "path": "/iotmms",
    "target": {
      "type": "destination",
      "name": "iotmms"
    },
    "description": "IOT Message Management API"
  }
]
```

You can now invoke the APIs present in the Internet of Things library.

5.7 Consuming the Internet of Things APIs

Following are the IOT APIs that are supported:

- `IOT.setServiceUrl(dmsUrl, mmsUrl)`
- `IOT.getMmsUrl()`
- `IOT.getDmsUrl()`
- `IOT.getDataTypes(fnComplete, fnFail, options)`
- `IOT.getDeviceTypes(fnComplete, fnFail, options)`
- `IOT.getMessageTypes(fnComplete, fnFail, options)`
- `IOT.getDevices(fnComplete, fnFail, options)`
- `IOT.deleteDeviceType(sId, fnComplete, fnFail, options)`
- `IOT.deleteMessageType(sId, fnComplete, fnFail, options)`
- `IOT.deleteDevice(sId, fnComplete, fnFail, options)`
- `IOT.addDeviceType(sName, fnComplete, fnFail, options)`
- `IOT.addMessageType(oData, fnComplete, fnFail, options)`
- `IOT.addDevice(sName, sDType, fnComplete, fnFail, options)`
- `IOT.getConfig(fnComplete, fnFail, options)`
- `IOT.setConfig(config, fnComplete, fnFail, options)`

- IOT.mapTable(config, fnComplete, fnFail, options)
- IOT.getData (sDevice, fnComplete, fnFail, options)
- IOT.postData (sDevice, sMode, sMessageType, aMessages, fnComplete, fnFail, options)
- IOT.pushData (sDevice, sMethod, sSender, sMessageType, aMessages, fnComplete, fnFail, options)

6 Troubleshooting

Triage your symptom to determine whether it's a known issue (a documented problem with a temporary workaround) or a usage or configuration error for which a permanent solution has been identified.

Context

Start by determining whether you are experiencing a known issue and follow the recommended workaround if one exists. If not, review the usage and configuration error list and see if the symptom is one you can correct.

1. Working Around Known Issues [page 97]

Known issues are documented problems with the toolkit or adverse interactions with the systems and services the toolkit is dependent upon. Review the symptoms of known issues, which require a short-term workaround until the issue is resolved.

2. Resolving Usage or Configuration Errors [page 109]

Locate your symptom and follow the recommended analysis and resolution steps.

3. Reporting an Unknown Issue [page 110]

If you experience an issue that is not documented, report the problem to SAP.

6.1 Working Around Known Issues

Known issues are documented problems with the toolkit or adverse interactions with the systems and services the toolkit is dependent upon. Review the symptoms of known issues, which require a short-term workaround until the issue is resolved.

Error Occurs When Cordova Camera Plugin is Added to the Application [page 100]

When you go to , add the Cordova Camera plugin to the application, and try to build the application, the build might fail.

Issue with Choosing Tabs on the Select Cordova Plugins Screen [page 100]

In the *Select Cordova Plugins* screen, when you try to navigate between the *Selected*, *Recommended*, *Public*, and *Custom* tabs using mouse clicks, it may not work.

Issue with Creating an Project Using SAPUI5 1.28 [page 100]

In SAP Web IDE, when you go to and choose one of the available templates to create an application, if the *SAPUI5 Version* that is selected is *SAPUI5 1.28*, the project may not be created with the correct structure. Some files may be missing from the project.

Issue with Previewing an iOS App Using Xcode version 7 [page 100]

When you create an iOS app using Xcode version 7 and try to preview it on an iPhone 5 device using Hybrid App Toolkit Companion, then the companion app might hang.

Issues with Running Apps Using Xcode version 8 on iOS Devices [page 100]

Running apps using Xcode version 8 on iOS devices may give the following issues.

[Issue with accessing a Fiori Mobile App in the Fiori Launchpad \[page 101\]](#)

When you build your app using Fiori Mobile build and deploy it to the Fiori Launchpad, clicking on the app tile may display the *Could not open app. Please try again later.* error.

[Issue with Displaying a List Report Application Built Using Fiori Mobile Build \[page 102\]](#)

A List Report application that you build using Fiori Mobile build appears different from a List Report application built using Hybrid App Toolkit local build. The application's *Master* page is displayed differently and on iOS devices, the *Details* page does not have a back button to navigate back to the *Master* page.

