PUBLIC
SAP Service and Asset Manager
Document Version: 2205 – 2022-07-06

SAP Service and Asset Manager Installation Guide

THE BEST RUN SAP

Content

Document History

Before you begin reading this guide, be sure that you have the latest version. Find the latest version at https://help.sap.com/viewer/p/SAP_ASSET_MANAGER.

The following table provides an overview of the most important document changes.

Document Version	Date	Description of Changes
1.0	OCT 2021	Original release of the *SAP Asset Manager Installation* guide, version 2110

1 SAP Asset Manager Architecture

1.1 SAP Business Technology Platform Overview

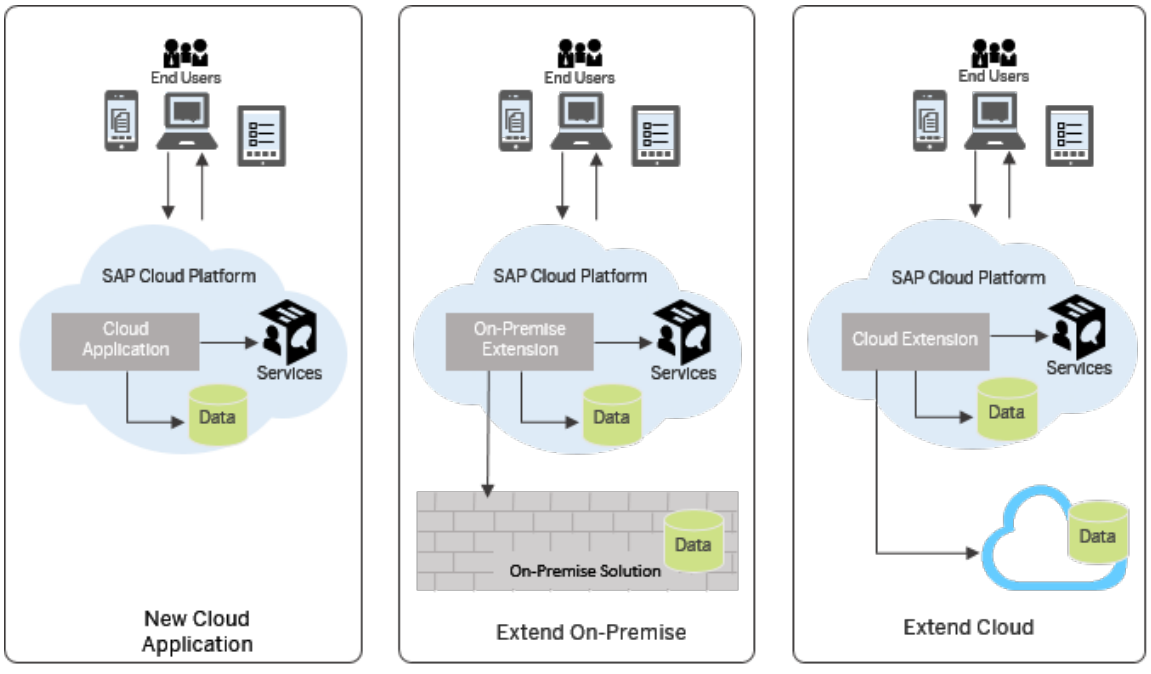
SAP Business Technology Platform enables customers and partners to rapidly build, deploy, and manage cloud-based enterprise applications that complement and extend your SAP or non-SAP solutions, either on-premise or on-demand.

> **i Note**
>
> For more information on prerequisites and procedures for setting up customer accounts on SAP Business Technology Platform, see the topic Getting Started with a Customer Account: Workflow in the Neo and Cloud Foundry environment.

SAP Business Technology Platform is an in-memory cloud platform based on open standards. It provides access to a feature-rich, easy-to-use development environment in the cloud. The platform includes a comprehensive set of services for integration, enterprise mobility, collaboration, and analytics.

As a Platform-as-a-Service operated by SAP, our product frees your administrators from any infrastructure and IT costs and offers state-of-the art quality of service.

Scenarios



- Develop new cloud applications
  This scenario is suitable for companies that need to start developing new applications from scratch. You can create brand new cloud applications and reach your end customers easily, with a low learning curve and small capital investment in software and hardware.
- Develop on-premise extensions
  This scenario is suitable for companies that have already invested a lot in on-premise IT infrastructure. You can create the new extensions to the system on the cloud, and integrate seamlessly with the on-premise components using Connectivity Service and Cloud Connector.
- Develop cloud extensions
  At SAP Business Technology Platform, you can also develop extensions to other cloud products, such as SuccessFactors.

Application development

You can use the following programming models to build highly scalable applications:

- Java - SAP Business Technology Platform is Java EE 6 Web Profile certified. You can develop Java applications just like for any application server. You can also easily run your existing Java applications on the platform.
- SAP HANA - you can use the SAP HANA development tools to create comprehensive analytical models and build applications with SAP HANA programmatic interfaces and integrated development environment.
- HTML5 - you can easily develop and run lightweight HTML5 applications in a cloud environment.
- SAPUI5 - use the UI Development Toolkit for HTML5 (SAPUI5) for developing rich user interfaces for modern Web business applications.

Solutions

In the context of SAP Business Technology Platform, a solution is comprised of various application types and configurations created with different technologies, and is designed to implement a certain scenario or task flow. You can deploy solutions by using the Change and Transport System (CTS+) tool, the console client, or by using the cockpit, where you can also monitor your solutions. To describe and technically realize the solutions, SAP Introduces the multi-target application (MTA) model. It encompasses and describes application modules, dependencies, and interfaces in an approach that facilitates validation, orchestration, maintenance, and automation of the application throughout its lifecycle.

Runtime container for applications

Applications developed on SAP Business Technology Platform run in a modular and lightweight runtime container. The platform provides a secure, scalable runtime environment with reusable platform services.

Virtual Machines

Virtual machines allow you to install and maintain your own applications in scenarios not covered by the platform. A virtual machine is the virtualized hardware resource (CPU, RAM, disk space, installed OS) that blends the line between Platform-as-a-Service and Infrastructure-as-a-Service.

Services

You can consume a set of services provided by SAP Business Technology Platform according to the technology you prefer and the use cases of your scenarios.

Integration with SAP and non-SAP software

SAP Business Technology Platform facilitates secure integration with on-premise systems running software from SAP and other vendors. Using the platform services, such as the connectivity service, applications can establish secure connections to on-premise solutions, enabling integration scenarios with your cloud based applications.

In-memory persistence

SAP Business Technology Platform includes persistence powered by SAP HANA, taking full advantage of its real-time, in-memory computing technology and built-in analytics.

Secure data

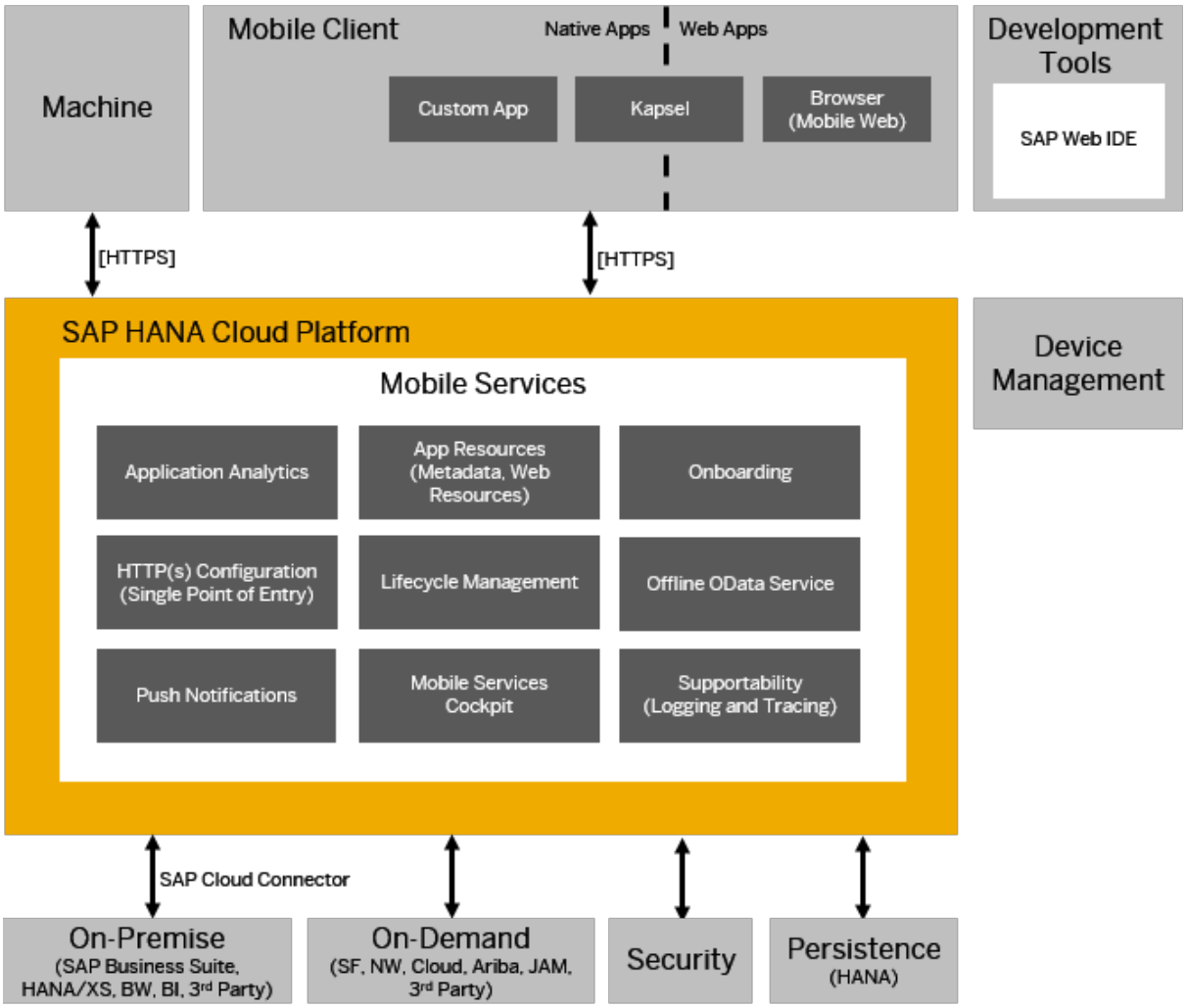
Comprehensive, multilevel security measures have been built into SAP Business Technology Platform. This security is engineered to protect your mission critical business data and assets and to provide the necessary industry standard compliance certifications.

Free trial

You can start by getting a free SAP Business Technology Platform developer license on SAP Business Technology Platform Developer Center that also gives you access to our community and all the free technical resources, tutorials, blogs, support you need.

1.1.1 SAP Business Technology Platform Mobile Services Overview

SAP Business Technology Platform Mobile Services, or SAP BTP services, provides services to mobile applications, such as application analytics, app resources, onboarding, and HTTP/HTTPS configuration.



Mobile application services consist of the following:

- **Application analytics:** Usage statistics that are displayed graphically in the SAP BTP cockpit
- **App resources:** Containers of dynamic configurations, styles, or content that are downloaded by native applications
- **Onboarding:** Authentication of users who are registering through SAP Mobile Place
- **HTTP/HTTPS configuration:** Open standards for client communications
- **Life cycle management:** Managing and deploying multiple versions of an application
- **Offline oData service:** Optimizes data transport between the back end and the client offline store
- **Push notifications:** Native notifications sent from back-end systems to the server, which forwards them on to the clients
- **SAP BTP cockpit:** Deploys, manages, and monitors applications

- **Supportability:** Logs for monitoring system health and troubleshooting

SAP BTP services can expose on-premise back end services through Cloud Connector, and on-demand back end services directly.

SAP BTP services security enables you to use an on-premise identity management system for on-demand applications. You can use basic authentication using LDAP, or form-based application authentication using SAML.

All configuration and runtime data is persisted in an SAP S/4HANA database.

1.1.2 SAP Business Technology Platform SDK Overview

The SAP Business Technology Platform SDK includes well defined layers (SDK frameworks, components, and platform services) that simplify development of enterprise-ready mobile native apps that take full advantage of the mobile platform features.

The SAP Business Technology Platform SDK is tightly integrated with the SAP Business Technology Platform Mobile Services Cockpit to provide the following:

- End-to-end integrated security
- Support for offline applications
- Enterprise grade logging and monitoring support
- Access to core SAP ERP or SAP S/4HANA data and business processes, as well as access to third-party data sources
- Access to SAP Business Technology Platform capabilities and services

For more information about the SAP Business Technology Platform SDK see the following topics, depending on your mobile client platform:

- SAP Cloud Platform SDK for iOS
- SAP Cloud Platform SDK for Android

2 SAP S/4HANA and SAP ERP Setup

2.1 SAP Mobile Add-On for ERP

Add the SAP Mobile Add-On for SAP ERP systems alongside the requisite service packs in order to provide the required OData services for specific versions of SAP Service and Asset Manager.

Depending on the version of SAP Service and Asset Manager, the following versions of the SAP Mobile Add-On are available for compatible SAP ECC 6.0 EHP7 SP14 systems and newer:

Add-On Component	SAM 2.0	SAM 3.0	SAM 4.0	SAM 1911
SMFND 630_740 SP01 SMERP 630_740 SP01 SMISU 630_740 SP01	Supported	Not supported	Not supported	Not supported
SMFND 630_740 SP02 SMERP 630_740 SP02 SMISU 630_740 SP02	Supported	Supported	Not supported	Not supported
SM3ND 630_740 SP02 SMERP 630_740 SP03 SMISU 630_740 SP03	Supported	Supported	Supported	Not supported

Ensure that the corresponding SAP Mobile Add-On and service packs are installed for the SAP Service and Asset Manager application you wish to run. For detailed information and instructions regarding the installation of the SAP Mobile Add-On for ECC 6.0 Systems, see the Mobile Add-On for ERP Installation Guide, or check primary notes 2577248 and 2660361. For detailed information and instructions regarding the configuration of SAP Mobile Add-On, see the SAP Service and Asset Manager Configuration Guide.

After downloading the files for the SAP Mobile Add-On desired support packages from the SAP Software Download Center, load the mobile add-on onto your system through the add-on manager, using the transaction code `SAINT`. Once the add-on is installed, load the support packages into your system through the Support Package Manager (accessed through transaction code `SPAM`).

Once the SAP Mobile Add-On and requisite support packages have been installed, follow the Post Installation - Required topic to fully configure the SAP Mobile Add-On for ERP. Ensure that the OData Service is assigned in `/IWFND/MAINT_SERVICE` and the B/C set is activated for the given version of SAP Service and Asset Manager.

After following the topic, use thefollowing checklist to ensure that the mobile application integration framework is properly installed and configured:

1. Ensure that the requisite Web Dynpro that controls the behavior of the SAP Mobile Add-On are properly activated.
    1. Transactions `/SYCLO/CONFIGPANEL` and `/SYCLO/ADMIN` open the Mobile Application Integration Framework Configuration Panel and Administration Panel for the desired back end SAP ERP system.
2. Ensure that the requisite B/C sets related to the desired version of the SAP Mobile Add-On are installed and activated.
    1. If these B/C sets have been properly activated, application configuration for the desired SAP Service and Asset Manager version appears in the transaction `/SYCLO/CONFIGPANEL` under *Mobile Application Parameters*.
3. Ensure that the OData service for the desired SAP Service and Asset Manager application is activated and assigned to the *Mobile Application OData Service Assignment*.
    1. The requisite OData service appears in the *Mobile Application OData Service Assignment* in the transaction `/SYCLO/CONFIGPANEL` and is assigned to the mobile application.
    2. The requisite OData service will also appear in the listing of OData services provided by the SAP Gateway system, found in transaction `/IWFND/MAINT_SERVICE`.
        1. Configure the alias assignment in the `/IWFND/MAINT_SERVICE` transaction. By selecting the desired OData service, the bottom-right panel informs administrators which back end connection alias is used for the connection to the backend SAP Mobile Add-On services.
        2. Perform a quick test of the OData service to ensure the proper OData service document is being returned by the service:
        After selecting the OData service, the bottom-right panel includes a link to an internal test using the gateway client. By using the internal gateway client tool with the HTTPS connection option, system administrators can ensure that their connections are properly reaching the correct back-end system from the SAP gateway and retrieving data for the proper data service providers for SAP Service and Asset Manager.
    3. Ensure that the idempotency jobs are set up from the SPRO configuration of the SAP gateway system, as SAP Service and Asset Manager relies on idempotency in HTTP OData services to ensure data integrity.
4. Ensure that the SAP back-end system is set up to allow authentication of HTTPS calls from the Cloud Connector via principal propagation.