[App Update Fails on iOS Device \[page 102\]](#)

When you try to update an app developed using SMP Hybrid SDK version SP13 on an iOS device, the update may not be successful.

[Issue with Snippet Pane in Safari Browser \[page 102\]](#)

On a Macintosh system, when you use the ► Tools ▶ *Mobile Development Pane* ▷ option to add mobile qualities to an application, the Snippet pane does not display a scroll bar.

[Issue with Smart Template on iOS Platform \[page 102\]](#)

When you create a hybrid mobile-enabled application using the Smart application template and latest SAPUI5 library version, you may not be able to deploy it on iOS platform. Instead, an authentication dialog may be displayed. This issue does not occur on the Android platform.

[Mobile-Enabled Fiori Master Detail Project Not Responding \[page 102\]](#)

On Hybrid App Toolkit Companion, when you try to preview the mobile-enabled Fiori Master Detail project, the project does not respond. This issue is observed while working with the latest UI5 Fiori template.

[Hybrid App Toolkit Companion Preview Fails for Web Extension Project on iOS Device or Simulator \[page 103\]](#)

On devices and simulators running iOS versions 7.2 and 9.2, when you try to preview a web extension project by scanning the application's barcode, the application will be launched on the Hybrid App Toolkit Companion. When you enter the login credentials and choose *OK*, a blank screen is displayed.

[Hybrid App Toolkit Companion Throws Exception When Home Button is Pressed \[page 103\]](#)

When you try to run a Fiori basic or Fiori Master-Detail application by choosing *Run as SAP Fiori Component*, the application is launched on the Hybrid App Toolkit Companion. But, when you choose Home, Hybrid App Toolkit Companion throws an exception multiple times.

[Voice Recorder Plugin Issue on Windows Platform \[page 103\]](#)

When you choose the Voice Recorder plugin for a Windows app and try to run the app by choosing *Run on Windows Phone*, you are automatically redirected to *Run on Windows Emulator*.

[Access File Plugin Issue \[page 103\]](#)

On Hybrid App Toolkit Companion, when you try to preview an Android or iOS app containing the Access File plugin, the plugin may not work as expected.

[Issue with Previewing an Application on Chrome \[page 103\]](#)

Previewing an application created using the 'SAPUI5 Master Detail Application' template may display a message.

[Deployment Issue with Internet Explorer Versions 10 and 11 \[page 104\]](#)

Deploying an application to SAP Cloud Platform mobile service for development and operations fails the second time.

[Fiori App Issue \[page 104\]](#)

Fiori app gives an error when the app is launched on an Android or iOS device or emulator.

[Issue with Fiori apps created using Shop or Manage Products as sample applications \[page 104\]](#)

Fiori apps created using the Shop or Manage Products as samples do not load correctly.

[VoiceRecording Plugin Issue \[page 104\]](#)

The VoiceRecording plugin does not work correctly in SAP Hybrid App Toolkit Companion or in package deployment.

[Compatibility Issue Occurs in Some Projects \[page 105\]](#)

Compatibility issue occurs in projects created using Hybrid App Toolkit 1.4 and earlier versions.

[Search Field Issue \[page 105\]](#)

Search field does not work correctly in the browser.

[SMP Deployment Issue \[page 106\]](#)

Deploying hybrid apps to SMP fails on a Macintosh system running local SAP Web IDE.

[Hybrid App Toolkit Companion's Back Button on iOS Simulator \[page 106\]](#)

On an iOS simulator, the Hybrid App Toolkit Companion's back button does not work correctly.

[Kapsel Calender Plugin \[page 106\]](#)

The Kapsel Calender plugin does not work correctly on an Android emulator.

[Project Directory Locks Unexpectedly on Android \[page 107\]](#)

After running hybrid app on the Android emulator, a corresponding process, adb.exe process keeps the project directory in the workspace on your local machine in a locked state.

[Android Emulator Fails to Start \[page 107\]](#)

Random startup emulator failures are an issue for hybrid app testing.

[Issue with Barcode Scanner UI in Android Devices \[page 108\]](#)

On your android device, when you tap the button to scan any QR code, the camera does not invoke. If you try to tap the scan button again, you get an alert message which displays "Scan is already in progress".

[Issue with Passcode Page Format \[page 108\]](#)

After installation on mobile devices, some packaged applications display the passcode window with no format. The passcode window will appear as plain page. This can happen for both Android and iOS devices.

[Issue with Downloading .ipa File in Safari Browser \[page 108\]](#)

App installation will fail if the .ipa file has a long and cryptic filename.

[Kapsel Push Plugin \[page 108\]](#)

The Kapsel Push plugin does not work correctly on Android devices.

[Issue with Kapsel Logon Manager Plugin in iOS Simulator \[page 109\]](#)

When you build your app with the Kapsel Logon Manager plugin and run it in on an iOS simulator, the simulator closes the app.

[Task overview: Troubleshooting \[page 97\]](#)

[Next task: Resolving Usage or Configuration Errors \[page 109\]](#)

6.1.1 Error Occurs When Cordova Camera Plugin is Added to the Application

When you go to ► [Project Settings](#) ► [Device Configuration](#), add the Cordova Camera plugin to the application, and try to build the application, the build might fail.

Workaround: Download Cordova Camera plugin version 2.3.0 to a local folder on your system, and add it to the application as a custom plugin. See [Selecting Custom Cordova Plugins for Building and Packaging an Application](#) [page 59].

6.1.2 Issue with Choosing Tabs on the Select Cordova Plugins Screen

In the [Select Cordova Plugins](#) screen, when you try to navigate between the *Selected*, *Recommended*, *Public*, and *Custom* tabs using mouse clicks, it may not work.

Workaround: Use the keyboard keys to navigate between the tabs and choose the required tab.

6.1.3 Issue with Creating an Project Using SAPUI5 1.28

In SAP Web IDE, when you go to ► [File](#) ► [New](#) ► [Project from Template](#) and choose one of the available templates to create an application, if the *SAPUI5 Version* that is selected is *SAPUI5 1.28*, the project may not be created with the correct structure. Some files may be missing from the project.

Workaround: Choose *SAPUI5 1.38* or *SAP Innovation (1.42)* as the SAPUI5 version and create the project.

6.1.4 Issue with Previewing an iOS App Using Xcode version 7

When you create an iOS app using Xcode version 7 and try to preview it on an iPhone 5 device using Hybrid App Toolkit Companion, then the companion app might hang.

There is no work around for this issue.

6.1.5 Issues with Running Apps Using Xcode version 8 on iOS Devices

Running apps using Xcode version 8 on iOS devices may give the following issues.

Issue 1:

When you try to run an app that uses Xcode version 8, on an iPhone 5 device in release mode, it may fail and give the following error:

This application does not support this device's CPU type. AMDeviceSecureInstallApplication(0, device, url, options, install_callback, 0).

Workaround: Use Xcode version 7 and change the iOS-Cordova version in the `config.json` file of the HAT Local Add-on to version 4.1.1 to resolve the issue.

Issue 2:

When you try to run an app that uses XCode version 8 on an iOS simulator, the app may not be launched successfully.

Workaround: Develop the application using any template without the Kapsel Logon plugin and then launch the application.

Issue 3:

When you try to run an app that uses Xcode version 8 on an iOS device, the app may not be launched automatically on the device.

Workaround: Upgrade the iOS Deploy tool to version 1.9.0 and try again.

6.1.6 Issue with accessing a Fiori Mobile App in the Fiori Launchpad

When you build your app using Fiori Mobile build and deploy it to the Fiori Launchpad, clicking on the app tile may display the *Could not open app. Please try again later.* error.

Workaround: Update the SAPUI5 library version to 1.40.x using the following steps:

1. In SAP Web IDE, go to *Tools* *SAP Cloud Platform Cockpit*.
2. In the left pane of the SAP Cloud Platform cockpit, go to *Services* *Portal Service* *Go to Service*.
3. In the *Portal Service* home page, go to your Fiori Launchpad site and click the edit icon.
4. In the left pane, choose *Settings*, go to *System Settings* and verify that the SAPUI5 Version is "Maintenance 1.38".
5. Choose *Edit* and change the SAPUI5 version to "Innovation" from the dropdown list.
6. Click *OK* on the warning message prompt.
7. Choose *Save* and ensure the status changes to "Modified" in the *General* section.
8. Choose the Globe icon in the toolbar, click *Publish* on the warning message prompt and ensure the status changes to "Published" in the *General* section.
9. Go to SAP Web IDE, rebuild the application, and launch it again.

6.1.7 Issue with Displaying a List Report Application Built Using Fiori Mobile Build

A List Report application that you build using Fiori Mobile build appears different from a List Report application built using Hybrid App Toolkit local build. The application's *Master* page is displayed differently and on iOS devices, the *Details* page does not have a back button to navigate back to the *Master* page.