When the SAP Mobile Add-On has been set up correctly, the OData service starts returning data in the SAP Gateway client, accessible from transaction `/IWFND/GW_CLIENT`.

2.2 Mobile Add-On for SAP S/4HANA

Add the SAP Mobile Add-On for SAP S/4HANA systems alongside the requisite service packs in order to provide the required OData services for specific versions of SAP Service and Asset Manager.

The following versions of Mobile Add-On for SAP S/4HANA are available for compatible SAP S/4HANA 1610 FPS01 systems and newer:

Add-On Component	SAM 1.0	SAM 1.1	SAM 2.0	SAM 3.0	SAM 4.0	SAM 1911
S4MFND 100 S4MERP 100	Supported	Not supported	Not supported	Not supported	Not supported	Not supported
S4MFND 100 SP01 S4MERP 100 SP01	Not supported	Supported	Not supported	Not supported	Not supported	Not supported
S4MFND 100 SP02 S4MERP 100 SP02 S4MISU 100	Not supported	Supported	Supported	Not supported	Not supported	Not supported
S4MFND 100 SP03 S4MERP 100 SP03 S4MISU 100 SP01	Not supported	Supported	Supported	Supported	Not supported	Not supported
S4MFND 100 SP04 S4MERP 100 SP04 S4MISU 100 SP02	Not supported	Supported	Supported	Supported	Supported	Not supported

> **i Note**
>
> It is required that you install SAP Mobile Add-On for 1909 and earlier versions of SAP S/4HANA system, but for later versions it is integrated and no add-on installation is needed.

Ensure that the corresponding SAP Mobile Add-On and service packs are installed for the SAP Service and Asset Manager application you wish to run. For detailed information and instructions regarding the installation of the SAP Mobile Add-On for 1610 FPS01 systems, see the Mobile Add-On for S/4HANA Installation Guide, or

check primary note 2493602 and 2977434. For detailed information and instructions regarding the configuration of SAP Mobile Add-On, see the SAP Service and Asset Manager Configuration Guide.

After downloading the files for the SAP Mobile Add-On desired support packages from the SAP Software Download Center, load the mobile add-on onto your system through the add-on manager, using the transaction code `SAINT`. Once the add-on is installed, load the support packages into your system through the Support Package Manager (accessed through transaction code `SPAM`).

Once the SAP Mobile Add-On and requisite support packages have been installed, follow the Post Installation - Required topic to fully configure the Mobile Add-On for SAP S/4HANA. Ensure that the OData Service is assigned in `/IWFND/MAINT_SERVICE` and the B/C set is activated for the given version of SAP Service and Asset Manager. After following the topic, use thefollowing checklist to ensure that the mobile application integration framework is properly installed and configured:

1. Ensure that the requisite Web Dynpro that controls the behavior of the SAP Mobile Add-On are properly activated.
    1. Transactions `/SYCLO/CONFIGPANEL` and `/SYCLO/ADMIN` open the Mobile Application Integration Framework Configuration Panel and Administration Panel for the desired back end SAP S/4HANA system.
2. Ensure that the requisite B/C sets related to the desired version of the SAP Mobile Add-On are installed and activated.
    1. If these B/C sets have been properly activated, application configuration for the desired SAP Service and Asset Manager version appears in the transaction `/SYCLO/CONFIGPANEL` under *Mobile Application Parameters*.
3. Ensure that the OData service for the desired SAP Service and Asset Manager application is activated and assigned to the *Mobile Application OData Service Assignment*.
    1. The requisite OData service appears in the *Mobile Application OData Service Assignment* in the transaction `/SYCLO/CONFIGPANEL` and is assigned to the mobile application.
    2. The requisite OData service will also appear in the listing of OData services provided by the SAP Gateway system, found in transaction `/IWFND/MAINT_SERVICE`.
        1. Configure the alias assignment in the `/IWFND/MAINT_SERVICE` transaction. By selecting the desired OData service, the bottom-right panel informs administrators which back end connection alias is used for the connection to the backend SAP Mobile Add-On services.
        2. Perform a quick test of the OData service to ensure the proper OData service document is being returned by the service:
           After selecting the OData service, the bottom-right panel includes a link to an internal test using the gateway client. By using the internal gateway client tool with the HTTPS connection option, system administrators can ensure that their connections are properly reaching the correct back-end system from the SAP gateway and retrieving data for the proper data service providers for SAP Service and Asset Manager.
    3. Ensure that the idempotency jobs are set up from the SPRO configuration of the SAP gateway system, as SAP Service and Asset Manager relies on idempotency in HTTP OData services to ensure data integrity.
4. Ensure that the SAP back-end system is set up to allow authentication of HTTPS calls from the Cloud Connector via principal propagation.

When the SAP Mobile Add-On has been set up correctly, the OData service starts returning data in the SAP Gateway client, accessible from transaction `/IWFND/GW_CLIENT`.

2.3 Setting Up Cloud Connector

Context

For detailed information on installing and configuring the SAP Cloud Connector, see the following links:

- SAP Development Tools for Cloud: SAP Development Tools for Cloud
- SAP Cloud Platform Connectivity – Cloud Connector: SAP Cloud Platform Connectivity – Cloud Connector

You must meet the following requirements when you set up SAP Cloud Connector for SAP Asset Manager:

- HTTP Configuration – SCPms communicates with SAP backend system using HTTP/ OData service
- Principal Propagation – Principal propagation to an ABAP system must be configured
- Certificate User Mapping in ABAP system – X.509 certificate must be mapped to named user in SAP ABAP system

Procedure

1. Install the SAP Cloud Connector software and follow the on-screen instructions. See Get Started with SAP Asset Manager - Cloud Connector for detailed information.
2. Perform an initial Cloud Connector configuration. See Initial Configuration - SAP Help Portal for detailed information.

   Mandatory field are as follows:

   - Region
   - Subaccount: S/4HANA account name
   - Subaccount user: S/4HANA Cloud Platform connection user
   - Password
3. Generate a CA certificate. See Installation of a System Certificate for Mutual Authentication for more information.
4. Set up the trust store with your backend system.
5. Set the System Certificate.

   Mandatory Field are as follows:

   - Common Name
   - Organizational Unit
   - Organization
   - Country
6. Set the Principle Propagation Subject Pattern.
7. Establish the connection between the cloud sub account and the cloud connector.
8. Establish Connection to the on-premise backend system.

9. Establish Cloud Connector Connection for the On-premise system.

   Cloud Connector is set up.

2.4 Setting Up Business Technology Platform

Context

In this procedure, you accomplish the following:

- Enable Cloud Foundry
- Configure entitlements
- Set up the Cloud Foundry space
- Create quote plans
- Assign a quote plan to a space

For detailed information, see the Set Up Customer Accounts topic in the *SAP Cloud Platform Mobile Services (Cloud Foundry)* guide. For configuration information, see

Procedure

1. **Enable Cloud Foundry**
   a. Log on to the SAP Business Technology Platform Cockpit, using the administrator global account for the Cloud Foundry environment.
   b. Select the Cloud Foundry subaccount tile.
   c. Click the *Enable Cloud Foundry* button.

      The *Create Cloud Foundry Organization* dialog box appears.
   d. Enter the name for your Cloud Foundry organization and select *Create*.

   Cloud Foundry is enabled.
2. **Configure entitlements**
   a. Select *Entitlements* from the left menu pane.
   b. Click *Configure Entitlements*, then click *Add Service Plans*.
   c. Add the following service plans:
      - Application Runtime
      - Mobile Services
3. **Set up the Cloud Foundry space**
   a. In the left menu panel, select *Spaces*. If a Cloud Foundry space doesn't yet exist in the Cloud Foundry organization, click the *Create Space* button and create a space.

b. Enter the name for your Cloud Foundry space in the *Create Space* dialog box.
c. Ensure the *Space Manager* checkbox is checked. Click *Create*.

4. **Create quota plans**
   a. In the left menu panel, select *Quota Plans*.
   b. Click *New Plan*.
   c. Create a space quota plan, entering your information in the following fields:
      - Name
      - Memory (MB)
      - Routes
      - Services
      - Instance Memory (MB)
      - App Instance
   d. Click *Save*.
5. **Assign quota plan to the space:** In the *Plan Assignment* section, select your quota plan that you created in the previous step and substeps.

2.5 Setting up the SAP Business Technology Platform Cloud Connector System Mapping

The SAP Business Technology Platform Cloud Connector serves as the link between on-demand applications in the SAP Business Technology Platform and existing on-premise systems.

For general Cloud Connector set up and configuration, see the SAP Business Technology Platform Cloud Connector documentation, specifically Configure Access Control (HTTP). The following topic addresses specific SAP Service and Asset Manager Cloud Connector configurations to make in the Cloud Connector cockpit. Perform these additional configuration modifications after the initial Cloud Connector installation and configuration.

1. Click the *Cloud to On-Premise* link, and in the *Access Control* tab, add your on-premise system in your Cloud to On-Premise mapping as follows:
   - **Protocol:** HTTPS
   - **Back-End Type:** ABAP
   - **Virtual Host:** Your choice
   - **Virtual Port** Your choice
   - **Internal Host:** SAP back-end system
   - **Internal Port:** SAP back-end system
   - **Principle Type:** X.509 Certificate
2. Add a resource to your Cloud to On-Premise account:
   - **URL Path:** `/sap/opu/odata`
   - Click the *Path and all sub-paths* radio button
3. Click the *Principle Propagation* tab and set each of the *Trust Configurations* to *Trusted* (checked).

2.6 Creating the SAP Asset Manager Mobile Application in SCPms

You can create a mobile app instance in Business Technology Platform/ Mobile Services using the app creator program.

Context

On how to create the SAP Service and Asset Manager Mobile Application in SCPms, see Step 9 inPost Installation- Required.

2.7 SAP Business Technology Platform Security Setup

SAP Service and Asset Manager requires the SAP Business Technology Platform Mobile Service to provide user onboarding, user authentication, mobile application lifecycle management, and OData offline support.

> **i** Note
>
> Enable OAuth 2.0 based user authentication for SAP Business Technology Platform as required by SAP Service and Asset Manager.

For more information about SAP Business Technology Platform security, see the Web site, SAP Cloud Platform Security: Trust Matters.

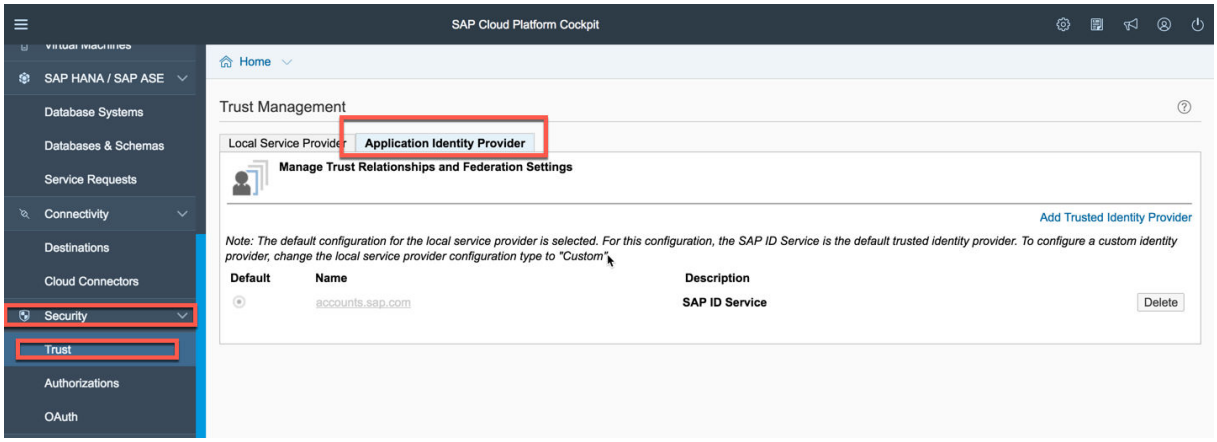
SAP Service and Asset Manager requires user principle propagation when setting up a mobile destination in the SAP Business Technology Platform Mobile Service. User principle propagation is necessary to properly perform data distribution calculation and authorization checks in the SAP back-end system.

For more information about the SAP Business Technology Platform Mobile Service, see the SAP Cloud Platform Mobile Service for Development and Operations guide.

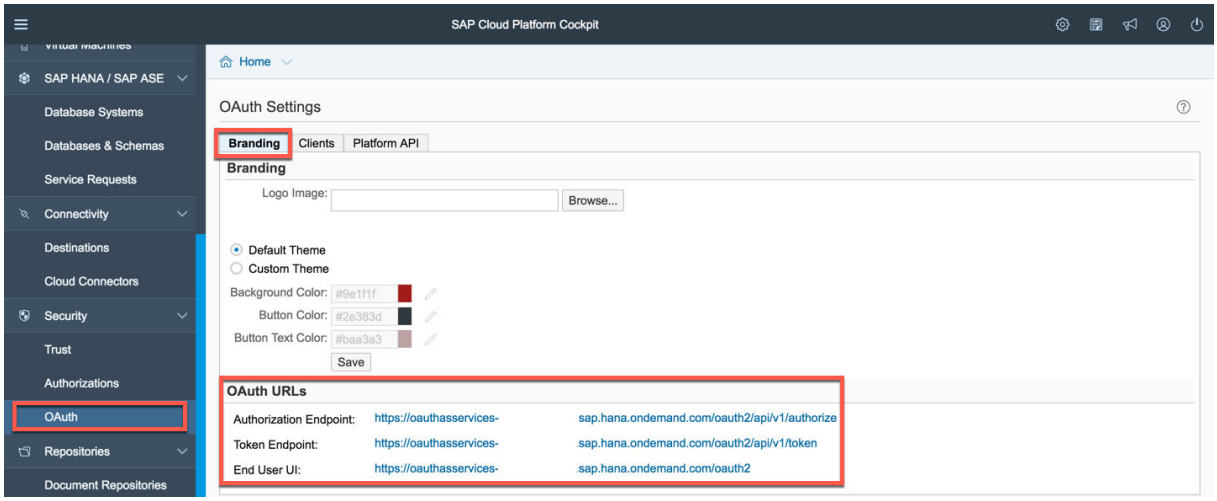
SAP Cloud Platform OAuth Authentication Service

SAP Business Technology Platform offers an OAuth 2.0 user authentication service that communicates with an Identity Provider or local trust store to provide a secure method of passing valid credentials through HTTP calls.