There is no workaround for this issue.

6.1.8 App Update Fails on iOS Device

When you try to update an app developed using SMP Hybrid SDK version SP13 on an iOS device, the update may not be successful.

There is no workaround for this issue.

6.1.9 Issue with Snippet Pane in Safari Browser

On a Macintosh system, when you use the ► *Tools* ► *Mobile Development Pane* ▶ option to add mobile qualities to an application, the Snippet pane does not display a scroll bar.

There is no workaround for this issue.

6.1.10 Issue with Smart Template on iOS Platform

When you create a hybrid mobile-enabled application using the Smart application template and latest SAPUI5 library version, you may not be able to deploy it on iOS platform. Instead, an authentication dialog may be displayed. This issue does not occur on the Android platform.

There is no work around for this issue.

6.1.11 Mobile-Enabled Fiori Master Detail Project Not Responding

On Hybrid App Toolkit Companion, when you try to preview the mobile-enabled Fiori Master Detail project, the project does not respond. This issue is observed while working with the latest UI5 Fiori template.

There is no work around for this issue.

6.1.12 Hybrid App Toolkit Companion Preview Fails for Web Extension Project on iOS Device or Simulator

On devices and simulators running iOS versions 7.2 and 9.2, when you try to preview a web extension project by scanning the application's barcode, the application will be launched on the Hybrid App Toolkit Companion. When you enter the login credentials and choose *OK*, a blank screen is displayed.

Workaround: For devices running iOS version 9.2, choose *Refresh*.

For devices running iOS version 7.2, there is no workaround.

6.1.13 Hybrid App Toolkit Companion Throws Exception When Home Button is Pressed

When you try to run a Fiori basic or Fiori Master-Detail application by choosing *Run as SAP Fiori Component*, the application is launched on the Hybrid App Toolkit Companion. But, when you choose Home, Hybrid App Toolkit Companion throws an exception multiple times.

This issue occurs on Android and iOS devices.

There is no workaround for this issue.

6.1.14 Voice Recorder Plugin Issue on Windows Platform

When you choose the Voice Recorder plugin for a Windows app and try to run the app by choosing *Run on Windows Phone*, you are automatically redirected to *Run on Windows Emulator*.

There is no workaround for this issue.

6.1.15 Access File Plugin Issue

On Hybrid App Toolkit Companion, when you try to preview an Android or iOS app containing the Access File plugin, the plugin may not work as expected.

This is a known limitation. There is no workaround for this limitation.

6.1.16 Issue with Previewing an Application on Chrome

Previewing an application created using the 'SAPUI5 Master Detail Application' template may display a message.

When you create an application using the 'SAPUI5 Master Detail Application' template and try to preview it using Cordova Facade preview, Chrome may ask for permission to store some files on the device. This is expected behavior and Chrome can be allowed to store the files.

There is no workaround for this issue.

6.1.17 Deployment Issue with Internet Explorer Versions 10 and 11

Deploying an application to SAP Cloud Platform mobile service for development and operations fails the second time.

When you use Internet Explorer versions 10 or 11 and deploy an application to SAP Cloud Platform mobile service for development and operations, attempting to deploy the same application again does not work. This is because Internet Explorer locks the application after it is deployed for the first time.

There is no workaround for this issue.

6.1.18 Fiori App Issue

Fiori app gives an error when the app is launched on an Android or iOS device or emulator.

When you create a mobile-enabled Fiori application based on the Shop product and run it on an Android or iOS device or emulator, tapping on an item to open the *Item Detail* page gives the *HTTP request failed* error. This issue does not occur when you preview the app using Web Preview or SAP Hybrid App Toolkit Companion app.

There is no workaround for this issue.

6.1.19 Issue with Fiori apps created using Shop or Manage Products as sample applications

Fiori apps created using the Shop or Manage Products as samples do not load correctly.

When you create a Fiori application using Shop or Manage Products as samples, run the application on an Android device/emulator, and log on, if you click *Register*, an empty blue page is displayed.

There is no workaround for this issue.

6.1.20 VoiceRecording Plugin Issue

The VoiceRecording plugin does not work correctly in SAP Hybrid App Toolkit Companion or in package deployment.