Using the SAP Business Technology Platform Cockpit, set Identity Providers in the *Trust* section of the *Security* page, under the *Application Identity Providers* tab:



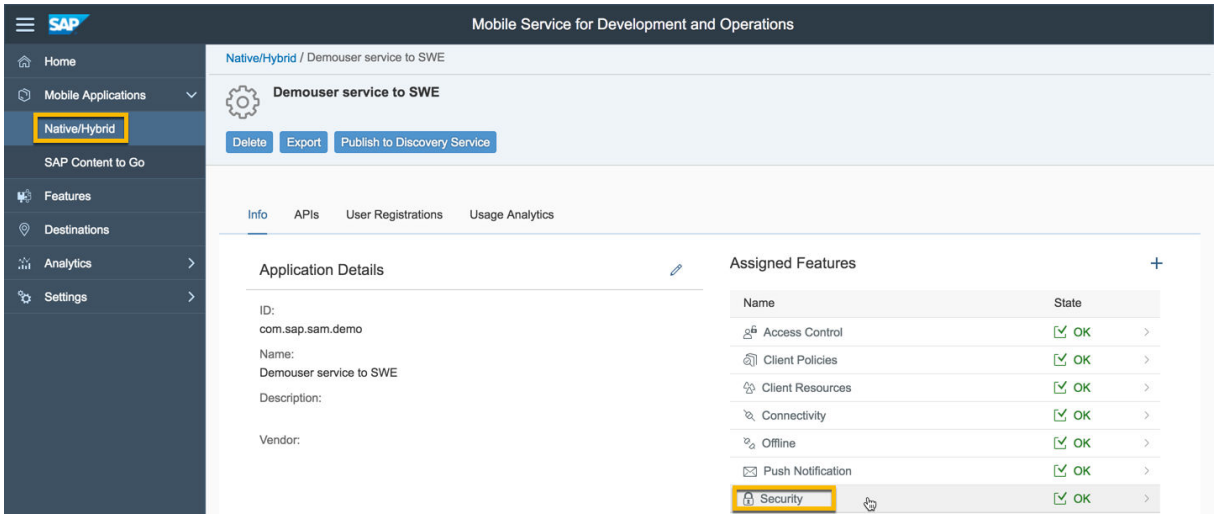
Once you set the Identity Provider, configure the OAuth service to generate tokens for requests accepted by the Identity Provider. Configure the requests using standard OAuth 2.0 token retrieval methods through the URLs on the bottom of the *Branding* tab:



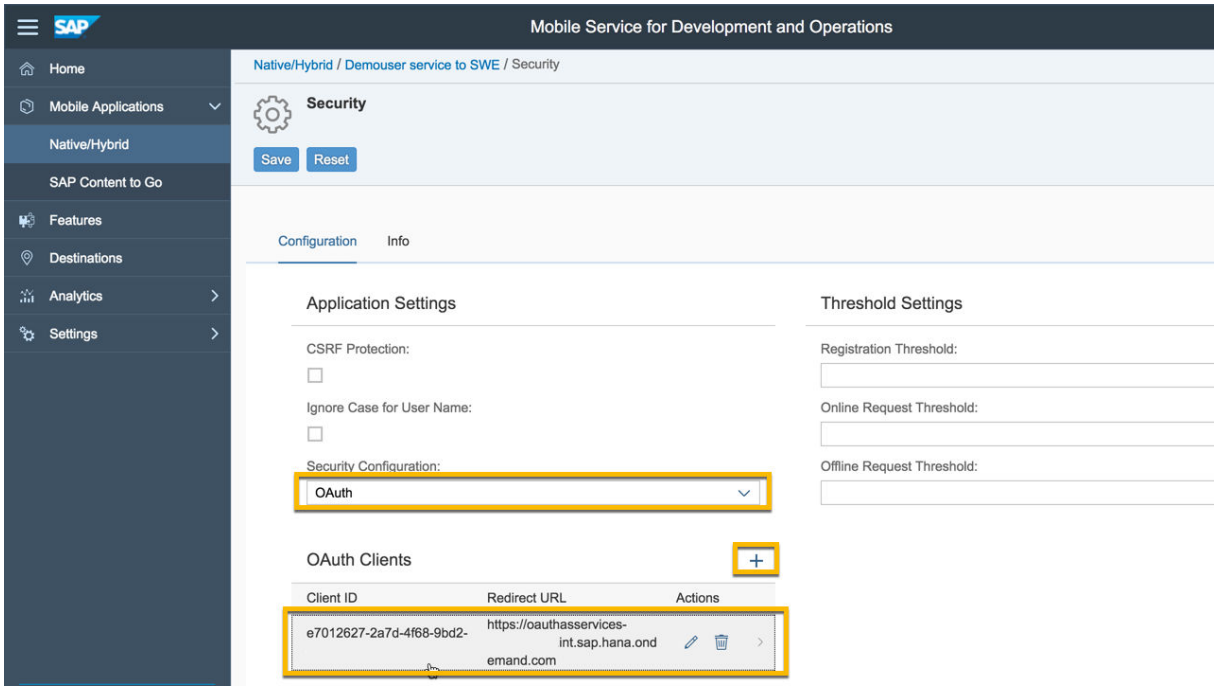
Adding an OAuth Authentication Client to a Mobile Services Application

When you create a mobile services application to provide OData services to client applications, you can secure the application by using the SAP Business Technology Platform as the authorization server through the oAuth service it provides. To secure the application, enable the *Security* feature of the mobile service application. Then, configure it to use the oAuth service of the SAP Business Technology Platform that was previously set up. Finally, add the client to the security settings of the application.

Adding a client to the OAuth platform service of the SAP Business Technology Platform is performed as a *Security* feature in the *Mobile Application* configuration on the Mobile Services Cockpit:



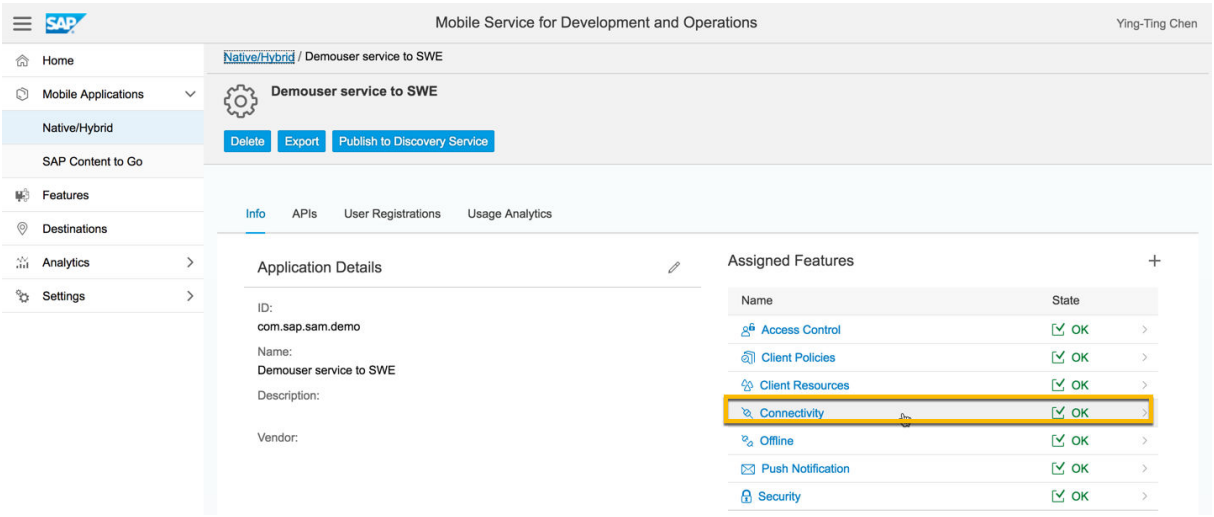
Inside of the security features of the application, set the *Security Configuration* to `<OAuth>` to use OAuth 2.0 tokens to authenticate to the application. Setting this field ensures that users can repeatedly authenticate into the application for the life of the OAuth token without having to contact the Identity Provider to enter their credentials again. The OAuth service tracks the identity of the token and uses the token to authenticate users into the application, as the token is passed in with the connection attempts. You can add individual clients to include individual redirect URLs or token expiration dates if necessary.



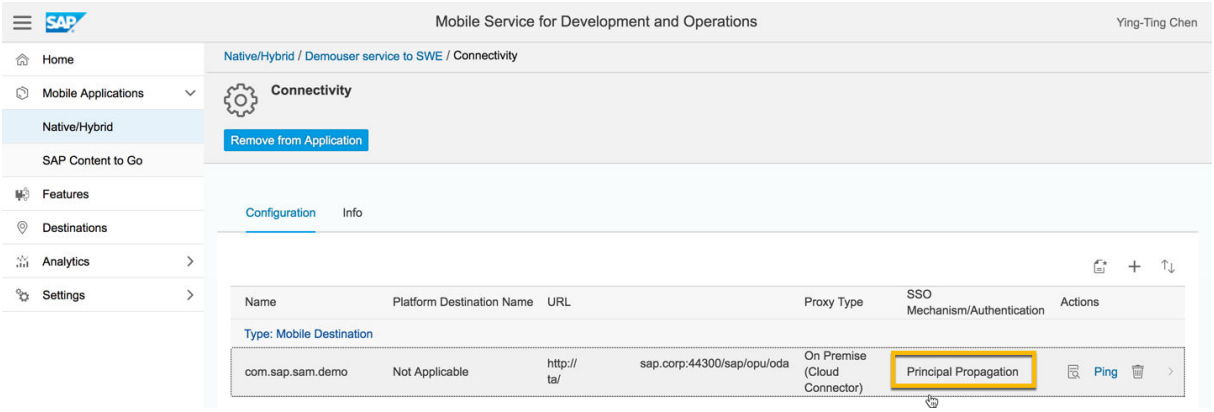
Configuration Principle Propagation to Backend Connections

Authentication to an individual component served by Mobile Services does not ensure authentication to a backend service. To propagate the authentication information of the application to the backend service, turn on principle propagation of the destination that is set as the backend connection of the application.

Check the setting in the *Connectivity* tab of the *Mobile Application*.

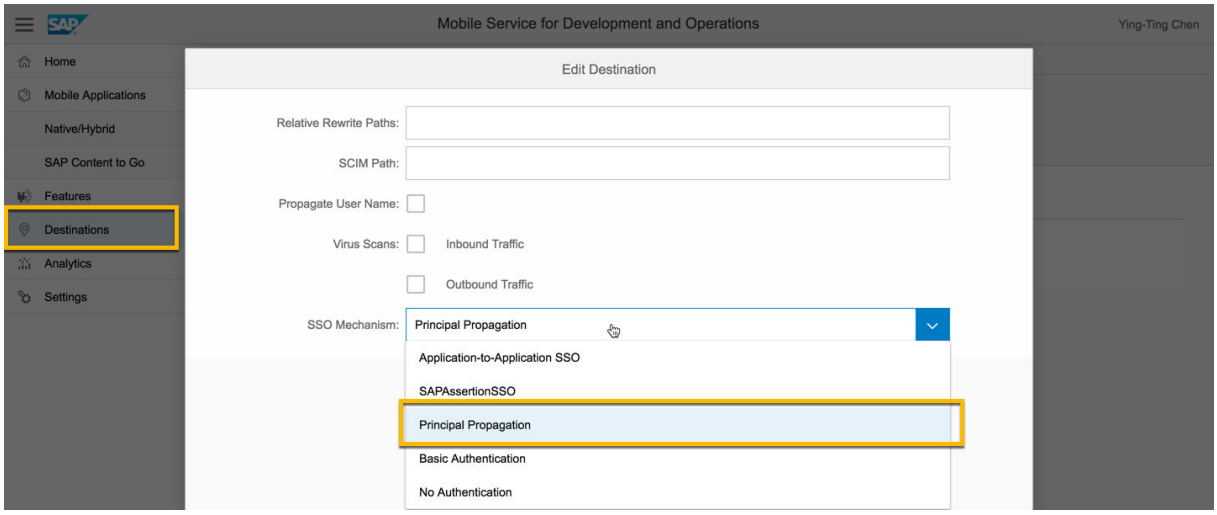


Configure the security method of the connection for *Principle Propagation*. If it is not configured for principle propagation, it will not provide the credentials from the application to the on-premise back-end system that is providing the data service for the SAP Mobile Add-On.



To edit the connection to the backend, edit it in the *Destinations* tab of the Mobiles Services Cockpit.

To edit the connection, select the connection, click *Edit*, and navigate through the wizard to the *SSO Mechanism* page. Change the *SSO Mechanism* to `<Principal Propagation>`.



Note

If user authentication services are not required, use the *Basic Authentication* setting to allow connections to the back-end system to log in as a single, predefined, set of credentials for every user that accesses the connection.

3 Deployment and Configuration of SAP Asset Manager

3.1 High-Level Installation Process for SAP Service and Asset Manager

To install the SAP Service and Asset Manager application, you must follow a specific order for certain steps in the installation process. Some objects in the installation rely on previously installed objects for IDs or URLs, or other data.

Use the following high-level process table for information on the process steps to take when installing the SAP Service and Asset Manager.

Step	Product	Role	Prerequisite Steps
Prerequisites	Before you can install the SAP Service and Asset Manager application, ensure that the following SAP components are installed: • Plant Maintenance is installed and running • HR: If you're using distribution work orders by personnel number, then the HR personnel management module is required.		
1	Creation or update of SAP Business Technology Platform account Use the default subaccount or create a subaccount within your SAP Business Technology Platform account. See the following: • Getting Started topic in the *SAP Business Technology Platform* guide • Neo installations: SCPms - Neo topic in the *SAP Asset Manager Getting Started* guide • Cloud Foundry installations: SCPms - Cloud Foundry topic in the *SAP Asset Manager Getting Started* guide	SAP Business Technology Platform administrator	Must be first step
2	**Mobile Add-On for SAP S/4HANA system installation procedure:** See Installing the Mobile Add-On for SAP S/4HANA **Mobile Add-On for SAP ERP system installation procedure (ECC):** See Installing the Mobile Add-On for ERP	SAP S/4HANA administrator or SAP ERP (ECC) administrator	None

Step	Product	Role	Prerequisite Steps
3	On-premise Cloud Connector setup. See the Cloud Connector Installation topic.	IT administrator	Relies on Step 1
4	Create your application in SAP BTP services (SAP Cloud Platform Mobile Services): Set application security settings. For more information, see the Security Administration topic in the SAP BTP services guide. Set application connection settings. For more information, see the topic Set Up Customer Accounts and the associated subtopics in the SAP BTP services guide.	SAP Business Technology Platform administrator or Business expert	Application connection settings rely on Step 2
5	Choose to set up or build the mobile application in one of the following ways: 1. Set up the mobile application: ○ Obtain onboarding URL and QR code. See the Onboarding to the MDK Client App topic for more information. 2. Build the mobile application (optional): ○ Building the SAP Service and Asset Manager Application Overview [page 64] ○ Building the SAP Service and Asset Manager Application Client [page 65] No matter which option you select, see also *Step 6* in the topic Building the SAP Asset Manager Client to link the onboarding URL with the SAP Service and Asset Manager client application.	Administrator	Relies on Application security settings in Step 4
6	Import the metadata definitions into the SAP Web IDE For more information, see Enabling the Mobile Development Kit [page 39]	Business expert	Relies on Steps 4–5
7	Launch the SAP Service and Asset Manager application using the onboarding URL. Navigate through the launch screens and perform the initial sync.	Administrator	Relies on Steps 1–6

3.2 Relevant SAP Notes

Read the following SAP Notes before you begin installation. The SAP Notes contain the most recent information about the installation, as well as any corrections to the installation process.

Make sure you have the most recent version of each SAP Note. Find SAP Notes on the SAP Service Marketplace at https://sap.com/notes.

SAP Note Number	Title	SAP System Version	Description
2495578	Release Information Note - Mobile Add-On for SAP S/4HANA 1.0 and Support Packages	SAP S/4HANA	Information and references to additional SAP Notes in the context of applying the Mobile Add-On for S/4HANA 1.0 and support packages to an SAP S/4HANA on premise system
2493602	SAP Asset Manager Mobile Add-On for SAP S/4HANA Installation Primary SAP Note	SAP S/4HANA	Release and Information (RIN) note about planning the installation and upgrades of the ABAP Add-On for the SAP Asset Manager application
2566071	SAP Cloud Platform (SCPms) Setup Info for SAP Asset Manager with an SAP S/4HANA On-Premise System	SAP S/4HANA	Information on how to set up SAP Business Technology Platform Mobile Services (SCPms) to run the SAP Service and Asset Manager application
2660862	SAP Cloud Platform (SCPms) Setup Info for SAP Asset Manager with an SAP ERP On-Premise System	SAP ERP	Information on how to set up SAP Business Technology Platform Mobile Services (SCPms) to run the SAP Service and Asset Manager application
2933065	SAP Asset Manager - MDK Support Matrix	Mobile Development Kit	Detailed information on SAP Service and Asset Manager and Mobile Development Kit version compatibility
2970982	MDK App Crash on Launch on Certain iOS 14 Devices	Mobile Development Kit	Corrects issues with iOS 14 when building an iOS client

You can find additional information for the Mobile Development Kit in the following continually updated blog post: Latest Mobile Development Kit Blogs and Videos.