When you run an app containing the VoiceRecording plugin in SAP Hybrid App Toolkit Companion or in package deployment and try to record audio, the plugin does not work correctly. When you deploy the application to an iOS device and tap "Record Voice Memo", error occurs.

Also, when you try to run the application in debug mode on an Android or iOS device/emulator/simulator, a blank page is displayed after the application launches. The application does not launch further.

There is no workaround for this issue.

6.1.21 Compatibility Issue Occurs in Some Projects

Compatibility issue occurs in projects created using Hybrid App Toolkit 1.4 and earlier versions.

Symptom

Projects based on the Basic UI5 Kapsel and SAP Fiori Master Detail Kapsel templates created in SAP Web IDE for Hybrid App Toolkit 1.4 and earlier versions are not compatible with newer versions of Hybrid App Toolkit.

Workaround: After configuring the device settings, open the `index.js` file and do the following changes:

Existing Code:

```
devapp.smplInfo.server = data.hybrid.msType === 1 ? data.hybrid.hcpmsServer : data.hybrid.server;  
devapp.smplInfo.port = data.hybrid.msType === 1 ? "443" : data.hybrid.port;
```

Updated Code:

```
devapp.smplInfo.server = data.hybrid.msType === 0 ? data.hybrid.hcpmsServer : data.hybrid.server;  
devapp.smplInfo.port = data.hybrid.msType === 0 ? "443" : data.hybrid.port;
```

6.1.22 Search Field Issue

Search field does not work correctly in the browser.

Symptom

When you create an application using the SAP UI5 Master Detail Kapsel Offline Template and run the application in the browser, the search field does not give any results.

There is no workaround for this issue.

6.1.23 SMP Deployment Issue

Deploying hybrid apps to SMP fails on a Macintosh system running local SAP Web IDE.

Symptom

When you try to deploy hybrid apps to SMP on a Macintosh system running local SAP Web IDE, it fails and gives the *Cannot read property 'statuscode' of undefined* error or *ECONNRESError*.

There is no workaround for this issue.

6.1.24 Hybrid App Toolkit Companion's Back Button on iOS Simulator

On an iOS simulator, the Hybrid App Toolkit Companion's back button does not work correctly.

Symptom

When you create a hybrid mobile extension project based on the Approve Shopping Cart or Approval Requisition Fiori applications, and run the Hybrid App Toolkit Companion preview on an iOS simulator, clicking the back button does not take you to the previous screen. It takes you to the initial companion app loading screen. This issue does not occur on the Android simulator.

There is no workaround for this issue.

6.1.25 Kapsel Calender Plugin

The Kapsel Calender plugin does not work correctly on an Android emulator.

Symptom

When you run an application on an Android emulator and create an event in the Kapsel Calendar plugin, the event is created successfully. But when you try to find the event or delete the event, the *No matching events exists.* error is displayed.

There is no workaround for this issue.

6.1.26 Project Directory Locks Unexpectedly on Android

After running hybrid app on the Android emulator, a corresponding process, adb.exe process keeps the project directory in the workspace on your local machine in a locked state.

Symptoms

In SAP Web IDE, when you use any run or deploy command, you receive `Failed to remove the hybrid directory` error message in the console, and the command does not complete, because the directory is locked.

Workaround: To unlock the project directory, either:

- Stop the ADB process :
 - On Windows: Press **CTRL** + **ALT** + **DELETE** then select *Start Task Manager*. In the Windows Task Manager, find adb.exe and end the process.
 - On Macintosh: Open a terminal window and enter `ps -o pid= | grep adb` to list all running processes. Make a note of the PID (process id) number associated with the ADB process. Then enter `sudo kill <PID>` to end the process.
- Or, if another program has opened the directory, locate the program, and close it.

Once the process ends, reselect ► *Deploy* ► *Deploy to local Hybrid Toolkit* ▶

6.1.27 Android Emulator Fails to Start

Random startup emulator failures are an issue for hybrid app testing.

Symptom

When you use the *Run on Android Emulator* command in SAP Web IDE, the emulator fails to start.

➔ Recommendation

If you encounter this problem follow this remediation sequence:

1. Check to see if you have enough storage space on the emulator to accommodate the size of the hybrid app you are previewing. If not, clear out user data:
 - Open the AVD manager.
 - Select the target AVD emulator and start it.
 - In the popup dialog, check *Wipe User Data* and click OK.
2. Check the number of Android SDK installed on your machine. If you have multiple instances, check that the system environment PATH variable points to the correct SDK path.

6.1.28 Issue with Barcode Scanner UI in Android Devices

On your android device, when you tap the button to scan any QR code, the camera does not invoke. If you try to tap the scan button again, you get an alert message which displays "Scan is already in progress".

There is no workaround for this issue.

6.1.29 Issue with Passcode Page Format

After installation on mobile devices, some packaged applications display the passcode window with no format. The passcode window will appear as plain page. This can happen for both Android and iOS devices.

There is no workaround for this issue.

6.1.30 Issue with Downloading .ipa File in Safari Browser

App installation will fail if the .ipa file has a long and cryptic filename.

Symptom

After successfully building a packaged app using Fiori Mobile in Hybrid App Toolkit on the Safari browser, clicking the .ipa file link on the [Build Results](#) page will download a file with a long cryptic name instead of the actual file name.

If the filename remains in a long cryptic form even after the download, the app installation fails.

Workaround: The two ways of solving the issue are:

- Rename the long cryptic name to a desired name after download is completed. The renamed .ipa file will install successfully on an iOS Device.
- Alternatively, you can choose to work with Chrome browser.

6.1.31 Kapsel Push Plugin

The Kapsel Push plugin does not work correctly on Android devices.

There is no workaround for this issue.

6.1.32 Issue with Kapsel Logon Manager Plugin in iOS Simulator

When you build your app with the Kapsel Logon Manager plugin and run it in on an iOS simulator, the simulator closes the app.

There is no workaround for this issue.

You can opt to rebuild the app without the Kapsel Logon Manager plugin.

6.2 Resolving Usage or Configuration Errors

Locate your symptom and follow the recommended analysis and resolution steps.

[Hybrid App Toolkit Connector Fails Without Error Message \[page 109\]](#)

An issue exists in the command window with error message recording when Hybrid App Toolkit Connector fails.

[Cordova Development Tooling Connections Fail \[page 110\]](#)

Problem: HTTP and HTTPS connection attempts fail in iOS or Android.

Task overview: [Troubleshooting \[page 97\]](#)

Previous task: [Working Around Known Issues \[page 97\]](#)

Next: [Reporting an Unknown Issue \[page 110\]](#)

6.2.1 Hybrid App Toolkit Connector Fails Without Error Message

An issue exists in the command window with error message recording when Hybrid App Toolkit Connector fails.

Symptoms

If you have this issue you may have detected the problem in these ways:

- You receive a Connection Failure for Hybrid App Toolkit error. The details of the message tell you to check the Hybrid App Toolkit Connector command window for errors. However, after checking this window, you find no error shown in the command window.
- Your browser console reports this error message: `OPTIONS https://localhost:9010/Kapsel net::ERR_INSECURE_RESPONSE`.

Workaround:SAP

Perform a certificate test. Ensure the Hybrid App Toolkit Connector is running and listening on a port 9010. In a Chrome browser, load <https://localhost:9010> in the address bar.

Expected behavior: You see a test page that shows an example of a Chrome address bar.

Triage required: If you see https shown in crossed text with a red lock in the Chrome address bar, this indicates that Hybrid App Toolkit Connector server certificate is not properly installed.

6.2.2 Cordova Development Tooling Connections Fail

Problem: HTTP and HTTPS connection attempts fail in iOS or Android.

Symptom

Connection attempts from tools like Git, npm, and emulators fail over HTTP and HTTPS.

➔ Recommendation

Check and resolve:

1. Proxy settings. First determine if your environment controls network access with a proxy server. Speak to your administrator if you are unsure. If you do use a proxy, ensure that you have configured all development tools in your environment to correctly use that proxy. For proxy setting details, see either [Preparing Windows for Android Development \[page 48\]](#) or [Preparing Macintosh OS X for iOS and Android Development \(New Installation\) \[page 50\]](#).
2. Network connections. If you do not use a proxy, and you still have issues with a subset of your tools, check the network configuration for each. All tools must share the same configuration. Correct outliers as required.

6.3 Reporting an Unknown Issue

If you experience an issue that is not documented, report the problem to SAP.