3.3 Supported Languages

The SAP Service and Asset Manager application supports the following languages:

- ar001 - Arabic
- zh_hans - Simplified Chinese
- zh_hant - Traditional Chinese
- csCZ - Czech Republic
- daDK - Danish
- nlBE - Dutch
- elGR - Greek
- enUS - English
- frFR - French

- deDE - German
- heIL - Hebrew
- hrHR - Croatian
- huHU - Hungarian
- itIT - Italian
- jaJP - Japanese
- koKR - Korean
- nbNO - Norwegian
- plPL - Polish
- ptBR - Portugese
- roRO - Romanian
- RuRU - Russian
- srSP - Serbian
- skSK - Slovak
- slSL - Slovenian
- esES - Spanish
- svSE - Swedish
- thTH - Thai
- trTR - Turkish

3.4 Installation Prerequisites

SAP Service and Asset Manager uses the oData service that is provided with the SAP Business Technology Platform Mobile Services.

The oData service provides a universally available service for the SAP Service and Asset Manager application.

> **i** Note
>
> To fully install and run the SAP Service and Asset Manager application, perform the Building the SAP Service and Asset Manager Application Client [page 65] procedure after performing the tasks in this Deployment and Configuration section.

Back End Landscape Prerequisites

- Ensure that an SAP Business Technology Platform Mobile Services account with an active mobile services account is available
- A Cloud Connector is required. For more information, see the Setting up the SAP Business Technology Platform Cloud Connector System Mapping [page 16] procedure.
- The correct SAP Mobile Add-On for your system is already installed. For more information, see the High-Level Installation Process for SAP Service and Asset Manager [page 22] topic, *Step 2*.

Prerequisites for SAP Service and Asset Manager Application Installation

- Access to your SAP Business Technology Platform Mobile Services environment
- Administrative access to the Mobile Development Kit
- Access to a computer to build and run the Mobile Development Kit client either in a simulator or on a device
- If using certificate-based authentication, acquire the certificates the IdP expects from mobile devices running the SAP Service and Asset Manager application. See the SAP Cloud Platform Identity Provider portal page for complete information on working with certificates and Identity Providers.
  If you optionally enable certificate based authentication, the Mobile Development Kit client passes the certificate to the IdP for use in authentication as long as the IdP supports and asks for certificates, and certificates are present on the mobile device.

Software Prerequisites

For all prerequisite and installation information for the Mobile Development Kit, see the https://help.sap.com/viewer/977416d43cd74bdc958289038749100e/Latest/en-US manual.

3.5 Creating an Application in SAP Business Technology Platform Mobile Services - Overview

To successfully deploy the SAP Service and Asset ManagerSAP Shop Floor Manager application, you must create a corresponding application in SAP Business Technology Platform Mobile Services.

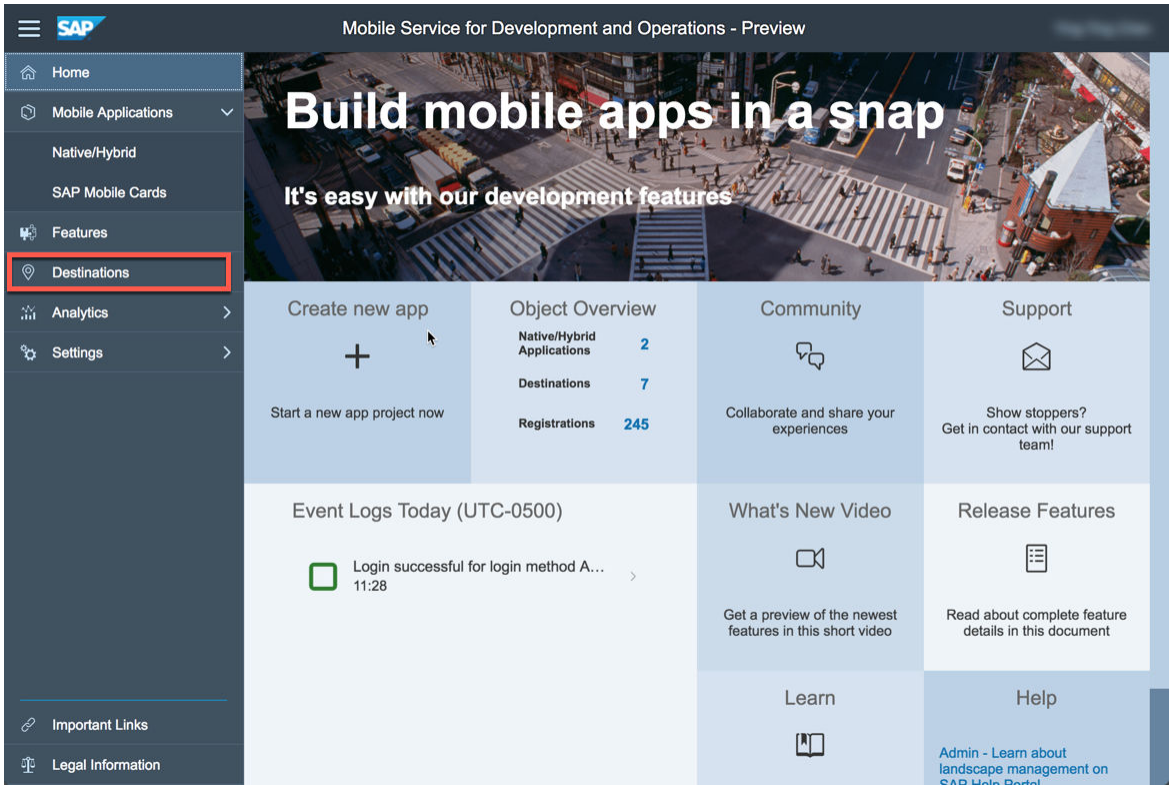
The corresponding application created in SAP Business Technology Platform Mobile Services connects a mobile device application to its corresponding data service on the on-premise back-end servers. There are two corresponding versions of SAP Business Technology Platform Mobile Services: an SAP Business Technology Platform Neo environment and an SAP Business Technology Platform Cloud Foundry environment.

3.5.1 Creating an Application in SAP Business Technology Platform Mobile Services - Neo

Procedure

1. Using theSAP Business Technology Platform Cockpit, navigate to the *Services* > *Development & Operations* tile. If it is not enabled, enable it. Then click the *Go to Service* link after clicking on the tile.

   The Mobile Service for Development and Operations, or Admin UI view, opens in a new window.



2. Click the *Destinations* link on the left, then click *New* to create a new destination. Fill in the following fields:
   - **Type:** Mobile destination
   - **Destination names:** Must match the names of the destinations defined in the *Application Metadata*.
     - For the SAP Service and Asset Manager application, the default destination name defined in the product release varies based on which version of the application is running. See the following chart for more information:

SAP Service and Asset Manager Version	Destination
SAP Service and Asset Manager 1.1	DEST_SAM10_PPROP
SAP Service and Asset Manager 2.0	DEST_SAM20_PPROP
SAP Service and Asset Manager 3.0	DEST_SAM30_PPROP
SAP Service and Asset Manager 4.0	DEST_SAM40_PPROP
SAP Service and Asset Manager 1911	DEST_SAM1911_PPROP
SAP Service and Asset Manager 2005	DEST_SAM2005_PPROP
SAP Service and Asset Manager 2010	DEST_SAM2010_PPROP
SAP Service and Asset Manager 2105	DEST_SAM2105_PPROP
SAP Service and Asset Manager 2110	DEST_SAM2110_PPROP

SAP Service and Asset Manager Version	Destination
SAP Service and Asset Manager 2.0 with Meter Management	DEST_SMM10_PPROP
SAP Service and Asset Manager 4.0 online service	DEST_SAM40_ONLINE_PPROP
SAP Service and Asset Manager 1911 online service	DEST_SAM1911_ONLINE_PPROP
SAP Service and Asset Manager 2005 online service	DEST_SAM2005_ONLINE_PPROP
SAP Service and Asset Manager 2010 online service	DEST_SAM2010_ONLINE_PPROP
SAP Service and Asset Manager 2105 online service	DEST_SAM2105_ONLINE_PPROP
SAP Service and Asset Manager 2110 online service	DEST_SAM2110_ONLINE_PPROP

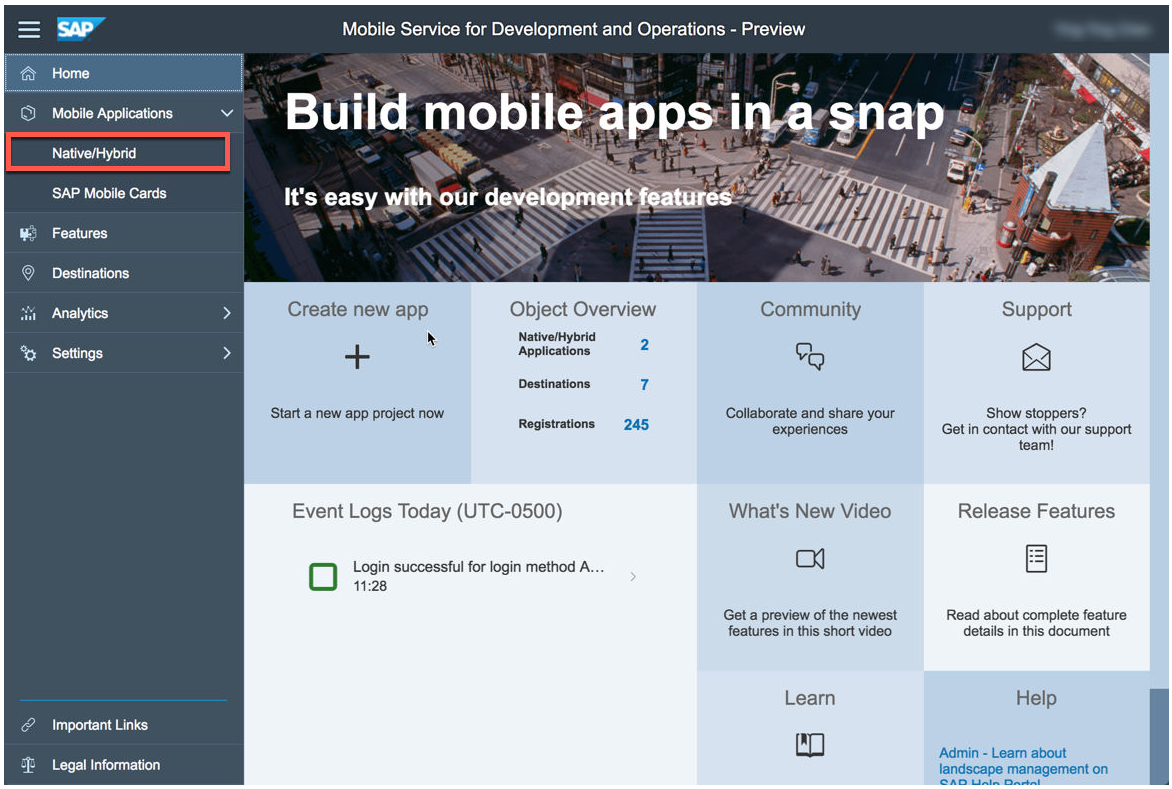
    - If the application metadata is edited to use a different destination name than what is defined in the `assetmanager.service` file, the created destination on the SAP Business Technology Platform Mobile Services must also reflect the service name. For more information on where to find application metadata, see *Step 1* of the Building the SAP Asset Manager Application Client procedure.

- **URL:** URL to reach the mobile add-on service through the Cloud Connector
    - The URL specified must use the given host name from the virtual host of the Cloud Connector regardless of port specification.
    - Include the full URL path of the OData service in the URL. All of the resources are specified in the ICF nodes that the OData service resides under.
    - If the Cloud Connector connecting to the Mobile Development Kit client is configured to use HTTPS, you do not have to make the HTTPS specification in the URL. Your on-premise connections through a properly configured Cloud Connector should specify a URL using an HTTP scheme, with the HTTP port specified.
    - A sample URL is as follows: `http://<cloud connector virtual host>:<cloud connector virtual port> /sap/opu/odata/<mobile add-on component>/<mobile add-on service>`, with `MERP` an example of `<mobile add-on component>` and `SAP_ASSET_MANAGER_<version>` an example of `<mobile add-on service>`.
- **Proxy Type:** On-premise (Cloud Connector)
- **Maximum Connections:** However many concurrent users are expected for your application connection
- **Timeout:** The timeout for each network transmission between the device and back end system. Settings are based on the following:
    - The expected timeout requirements for the SAP Mobile Add-On for SAP S/4HANA on-premise connections to SAP Service and Asset Manager is 600,000ms.
    - The expected timeout requirements for the SAP Mobile Add-On for SAP ERP ECC6.0 EHP 7 SP14 connections to SAP Service and Asset Manager is 1,200,000ms.
- **Rewrite Mode:** Rewrite URL. Setting must match the mapping in the Cloud Connector
    - If the virtual host and the port in the Cloud Connector are set the same as the internal host and the port of the SAP Mobile Add-On OData service, you can set the Rewrite Mode to *Rewrite URL*.

- If the virtual host and the port in the Cloud Connector do not match the internal host and the port of the SAP Mobile Add-On OData service, set the Rewrite Mode to *Custom Rewrite URL*. Setting the Rewrite Mode to Custom Rewrite URL properly maps the outgoing URLs from the Cloud Connector.
- **OPTIONAL** If the *Custom Rewrite URL* is set, add an outbound rewrite URL when you create system mapping. For more information, see the Rewrite Modes topic.

- **Custom Headers:** Not applicable
  - If a default client is not properly set for your oData service, you can override the connected client by adding a custom header with a `<Header Key>` of *sap-client* and a `<Header Value>` of your client number.
- **SSO Mechanism / Authentication:** Principal Propagation
  - Principal propagation is the officially supported authentication method between the SAP Service and Asset Manager and the SAP Mobile Add-On. For more information, see the topic Configure Principal Propagation to an ABAP System for HTTPS.

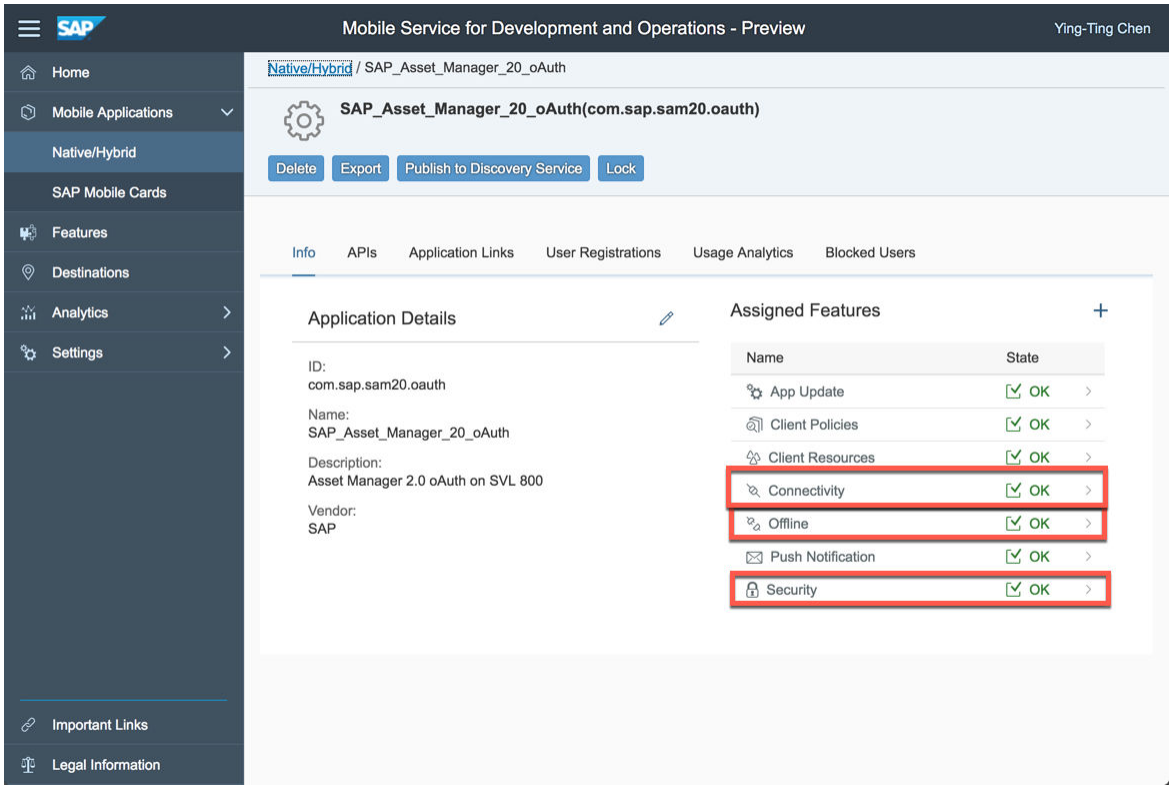
3. Click the *Mobile Applications* link on the left, then click *Native / Hybrid*.



4. Click the *New* button. When the *New Application* window displays, fill the fields as follows:
   - **Config Templates:** Select *Mobile Development Kit* from the dropdown
   - **ID:** Whatever your administrator wishes to use that is meaningful to them. The ID is used to control features on the application and is displayed in branded settings on the client.
   - **Name:** Name of your application
   - **Description:** Optional description of your application
   - **Vendor:** Optional name of your vendor
5. Click *Save* to create the preliminary cloud application, then click the newly created cloud application.

You access the detail screen of your newly created application.

6. Some *Assigned Features* need additional configuration, or you can optionally configure them at a later date:
   - **App Update:** You can configure this later. This setting controls how new metadata definitions are pushed to the Mobile Development Kit clients.
   - **Client Policies:** Enabled by default. Can control uploading of client error logs to the SAP Business Technology Platform Mobile Services, whether or not passwords are allowed to secure applications, and locking and wiping policies.
     If users need to lock their devices with passwords,select *Enable Passcode Policy*. You can define the password requirements after selection.
   - **Connectivity:** Additional configuration needed (performed in the *Step 8*). How the SAP Business Technology Platform communicates with the back end.
   - **Offline:** See Updating Offline Settings for SAP Service and Asset Manager Overview [page 44]
   - **Push Notification:** See the Activating Default Push Services for SAP Service and Asset Manager - Neo procedure
   - **Security:** Additional configuration needed (performed in the current procedure). How authentication is performed.



7. **Optional:** To debug an existing application, click the *Client Policies* link and ensure the following logs are enabled:

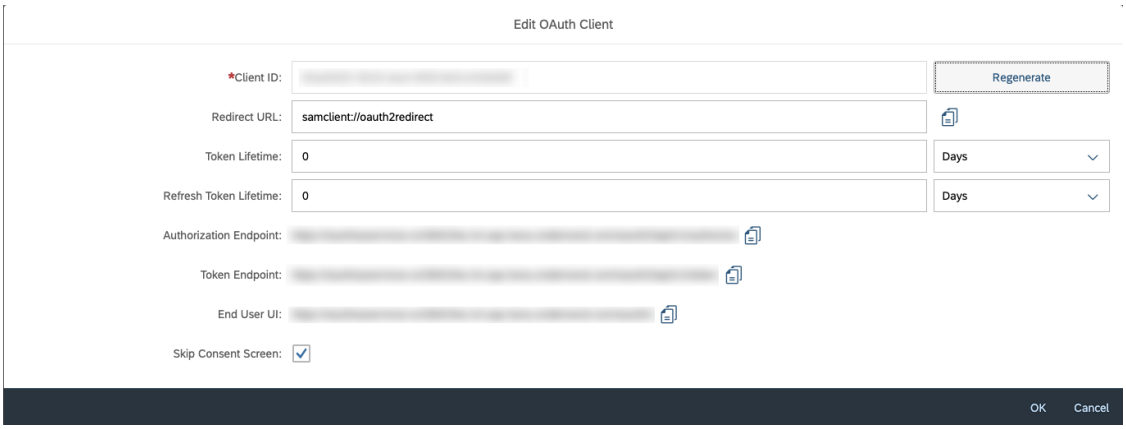
   Set the log levels for the lowest level you wish to see logs. For example:
   - **ERROR:** Displays only errors
   - **PATH:** Displays everything that can show up in the logger

8. Click the *Connectivity* link.

   The Connectivity window displays.

9. Select the *Add Destination* button (+). Then select the appropriate destination that you defined in *Step 2* of this procedure.
10. *Save* your changes. Click back to your new application and click the *Security* link.

    The Security window displays.
11. In the *Security Configuration* dropdown field, select *OAuth*. *Save* your changes, click back into the main application window to refresh the configuration. Then click the *Security* link again to enter the Security window.

    The Security window displays with new *OAuth Client* section visible.
12. Click an OAuth client to view or change the following settings:
    - **Token Lifetime:** Default is set to *0*, which is infinite (never expires)
    - **Refresh Token Lifetime:** Default is set to *0*, which is infinite (never expires)
    - **Redirect URL:** If enabling certificate-based authentication, ensure the field contains the same *samclient* URL scheme that is contained in the *Onboarding URL* scheme. The redirect URL must also match the redirect URL defined for the SAP Service and Asset Manager application.
      See the following screenshot for an example.
    - 

    Make a note of the following settings, as you need them during the Building the SAP Service and Asset Manager Application Client [page 65] procedure:
    - **Client ID**
    - **Authorization Endpoint**
    - **Token Endpoint**
13. Click back to the main application page and click the *APIs* tab. Make a note of the *Server* setting, minus the trailing /, as you need it during the Building the SAP Service and Asset Manager Application Client [page 65] procedure.

3.5.2 Creating an Application in SAP Business Technology Platform Mobile Services - Cloud Foundry

Procedure

1. Using the SAP Business Technology Platform Cockpit, navigate to the space you want to work in. If mobile services are already enabled, navigate to a *Service Instance* and open it.

   > **i** Note
   >
   > If mobile services are not enabled, enable it in the *Service Marketplace*.

2. Click the *Service* link. Follow the navigation, and click *Support* to open mobile services on Cloud Foundry.

   The Mobile Service for Development and Operations, or Admin UI view, opens in a new window.
3. Click the *Mobile Application* link on the left, then click *Native/Hybrid* to create a new application.
4. Click the *New* button. When the *New Application* window displays, fill the fields as follows:
   - **ID:** Whatever your administrator wishes to use that is meaningful to them. The ID is used to control features on the application and is displayed in branded settings on the client.
   - **Name:** Name of your application
   - **Description:** Optional description of your application
   - **Vendor:** Optional name of your vendor
   - **Back-End Connection Timeout:** The timeout for each network transition between the device and the back-end system. Settings are based on the following:
     - The expected timeout requirements for the SAP Mobile Add-On for SAP S/4HANA on-premise connections to SAP Service and Asset Manager is 600,000 ms.
     - The expected timeout requirements for the SAP Mobile Add-On for SAP ERP ECC6.0 EHP 7 SP14 connections to SAP Service and Asset Manager is 1,200,000ms.
5. Click *Save* to create the preliminary cloud application, then click the newly created cloud application.

   You access the detail screen of your newly created application.
6. Some *Assigned Features* need additional configuration, or you can optionally configure them at a later date:
   - **Mobile App Update:** You can upload application metadata bundles to update mobile apps on devices here. This setting controls how new metadata definitions are pushed to the Mobile Development Kit clients.
   - **Mobile Settings Exchange:** Can control uploading of client error logs to the SAP Business Technology Platform Mobile Services, whether or not passwords are allowed to secure applications, and locking and wiping policies.
     If users need to lock their devices with passwords, select *Enable Password Policy*. Then define the password requirements.
   - **Mobile Connectivity:** How the SAP Business Technology Platform communicates with the back end. Additional configuration needed (performed in *Step 7*).
   - **Mobile Offline Access:** See Updating Offline Settings for SAP Service and Asset Manager Overview [page 44].

- **Mobile Push Notification:** See the Activating Default Push Services for SAP Service and Asset Manager - Neo procedure.
- **Mobile Network Trace:** Sets up network tracing for the offline and connectivity components.

7. Click the *Mobile Connectivity* feature. Then click *New* under *Mobile Destinations* to create a new destination. Fill in the following fields:
   - **Type:** Mobile destination
   - **Destination names:** Must match the names of the destinations defined in the *Application Metadata*.
     - For the SAP Service and Asset Manager application, the default destination name defined in the product release varies based on which version of the application is running. See the following chart for more information:

SAP Service and Asset Manager Version	Destination
SAP Service and Asset Manager 1.1	DEST_SAM10_PPROP
SAP Service and Asset Manager 2.0	DEST_SAM20_PPROP
SAP Service and Asset Manager 3.0	DEST_SAM30_PPROP
SAP Service and Asset Manager 4.0	DEST_SAM40_PPROP
SAP Service and Asset Manager 1911	DEST_SAM1911_PPROP
SAP Service and Asset Manager 2005	DEST_SAM2005_PPROP
SAP Service and Asset Manager 2010	DEST_SAM2010_PPROP
SAP Service and Asset Manager 2105	DEST_SAM2105_PPROP
SAP Service and Asset Manager 2110	DEST_SAM2110_PPROP
SAP Service and Asset Manager 2.0 with Meter Management	DEST_SMM10_PPROP
SAP Service and Asset Manager 4.0 online service	DEST_SAM40_ONLINE_PPROP
SAP Service and Asset Manager 1911 online service	DEST_SAM1911_ONLINE_PPROP
SAP Service and Asset Manager 2005 online service	DEST_SAM2005_ONLINE_PPROP
SAP Service and Asset Manager 2010 online service	DEST_SAM2010_ONLINE_PPROP
SAP Service and Asset Manager 2105 online service	DEST_SAM2105_ONLINE_PPROP

     - If the application metadata is edited to use a different destination name than what is defined in the `assetmanager.service` file, the created destination on the SAP Business Technology Platform Mobile Services must also reflect the service name. For more information on where to find application metadata, see *Step 1* of the Building the SAP Asset Manager Application Client procedure.
   - **URL:** URL to reach the mobile add-on service through the Cloud Connector

        - The URL specified must use the given host name from the virtual host of the Cloud Connector regardless of port specification.
        - Include the full URL path of the OData service in the URL. All of the resources are specified in the ICF nodes that the OData service resides under.
        - If the Cloud Connector connecting to the Mobile Development Kit client is configured to use HTTPS, you don't have to make the HTTPS specification in the URL. Your on-premise connections through a properly configured Cloud Connector should specify a URL using an HTTPS scheme, with the HTTPS port specified.
        - A sample URL is as follows: `http://<cloud connector virtual host>:<cloud connector virtual port> /sap/opu/odata/<mobile add-on component>/<mobile add-on service>`, with `MERP` an example of `<mobile add-on component>` and `SAP_ASSET_MANAGER_<version>` an example of `<mobile add-on service>`.
    - **Use Cloud Connector:** Selected
    - **Cloud Connector Location ID:** Set to the location ID designated by the Cloud Connector
    - **Maximum Connections:** However, many concurrent users are expected for your application connection
    - **Timeout:** The timeout for each network transmission between the device and back-end system. Settings are based on the following:
        - The expected timeout requirements for the SAP Mobile Add-On for SAP S/4HANA on-premise connections to SAP Service and Asset Manager is 600,000ms.
        - The expected timeout requirements for the SAP Mobile Add-On for SAP ERP ECC6.0 EHP 7 SP14 connections to SAP Service and Asset Manager is 1,200,000ms.
    - **Rewrite Mode:** Rewrite URL. Setting must match the mapping in the Cloud Connector.
        - If the virtual host and the port in the Cloud Connector are set the same as the internal host and the port of the SAP Mobile Add-On OData service, you can set the Rewrite Mode to *Rewrite URL*.
        - If the virtual host and the port in the Cloud Connector don't match the internal host and the port of the SAP Mobile Add-On OData service, set the Rewrite Mode to *Custom Rewrite URL*. Setting the Rewrite Mode to Custom Rewrite URL properly maps the outgoing URLs from the Cloud Connector.
        - **OPTIONAL** If the *Custom Rewrite URL* is set, add an outbound rewrite URL when you create system mapping. For more information, see the Rewrite Modes topic.
    - **Keep X-Forwarded-* Header:** Not applicable for the default shipment of this version of SAP Service and Asset Manager
    - **Custom Headers:** Additional headers needed for OData service
        - If a default client isn't properly set for your oData service, you can override the connected client by adding a custom header with a `<Header Key>` of *sap-client* and a `<Header Value>` of your client number.
    - **SSO Mechanism / Authentication:** Cloud Connector SSO
        - Principal propagation is the officially supported authentication method between the SAP Service and Asset Manager and the SAP Mobile Add-On. For more information, see the topic Configure Principal Propagation to an ABAP System for HTTPS.

8. **Optional:** To debug an existing application, click the *Enable Detailed Event Log* link in the top-right corner of each feature. Ensure that the logs for the feature are enabled.
9. *Save* your changes. Click back to your new application and click the *Security* tab.

   The Security information displays.
10. In *OAuth Settings* section, set the following parameters:

- **Token Lifetime:** Default is set to *12 hours*
- **Refresh Token Lifetime:** Default is set to *30 days*
- **Redirect URL:** If enabling certificate-based authentication, ensure that the field contains the same *samclient* URL scheme that is contained in the *Onboarding URL* scheme. The redirect URL must also match the redirect URL defined for the SAP Service and Asset Manager application.
- **Client ID:** ID of the client used to identify the application to the OAuth client

11. Make a note of the following settings, as you need them during the Building the SAP Service and Asset Manager Application Client [page 65] procedure:
    - **Client ID**
    - **Redirect URL**
    - **Authorization Endpoint**
    - **Token Endpoint**
12. Click back to the main application page and click the *APIs* tab. Make a note of the *Server* setting, as you need it during the Building the SAP Service and Asset Manager Application Client [page 65] procedure.

3.5.2.1 Creating an Application in SAP Business Application Studio

Procedure

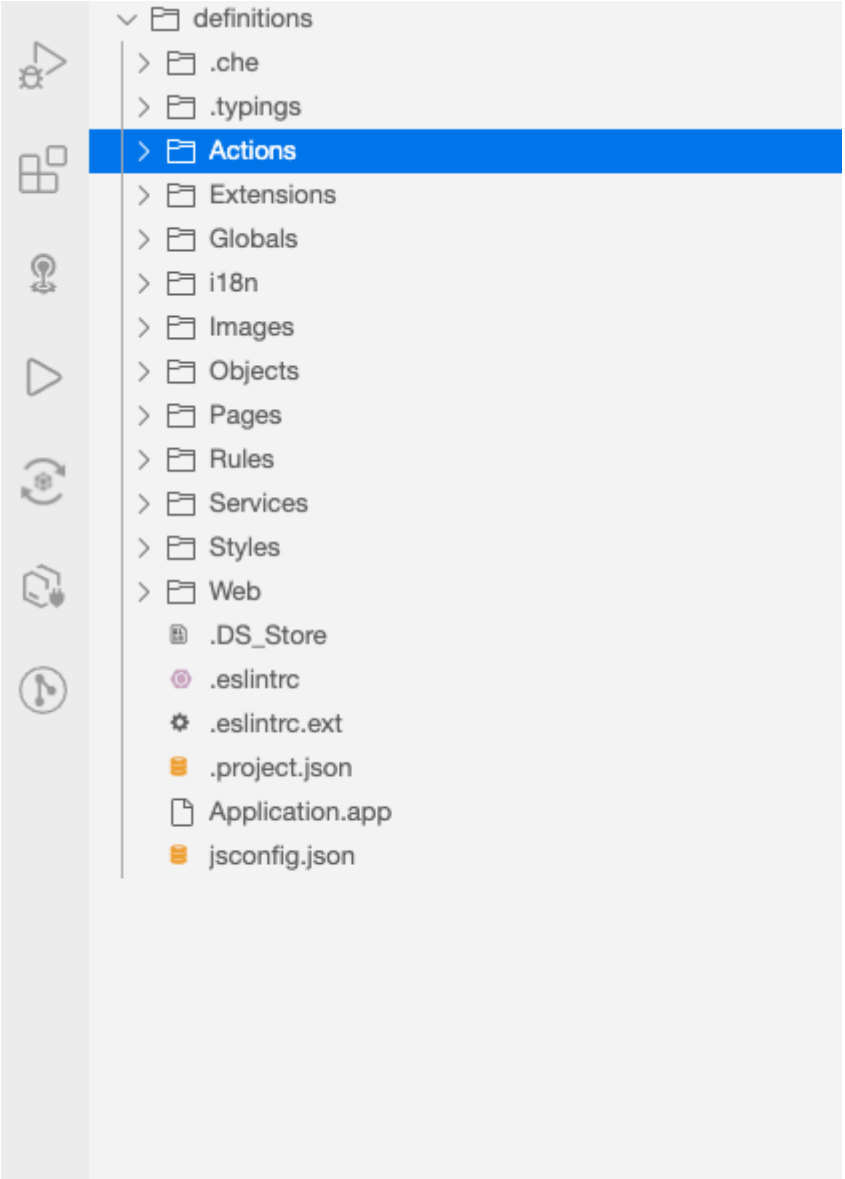
1. Import SAP Service Asset Manager project into the SAP Business Application Studio.
    a. Launch the *Dev* space in SAP Business Application Studio.
    b. Click *Import* on the welcome page, or select *Import Project* from the command palette.
    c. Select the desired ZIP or TAR file.
    d. Select *Open*.