- If you are an external customer,
 1. Go to <https://support.sap.com/incident>.
 2. Provide the required information and create the incident.
- If you are an SAP internal customer,
 1. Go to the BCP portal.
 2. Select [ZCSSINTPROCE-Development / IMS](#).
 3. Navigate to  [Create > Internal Incident](#).

-
- 4. Select the CA-WDE component in the *Component* field and provide the required information. In the *Description* field, mention [MOB] followed by the description.
 - 5. Click **Save**.
- If you are a trial customer or want to understand more about the product,
 1. Go to <http://scn.sap.com/docs/DOC-60529>.
 2. Add your comments in the comments section.

Parent topic: [Troubleshooting \[page 97\]](#)

Previous task: [Resolving Usage or Configuration Errors \[page 109\]](#)

7 Previous Releases

The new features, enhancements, experimental features, resolved issues and other changes introduced in the previous release of Hybrid App Toolkit are listed according to the release version. Check the relevant release version to understand the changes.

7.1 Hybrid App Toolkit 1.27

Learn about the changes to Hybrid App Toolkit.

New Features

Disable the Privacy Screen to display app content in the app switcher

- Privacy Screen feature hides the app content when you switch between apps. The new check box in the *Configure Fiori Mobile Build Settings* wizard allows you to disable this feature.

Experimental Features

Experimental features are those that are currently still in development. They are available only in trial SAP Web IDE landscapes.

There are no new experimental features in this release.

Changed Requirements

Table 21:

Development Environment Dependency	Previous Requirement	Current Requirement
SAPUI5 library	1.44.7	1.44.12

Resolved Issues

For a list of open issues, see [Troubleshooting \[page 97\]](#).

 Note

To understand the changes introduced in the previous releases, see [Previous Releases \[page 112\]](#).

7.2 Hybrid App Toolkit 1.26

Learn about the changes to Hybrid App Toolkit.

New Features

- **Download build logs and turn on verbose loggings for each hybrid project**
Build logs can be downloaded both for failed and successful builds. When you configure hybrid apps for a Fiori build, you can opt to enable verbose logging. If you enable this option, after the build completes, you get more insight on what happened during the build.
See [Understanding the Verbose Logs \[page 36\]](#).
- **Minimum operating system version for a Fiori mobile application**
When you configure hybrid apps for a Fiori build, you can specify the Minimum OS Version for both Android and iOS. This helps you to build applications for the specified operating system versions.
See [Configuring Apps for Fiori Mobile Build \[page 32\]](#).

Experimental Features

Experimental features are those that are currently still in development. They are available only in trial SAP Web IDE landscapes.

There are no new experimental features in this release.

Changed Requirements

There are no changes in SAPUI5 library requirements or SDK tool version.

Resolved Issues

For a list of open issues, see [Troubleshooting \[page 97\]](#).

Note

To understand the changes introduced in the previous releases, see [Previous Releases \[page 112\]](#).

7.3 Hybrid App Toolkit 1.25

Learn about the changes to Hybrid App Toolkit.

New Features

- **Build Developer Companion for each hybrid project**

Hybrid App Toolkit now supports developers to build project-specific companion app in SAP Web IDE for each hybrid app. Every time you make some changes in the hybrid project, you just need to trigger a reloading in the Developer Companion on the mobile device/simulator/emulator to view the new runtime behaviour.

See [Building Developer Companion for Fiori Mobile Apps \[page 30\]](#).

- **Push Notifications for Fiori Mobile Apps**

You can send push notifications for your hybrid apps. Push notifications allows your apps to notify a user of new messages or events even when the user is not actively using the apps.

See [Push Notifications for Fiori Mobile Apps \[page 37\]](#).

Experimental Features

Experimental features are those that are currently still in development. They are available only in trial SAP Web IDE landscapes.

There are no new experimental features in this release.

Changed Requirements

Table 22: Required Versions

Development Environment Dependency	Previous Requirement	Current Requirement
SAPUI5 library	1.42.11	1.44.7
SDK Tool version		25.2.3 and below

Resolved Issues

For a list of open issues, see [Troubleshooting \[page 97\]](#).

i Note

To understand the changes introduced in the previous releases, see [Previous Releases \[page 112\]](#).

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.

Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

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: <http://help.sap.com/disclaimer>).

[go.sap.com/registration/
contact.html](http://go.sap.com/registration/contact.html)



© 2017 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 [http://www.sap.com/corporate-en/legal/copyright/
index.epx](http://www.sap.com/corporate-en/legal/copyright/index.epx) for additional trademark information and notices.