    The project is added to your dev space in a new workspace or as part of a multi-root workspace, depending on the workspace preferences that you defined.

> **i Note**
>
> If you've an SAP Web IDE project or a project you worked on in another IDE, you can save it as a ZIP or a TAR file and import it to SAP Business Application Studio.

2. Get familiar with the generated project structure.

    This is how the project structure looks like within the workspace.

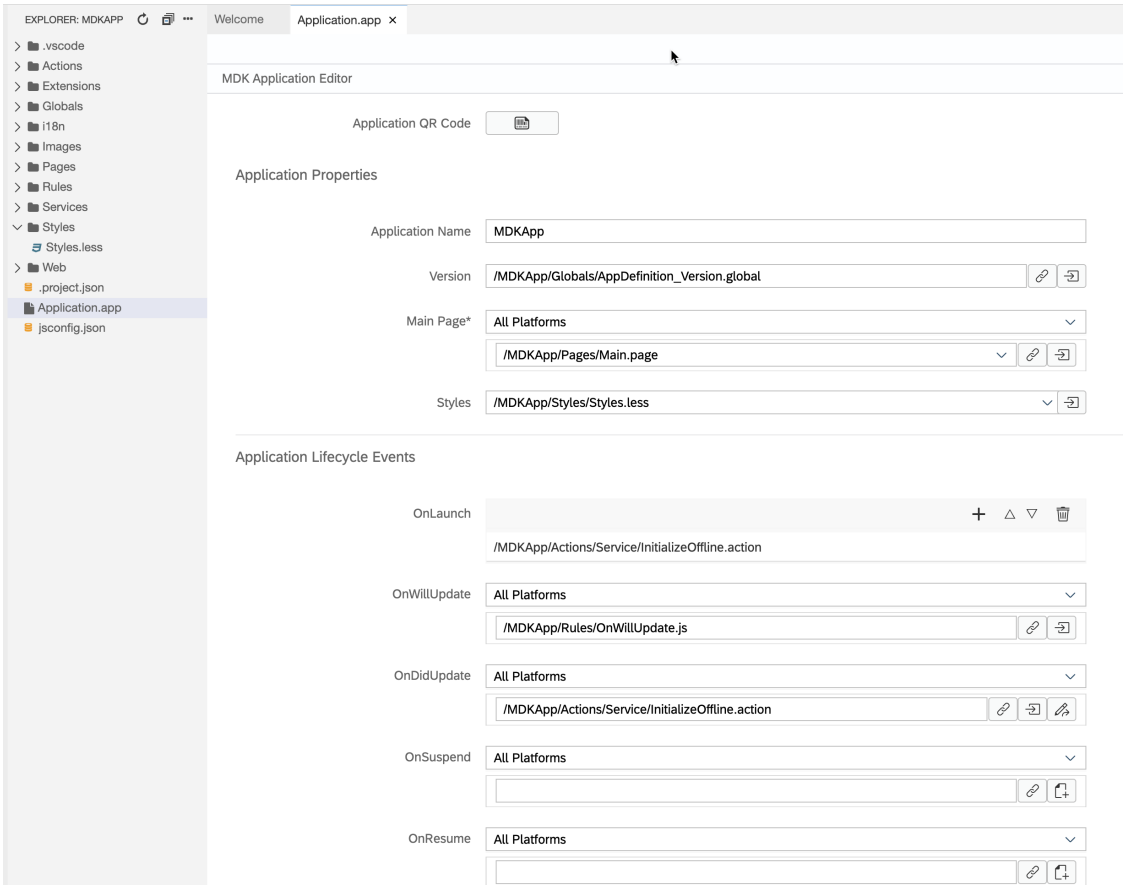


These are the metadata definitions available in the editor and the format in which these metadata definitions are stored in the editor:

- `InitializeOffline.action`: For Mobile applications, this action binds the application to the Mobile Services Offline OData server and downloads the required data to the offline store on the mobile device. For Web applications, it will initialize the service to be consumed in online mode.
- `DownloadOffline.action` and `UploadOffline.action`: These actions are applicable to Mobile client only. Using app initialization, data is downloaded to the offline store. If you want to have the application download any updated data from the backend server or upload changed data to the backend server, these actions will be needed.
- Success & Failure Message action: Here are some messages showing up in the app on a successful or failure of data initialization, sync etc.
- `Main.page`: This is the first page of your MDK application that is shown. For this application you will use this as a launching page to get to application functionality.

- `OnWillUpdate.js`: This rule is applicable to Mobile client only. MDK applications automatically download updates and apply them to the client without the end-user needing to take any action. The OnWillUpdate rule empowers the user to run business logic before the new definitions are applied. This allows the app designer to include logic to prompt the user to accept or defer applying the new definitions based on their current activity. For example, if the end-user is currently adding new customer details or in the middle of a transaction, they will be able to defer the update. The app will prompt again the next time it checks for updates.
- Web: In this folder, you can provide web specific app resource files and configurations.
- `Application.app`: this is the main configuration file for your application from within SAP Business Application Studio. Here you define your start page (here in this tutorial, it is main.page), action settings for different stages of the application session lifecycle, push notifications, and more.

a. Open the application settings in the application editor by clicking the Application.app.



3. Deploy the application.

   So far, you have learned how to build an MDK application in the SAP Business Application Studio editor. Now, you will deploy this application definition to Mobile Services and Cloud Foundry to consume it as Mobile and Web application respectively.

   a. Right-click `Application.app` and select `MDK: Deploy`.
   b. Select deploy target as `Mobile & Cloud`.

      2.1 Select the organization to which you want to connect. (we can skip this step in quick guide)

      2.2 Select the desired space within the organization. (we can skip this step in quick guide)

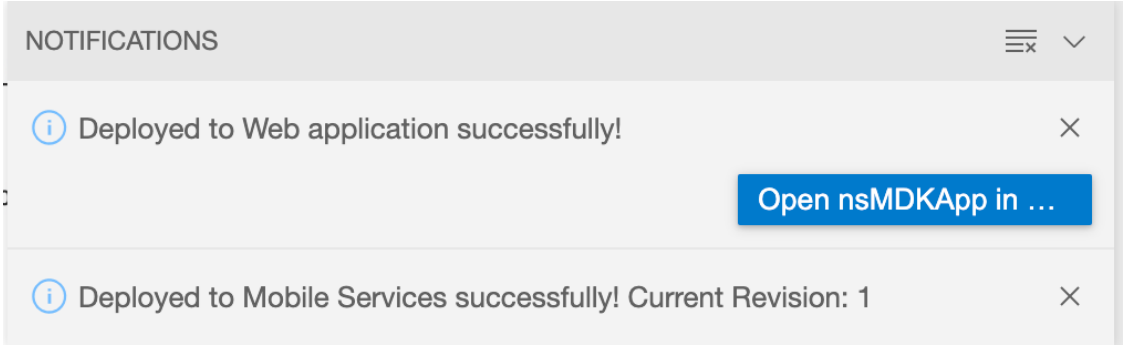
2.3 Enter the application version (we can skip this step in quick guide)

2.4 Include Source map file for debugging? (we can skip this step in quick guide)

2.5 Select Web runtime Version (we can skip this step in quick guide)

c. MDK editor will deploy the metadata to Mobile Services (for Mobile application) followed by to Cloud Foundry (for Web application).

d. You should see successful messages for both deployments. (same image).

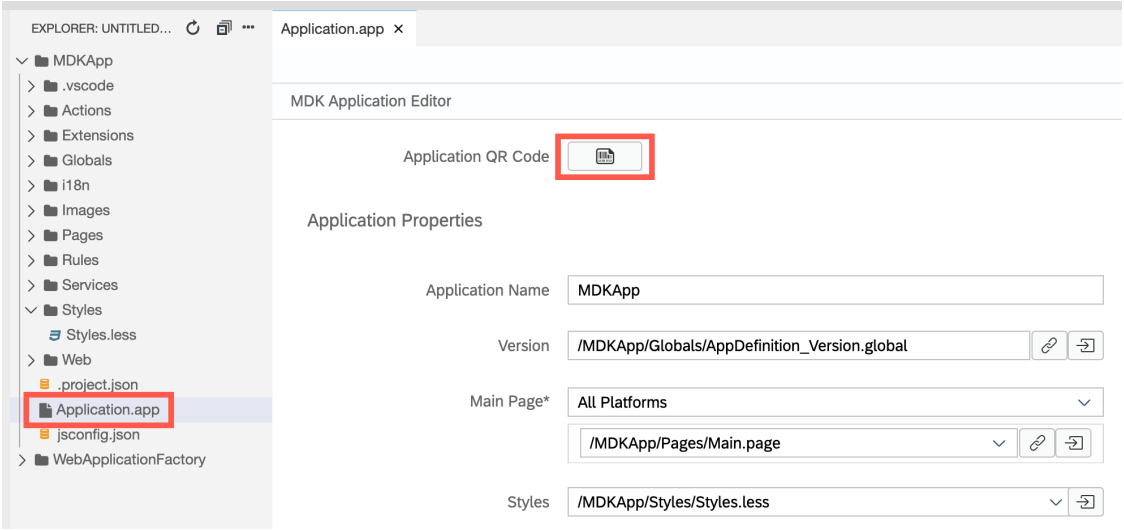


> **i Note**
>
> First web deployment takes 2-3 minutes as it creates five service instances for the application, you can find these details in space cockpit. These are the following:
>
> • XSUAA
>
> • destination
>
> • connectivity
>
> • HTML Repo host
>
> • HTML repo runtime

4. Display the QR code for onboarding the Mobile app.

SAP Business Application Studio has a feature to generate QR code for onboarding the mobile app.

a. Click the Application.app to open it in MDK Application Editor and click Application QR Code icon.



The On-boarding QR code is now displayed.

b. Leave the Onboarding dialog box open for the next step.

5. Run the app.
6.

3.6 Enabling the Mobile Development Kit

Procedure

1. Log in to the SAP Business Technology Platform Cockpit and from *Home* > *Region*, click the *Region* in which your account is based.
2. Select your *Global Account*. From the Global Account page, select the *Subaccounts* tab on the left side. Then select the subaccount where you wish to deploy your applications.
3. Find the *SAP Web IDE Full-Stack* tile. If it is not *Enabled*, enable it. Then click *Go to Service*.

   The SAP Web IDE Full-Stack browser opens.
4. Click the *Preferences* icon ( ) on the left.
5. Click *Extensions*. Ensure the *Mobile Services App Development Tools* editor tile is enabled. If not, enable it by clicking the radio button at the top right of the tile. After enabling the tool, click *Save*.

   The SAP Web IDE prompts to refresh. Refresh the SAP Web IDE to view the new tile.
6. Click the *MDK Development* icon. If the Mobile Development Kit icon does not appear, ensure it is enabled. After enabling, refresh the SAP Web IDE.

Next Steps

Proceed to the Importing Metadata Definitions to the SAP Web IDE Mobile Development Kit [page 40] procedure.

3.6.1 Importing Metadata Definitions to the SAP Web IDE Mobile Development Kit

You can import a project and its resources from your local file system to the SAP Web IDE Mobile Development Kit workspace.

Prerequisites

Ensure that the SAP Service and Asset Manager 2205 and Mobile Development Kit 4.3.2 installation files are downloaded from the SAP Download Center.

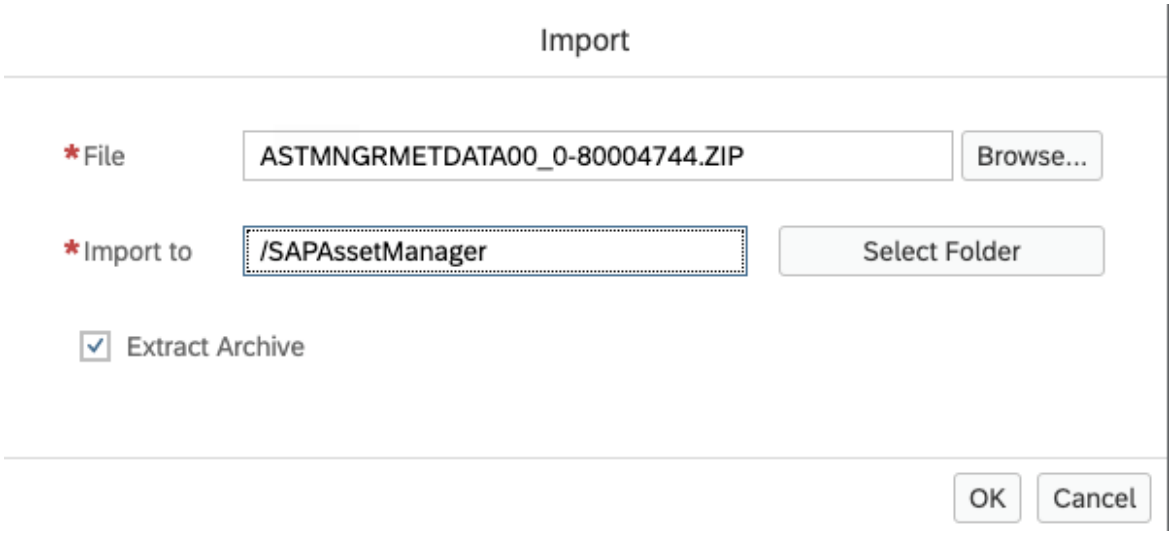
If an earlier version of the Mobile Development Kit is already installed on your back-end system when compared to your version of the SAP Service and Asset Manager application that you are attempting to install, you must also import and deploy the current version of the Mobile Development Kit to the SAP Web IDE.

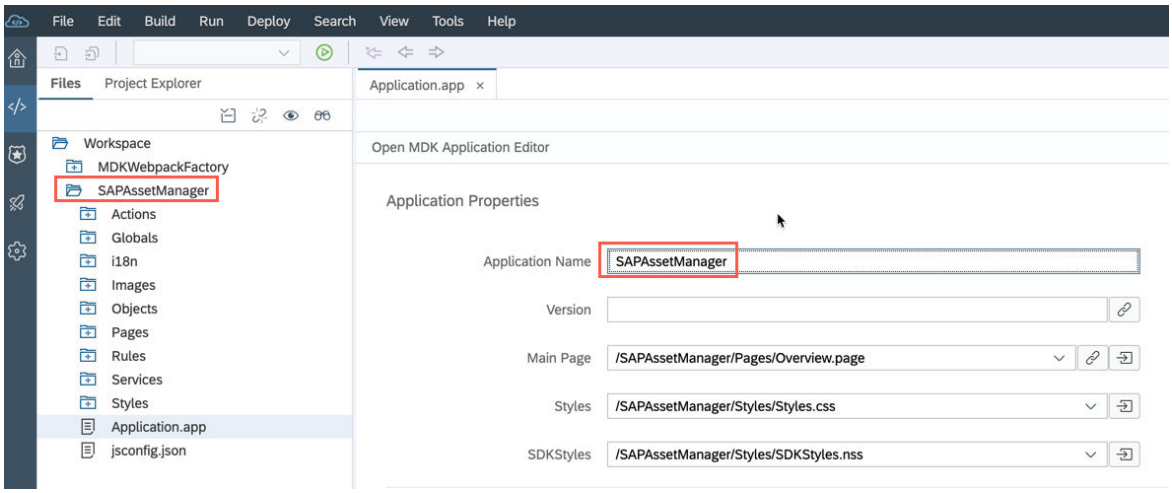
Procedure

1. Click the *MDK Development* icon ().
2. Select the folder to which you want to import your project from your Mobile Development Kit workspace.
3. Import your project. Select *File* > *Import* > *From File System* from the main menu. Alternatively, you can right-click on your *Workspace* folder and choose *Import* from the menu choices.

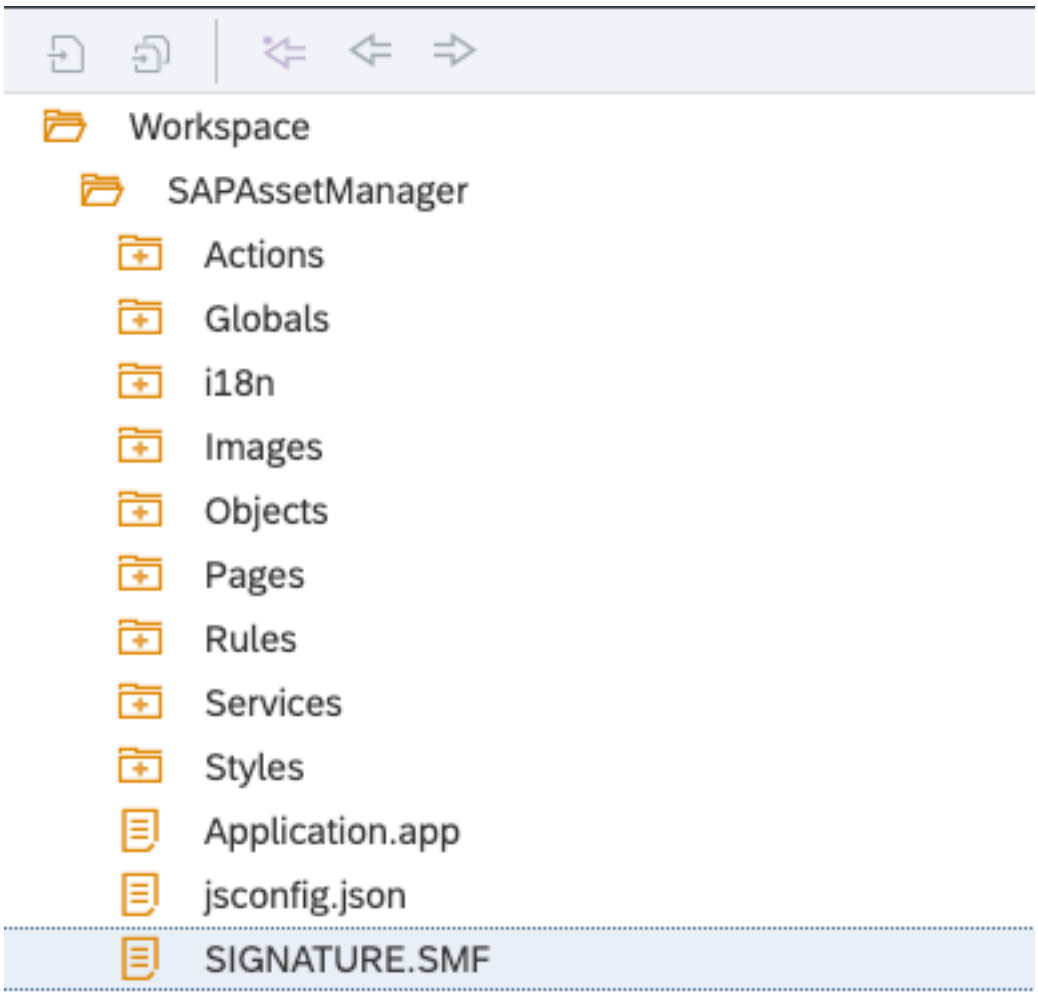
   An Import window displays.
4. Click the *Browse* button and browse to the folder where your ZIP file is located. If the folder listed in the *Import to* field is incorrect, click the *Select Folder* button to choose the correct folder. If you're importing an entire project, the top-level folder is the correct folder.
5. You can optionally change the `<Import To>` folder name to *SAPAssetManager*, but it's not required.

   If you decide to leave the folder name as-is, once the import is complete, hit refresh on your browser to reload the SAP Web IDE. After refreshing, the folder name shows up and correctly matches the *Application Name* in your `Application.app` file.





6. Delete the `SIGNATURE.SMF` file.



7. Check the *Extract Archive* checkbox and click *OK*.

   The project is imported into the selected folder with the same name as the `<Import To>` field.

> **i** Note
>
> If the target folder already exists in your workspace, you're prompted to approve the overwriting of the existing files.

3.6.2 Deploying Metadata Definitions to Mobile Services

Prerequisites

In the SAP Business Technology Platform Cockpit, ensure that the *mobileservices* destination has the additional property of *WebIDEUseage* set with the property of *mobile*.

> **i** Note
>
> SAP Service and Asset Manager metadata versions 2105 or older are not compatible with Mobile Development Kit 6.x. See the Migrating SAP Service and Asset Manager 2105 or Earlier Metadata to Mobile Development Kit 6.x procedure found in the *Upgrading SAP Asset Manager* guide for more details.

Context

Once your desired changes to the metadata definitions for the Mobile Development Kit are complete, use the Mobile Services app development tools deployment feature to deploy the metadata definitions to your application in Mobile Services. Then, when you change the SAP Service and Asset Manager application, and redeploy the metadata files in the Mobile Development Kit editor, a timer looks for new definitions within the bundle.

Procedure

1. Click the *Development* icon ().
2. From your Mobile Development Kit workspace, select the folder from which you want to deploy your project.
3. Right-click and select *File* > *MDK Deploy and Activate*.

   A ZIP file is created of your application definitions.
4. Select the options you would like to deploy with and click *Next*.
5. In the `<Destination Name>` field, select *mobileservices*. In the `<Application ID>` field, select the desired application ID you want to deploy to. Click *Next*.

Results

The `bundle.js` is uploaded from the SAP Web IDE to Mobile Services.

4 Updating Offline Settings for the SAP Asset Manager Application

4.1 Updating Offline Settings for SAP Service and Asset Manager Overview

The offline OData feature of the SAP Business Technology Platform SDK allows for an always-available application that can respond quickly to changes in its connection state.

Instead of calling OData services directly, the OData service call is redirected to the offline OData feature, which mimics a response using the latest synchronized data.

Features of the offline OData feature include:

- Synchronize OData services and run them locally on a mobile device
- MobiLink UltraLite as a client database, which is optionally encrypted
- Optimized for OData services, supporting delta queries to synchronize only new, and changed items
- Offline OData services work in both read and write mode, allowing users to enter data on their mobile device. OData updates are played back to their originating OData service when the user comes back online.
- Data synchronization between the app and the SAP Business Technology Platform leverage the MobiLink protocol, which is designed for synchronizing with remote databases.

4.1.1 Using the OData Offline Configuration Builder to Update Offline OData Settings

Context

> **i** Note
>
> The configuration files are always needed, even if you don't plan to change the default behavior of the application.

> **i** Note
>
> The following procedure is handled as part of the */MERP/CPMS_APPCREATE* process and is only necessary during updates or if you're manually creating an application.

If you change the OData model to add or remove entity sets, update the corresponding configuration file to reflect the new data distribution rules associated with the model change.

These offline OData settings are created using the oDataOffline Configuration Builder. As the oData model in the ConfigPanel changes, the file generated by the program reflects the changes in that system. You must regenerate the file anytime you change the model.

See the following topics for additional configuration information needed for your offline settings:

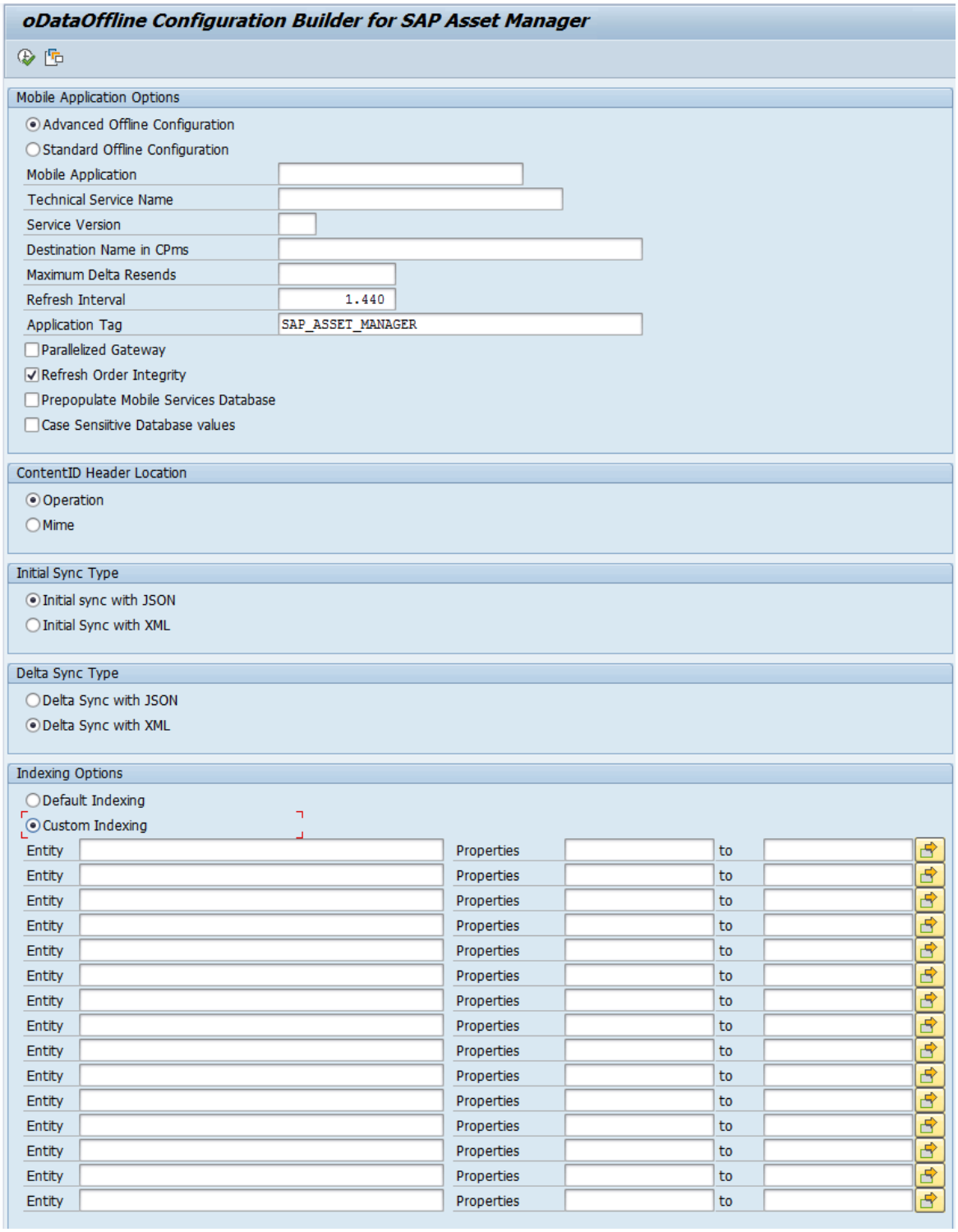
- For detailed information on developing and modifying offline applications on the SAP Business Technology Platform SDK, see the following guides:
  - Developing Offline Applications and the associated subtopics
- For detailed information on developing and modifying offline applications on the SAP Business Technology Platform SDK for Android, see the SAP Cloud Platform SDK for Android portal page.
- For information on defining offline settings for mobile applications through the SAP BTP services, see specifically *Step 6* of the Defining Offline Settings for Applications topic and the associated subtopics.

Procedure

1. Using the SAP GUI, enter transaction *SE38*, then the program */MERP/CORE_OFFLINE_CONFIG_PROG*.
2. Tap the *Variant Catalog* icon. Choose *SAP&SAM<version>* from the list.

   The OfflineOData Configuration Builder for SAP Service and Asset Manager displays. Some fields are automatically filled in for you. The remaining empty fields are mandatory.
3. Fill in the empty fields in all sections. See the following list for details on how to obtain the information required:

   The following screenshot and information on all fields uses the *Advanced Offline Configuration* selection. Choose the *Standard Offline Configuration* to view only the mandatory fields.



i Note

For detailed information on parameters listed in the ODataOffline Configuration Builder and found in the Application Configuration file, see the topic Application Configuration File.

- **Mobile Application Options:**
  - Mobile Application (mandatory): The name of your mobile app as shown in the *Mobile Application Configuration* page in the ConfigPanel
  - Technical Service Name (mandatory): The technical service name listed in the ConfigPanel at *oData Service Assignment* > *oData Service Assignment List table* > *Service column*
  - Service Version (mandatory): The service version is the version listed in the ConfigPanel at *oData Service Assignment* > *oData Service Assignment List table* > *Service Version column*
  - Destination Name in SAP BTP services: By default, the destination name is the name associated with your app in the app store. If you're using a custom destination, find the destination name in the *Destinations* page of the SAP BTP cockpit.
  - Maximum Delta Resends: Default is *5*. When multiple defining queries are delta enabled, the delta requests are automatically sent to the back end repeatedly until all delta responses are empty, in order to ensure data consistency. Use this option to configure the number of times a delta link is resent during a download.
  - Refresh Interval: Default is *15 minutes*. The interval time, in minutes, between the downloads of the shared data.
  - Application Tag: Defines the product license that the mobile application is associated with. Do not change the application tag during the installation process.
  - Parallellized Gateway: Selection required if you're running the SAP Service and Asset Manager application with parallelized gateways turned on. See the Defining Parallelization of Batch Queries topic for more information.
  - Refresh Order Integrity: Default is *N*. Determines whether or not to download defining queries in the order in which they appear in the configuration file. If *Y*, the defining queries are downloaded in the order in which they are defined in the configuration file.
  - Prepopulate Mobile Services Database: For SAP Service and Asset Manager, the default is *N*.

Neo	Specifies the type of data that is pre-populated when creating a new database. ○ *N:* Creates an empty database with only a schema. The database is refreshed automatically after it is deployed to the client to obtain data. ○ *SHARED-ONLY:* The database includes only shared data and no user-specific data. The database is refreshed automatically after it is deployed to the client to obtain user-specific data. Default value is *N*

Cloud Foundry	Specifies the type of data that is pre-populated when creating a new database. ○ *Y:* Both shared and unshared data is included in the database when it is created on the server. The fully populated database, as defined by the defining queries, is deployed to the client. ○ *N:* Creates an empty database with only a schema. The database is refreshed automatically after it is deployed to the client to obtain data. ○ *SHARED-ONLY:* The database includes only shared data and no user-specific data. The database is refreshed automatically after it is deployed to the client to obtain user-specific data. Default value is *Y*

    - Case Sensitive Database Values: Set to *N*. If set to *Y*, alphabetical sorting in SAP Service and Asset Manager does not work properly.

- **ContentID Header Location:** Default is *operation*. In a batch request, the header section of the OData change set to put the ContentID header. The option that is supported depends on which version of NetWeaver Gateway you're using.
    - mime: The ContentID header is included in the mime header section of the OData change set
    - operation: The ContentID header is included in the operation header section of the OData change set. Use this option when the back end does not allow the ContentID header to be in the mime header section
- **Initial Sync Type:** Recommended selection is *Initial Sync with JSON*. Whenever an initial sync is performed, the client sends information back in either JSON or XML. If your Gateway system doesn't handle json, select *Initial Sync with XML*.
- **Delta Sync Type:** Whenever a delta sync is performed, the client sends information back in either JSON or XML. Default is *Initial Sync with XML*. XML is the only supported delta sync format for SAP back ends.
- **Indexing Options:** Indexing pre-loads database values onto the mobile device with an index on the values. Indexing database values renders the values faster on the device, however the sync process becomes slower. The more indexes you add, the screen response becomes faster, but the sync time becomes slower.
    - Default Indexing: Recommended.
    - Custom Indexing: You can index any entity set, up to 15 entity sets.

4. *Execute* the program.

   Your offline OData configuration file is created and is ready to upload using the SAP BTP cockpit.
5. Download the resulting `.ini` file to a local folder. Then upload the `.ini` file to the *Offline* tile in the SAP BTP cockpit.

4.2 SAP Service and Asset Manager Entity Sets

The `INI` configuration file found on the SAP Business Technology Platform Mobile Services defines the order in which the defining requests are sent to call entity sets.

Detailed information on the `INI` configuration file settings are found in the Application Configuration File topic.

Ordered Entity Sets

The following entity sets must be retrieved in order:

Order #	Entity Set Name	Batched	Sync Priority	Entity Type	Mobile OData Object	Lead Entity
1	MyWorkOrderHeaders	All Work Orders	10	MyWorkOrderHeader	SAM2205_WORK_ORDER_GENERIC	X
2	MyWorkOrderOperations		10	MyWorkOrderOperation	SAM2205_WORK_ORDER_GENERIC	
3	MyWorkOrderSubOperations		10	MyWorkOrderSubOperation	SAM2205_WORK_ORDER_GENERIC	
4	MyWorkOrderComponents		10	MyWorkOrderComponent	SAM2205_WORK_ORDER_GENERIC	
5	MyWorkOrderComponentMatDocs		10	MyWorkOrderComponenetMatDoc	SAM2205_WORK_ORDER_GENERIC	
6	MyWorkOrderTools		10	MyWorkOrderTools	SAM2205_WORK_ORDER_GENERIC	
7	MyWorkOrderDocuments		10	MyWorkOrderDocument	SAM2205_WORK_ORDER_GENERIC	
8	MyWorkOrderGeometries		10	MyWorkOrderGeometry	SAM2205_WORK_ORDER_GENERIC	
9	MyWorkOrderPartners		10	MyWorkOrderPartner	SAM2205_WORK_ORDER_GENERIC	
10	MyWorkOrderComponentLongTexts		10	MyWorkOrderComponentLongText	SAM2205_WORK_ORDER_GENERIC	
11	MyWorkOrderHeaderLongTexts		10	MyWorkOrderHeaderLongText	SAM2205_WORK_ORDER_GENERIC	

Order #	Entity Set Name	Batched	Sync Priority	Entity Type	Mobile OData Object	Lead Entity
12	MyWorkOrderOperationLongTexts		10	MyWorkOrderOperationLongText	SAM2205_WORK_ORDER_GENERIC	
13	MyWorkOrderSubOpLongTexts		10	MyWorkOrderSubOpLongText	SAM2205_WORK_ORDER_GENERIC	
14	MyWorkOrderSales		10	MyWorkOrderSale	SAM2205_WORK_ORDER_GENERIC	
	MarkedJobs		12	MarkedJob	SAM2205_MARKED_JOBS	
15	MyRoutes	All Routes	15	MyRoute	SAM2205__ROUTE	X
16	MyTechObjects		15	MyTechObject	SAM2205__ROUTE	
17	MyRouteStops		15	MyRouteStop	SAM2205__ROUTE	
18	MyRoutePoints		15	MyRoutePoint	SAM2205__ROUTE	
19	MyNotificationHeaders	All Notifications	20	MyNotificationHeader	SAM2205_NOTIFICATION_GENERIC	X
20	MyNotificationItems		20	MyNotificationItem	SAM2205_NOTIFICATION_GENERIC	
21	MyNotificationTasks		20	MyNotificationTask	SAM2205_NOTIFICATION_GENERIC	
22	MyNotificationItemActivities		20	MyNotificationItemActivity	SAM2205_NOTIFICATION_GENERIC	
23	MyNotificationItemCauses		20	MyNotificationItemCause	SAM2205_NOTIFICATION_GENERIC	
24	MyNotificationItemTasks		20	MyNotificationItemTask	SAM2205_NOTIFICATION_GENERIC	
25	MyNotificationPartners		20	MyNotificationPartner	SAM2205_NOTIFICATION_GENERIC	
26	MyNotifDocuments		20	MyNotifDocument	SAM2205_NOTIFICATION_GENERIC	
27	MyNotifGeometries		20	MyNotifGeometry	SAM2205_NOTIFICATION_GENERIC	
28	MyNotifHeaderLongTexts		20	MyNotifHeaderLongText	SAM2205_NOTIFICATION_GENERIC	

Order #	Entity Set Name	Batched	Sync Priority	Entity Type	Mobile OData Object	Lead Entity
29	MyNotificationActivities		20	MyNotificationActivity	SAM2205_NOTIFICATION_GENERIC	
30	MyNotifActivityLongTexts		20	MyNotifActivityLongText	SAM2205_NOTIFICATION_GENERIC	
31	MyNotifItemActivityLongTexts		20	MyNotifItemActivityLongText	SAM2205_NOTIFICATION_GENERIC	
32	MyNotifItemCauseLongTexts		20	MyNotifItemCauseLongText	SAM2205_NOTIFICATION_GENERIC	
33	MyNotifItemLongTexts		20	MyNotifItemLongText	SAM2205_NOTIFICATION_GENERIC	
34	MyNotifItemTaskLongTexts		20	MyNotifItemTaskLongText	SAM2205_NOTIFICATION_GENERIC	
35	MyNotifTaskLongTexts		20	MyNotifTaskLongText	SAM2205_NOTIFICATION_GENERIC	
36	MyNotificationSales		20	MyNotificationSale	SAM2205_NOTIFICATION_GENERIC	
37	WorkOrderHistories	All Work Order Histories	30	WorkOrderHistory	SAM2205_WORK_ORDER_HISTORY	X
38	WorkOrderHistoryTexts		30	WorkOrderHistoryText	SAM2205_WORK_ORDER_HISTORY	
39	WorkOrderTransfers		31	WorkOrderTransfer	SAM2205_WORK_ORDER_TRANSFER	
40	OrderISULinks		40	OrderISULink	SAM2205_ORDER_ISULINK	
41	NotificationHistories	All Notification Histories	50	NotificationHistory	SAM2205_NOTIFICATION_HISTORY	X
42	NotificationHistoryTexts		50	NotificationHistoryText	SAM2205_NOTIFICATION_HISTORY	
45	Confirmations	All Confirmations	60	Confirmation	SAM2205_PM_CONFIRMATION	X
46	ConfirmationLongTexts		60	ConfirmationLongText	SAM2205_PM_CONFIRMATION	

Order #	Entity Set Name	Batched	Sync Priority	Entity Type	Mobile OData Object	Lead Entity
47	ConfirmationOverviewRows		60	ConfirmationOverviewRow	SAM2205_PM_CONFIRMATION	
48	MeterReadingUnits	All Street Routes	65	MeterReadingUnit	SAM2205_METER_READING_UNIT	X
49	StreetRoutes		66	StreetRoute	SAM2205_STREET_ROUTE	X
50	StreetRouteConnectionObjects		66	StreetRouteConnectionObject	SAM2205_STREET_ROUTE	
51	ConnectionObjects	All Connection Objects	70	ConnectionObject	SAM2205_CONNECTION_OBJECT	X
52	ConnectionObjectMRNotes		70	ConnectionObjectMRNote	SAM2205_CONNECTION_OBJECT	
53	DeviceLocations	All Device Locations	75	DeviceLocation	SAM2205_DEVICE_LOCATION	X
54	DeviceLocationMRNotes		75	DeviceLocationMRNote	SAM2205_DEVICE_LOCATION	
55	MyFunctionalLocations	All Functional Locations	80	MyFunctionalLocation	SAM2205_FUNCLOC	X
56	MyFuncLocDocuments		80	MyFuncLocDocument	SAM2205_FUNCLOC	
57	MyFuncLocGeometries		80	MyFuncLocGeometry	SAM2205_FUNCLOC	
58	MyFuncLocObjectStatuses		80	MyFuncLocObjectStatus	SAM2205_FUNCLOC	
59	MyFuncLocUserStatuses		80	MyFuncLocUserStatus	SAM2205_FUNCLOC	
60	MyFuncLocSystemStatuses		80	MyFuncLocSystemStatus	SAM2205_FUNCLOC	
61	MyFuncLocPartners		80	MyFuncLocPartner	SAM2205_FUNCLOC	
62	MyFuncLocClasses		80	MyFuncLocClasses	SAM2205_FUNCLOC	
63	MyFuncLocClassCharValues		80	MyFuncLocClassCharValue	SAM2205_FUNCLOC	
64	MyEquipments	All Equipments	90	MyEquipment	SAM2205_EQUIPMENT	X
65	MyEquipDocuments		90	MyEquipDocument	SAM2205_EQUIPMENT	

Order #	Entity Set Name	Batched	Sync Priority	Entity Type	Mobile OData Object	Lead Entity
66	MyEquipGeometries		90	MyEquipGeometry	SAM2205_EQUIPMENT	
67	MyEquipObjectStatuses		90	MyEquipObjectStatus	SAM2205_EQUIPMENT	
68	MyEquipSystemStatuses		90	MyEquipSystemStatus	SAM2205_EQUIPMENT	
69	MyEquipUserStatuses		90	MyEquipUserStatus	SAM2205_EQUIPMENT	
70	MyEquipPartners		90	MyEquipPartner	SAM2205_EQUIPMENT	
71	MyEquipWarranties		90	MyEquipWarranty	SAM2205_EQUIPMENT	
72	MyEquipWarrantyLongTexts		90	MyEquipWarrantyLongText	SAM2205_EQUIPMENT	
73	MyEquipClasses		90	MyEquipClass	SAM2205_EQUIPMENT	
74	MyEquipClassCharValues		90	MyEquipClassCharValue	SAM2205_EQUIPMENT	
75	Fleets		91	Fleet	SAM2205_FLEET	
76	ActivityReasons		100	ActivityReason	SAM2205_ACTIVITY_REASON	
77	Devices	All Devices	110	Device	SAM2205_DEVICE	X
78	DeviceGoodsMovements		110	DeviceGoodsMovement	SAM2205_DEVICE	
79	DeviceMeterReadings		110	DeviceMeterReading	SAM2205_DEVICE	
80	Installations		120	Installation	SAM2205_INSTALLATION	X
81	MeterReadings	All Meter Readers	130	MeterReading	SAM2205_METER_READING	X
82	PeriodicMeterReadings		131	PeriodicMeterReading	SAM2205_MR_PERIODIC	
83	MeterReadingLimits		132	MeterReadingLimit	SAM2205_METER_READING_LIMITS	
84	Premises		140	Premise	SAM2205_PREMISE	
85	RegisterGroups	All Registers	150	RegisterGroup	SAM2205_REGISTER	X

Order #	Entity Set Name	Batched	Sync Priority	Entity Type	Mobile OData Object	Lead Entity
86	Registers		160	Register	SAM2205_REGISTER	
87	ClassTypes	All Classifications	170	ClassType	SAM2205_CLASS_TYPE	
88	ClassDefinitions		171	ClassDefinition	SAM2205_CLASSIFICATION_GENERIC	X
89	ClassCharacteristics		171	ClassCharacteristic	SAM2205_CLASSIFICATION_GENERIC	
90	Characteristics		171	Characteristic	SAM2205_CLASSIFICATION_GENERIC	
91	ClassCharacteristicValues		171	ClassCharacteristicValue	SAM2205_CLASSIFICATION_GENERIC	
92	Documents		180	Document	SAM2205_DOCUMENT	
93	Geometries		190	Geometry	SAM2205_GIS_OBJECT_DATA	
94	MeasuringPoints	All Measuring Points	200	MeasuringPoint	SAM2205_MEASURING_POINT	X
95	MeasuringPointTexts		200	MeasuringPointText	SAM2205_MEASURING_POINT	
96	MeasurementDocuments	All Measurement Documents	210	MeasurementDocument	SAM2205_MEASUREMENT_DOCUMENT	X
97	MeasurementDocumentLongTexts		210	MeasurementDocumentLongText	SAM2205_MEASUREMENT_DOCUMENT	
98	Materials	All Materials	220	Material	SAM2205_MATERIAL_PLANT_VIEW	X
99	MaterialPlants		220	MaterialPlant	SAM2205_MATERIAL_PLANT_VIEW	
100	MaterialSLocs		220	MaterialSLoc	SAM2205_MATERIAL_PLANT_VIEW	
101	MaterialUOMs		220	MaterialUOM	SAM2205_MATERIAL_PLANT_VIEW	
102	MaterialDocuments	All Material Documents	230	MaterialDocument	SAM2205_MATERIAL_DOCUMENT	X

Order #	Entity Set Name	Batched	Sync Priority	Entity Type	Mobile OData Object	Lead Entity
103	MaterialDocItems		230	MaterialDocItem	SAM2205_MATERIAL_DOCUMENT	
104	MobileStatuses		240	MobileStatus	SAM2205_MOBILE_STATUS	
105	CatsTimesheetOverviewRows	All Timesheets	250	CatsTimesheet	SAM2205_CATS_TIMESHEET	X
106	CatsTimesheets		250	CatsTimesheetOverviewRow	SAM2205_CATS_TIMESHEET	X
107	CatsTimesheetTexts		250	CatsTimesheetText	SAM2205_CATS_TIMESHEET	
108	Employees	All Employees	260	Employee	SAM2205_HR_EMPLOYEE	X
109	EmployeeAddresses		260	EmployeeAddress	SAM2205_HR_EMPLOYEE	
110	EmployeeCommunications		260	EmployeeCommunication	SAM2205_HR_EMPLOYEE	
111	Addresses	All Addresses	270	Address	SAM2205_ADDRESS	X
112	AddressCommunications		270	AddressCommunication	SAM2205_ADDRESS	X
113	AddressesAtWork		271	AddressAtWork	SAM2205_ADDRESS_WORKPLACE	X
114	AddressesAtWorkComm		271	AddressAtWorkComm	SAM2205_ADDRESS_WORKPLACE	X
115	CrewLists	All Crew	280	CrewList	SAM2205_CREW	X
116	CrewListItems		280	CrewListItem	SAM2205_CREW	
117	DisconnectionDocuments	All Disconnections	290	DisconnectionDocument	SAM2205_DISCONNECTION_DOCUMENT	X
118	DisconnectionActivities		290	DisconnectionActivity	SAM2205_DISCONNECTION_DOCUMENT	
119	DisconnectionObjects		290	DisconnectionObject	SAM2205_DISCONNECTION_DOCUMENT	

Non-Ordered Entity Sets

The following entity sets can be retrieved in any order:

Entity Set Name	Sync Priority	Entity Type	Mobile oData Object	Lead Entity
AcctIndicators	0	AcctIndicator	SAM2205_ACCOUNTING_INDICATOR	
ActivityTypes	0	ActivityType	SAM2205_PM_ACTIVITY_TYPE	
AppParameters	0	AppParam	SAM2205_APP_PARAMS	
AttendanceTypes	0	AttendanceType	SAM2205_HR_ATTENDANCE_TYPE	
BusinessAreas	0	BusinessArea	SAM2205_BUSINESS_AREA	
CharValueCodes	0	CharValueCode	SAM2205_CHAR_VALUE_CODE	
COActivityTypes	0	COActivityType	SAM2205_COA_ACTIVITY_TYPE	
ControlKeys	0	ControlKey	SAM2205_OPERATION_CONTROL_KEY	
Countries	0	Country	SAM2205_COUNTRY	
DeviceCategories	0	DeviceCategory	SAM2205_DEVICE_CATEGORY	
DisconnectActivityStatuses	0	DisconnectActivityStatus	SAM2205_DISCONNECTION_STATUS	
DisconnectionActivityTypes	0	DisconnectionActivityType	SAM2205_DISCON_ACTIVITY_CATEGORY	
DisconnectDocStatuses	0	DisconnectDocStatus	SAM2205_DISCONNECT_DOC_STATUS	
DisconnectionReasons	0	DisconnectionReason	SAM2205_DISCONNECTION_REASON	
DisconnectionTypes	0	DisconnectionType	SAM2205_DISCONNECTION_TYPE	

Entity Set Name	Sync Priority	Entity Type	Mobile oData Object	Lead Entity
Divisions	0	Division	SAM2205_DIVISION	
EquipmentCategories	0	EquipmentCategory	SAM2205_EQUIP_CATEGORY	
EquipObjectTypes	0	EquipObjectType	SAM2205_EQUIP_OBJECT_TYPE	
GISMapParameters	0	GISMapParameter	SAM2205_GIS_MAP_CONTROL	
Locations	0	Location	SAM2205_LOCATION	
MeterReadingNotes	0	MeterReadingNote	SAM2205_METER_READING_NOTE	
MeterReadingTypes	0	MeterReadingType	SAM2205_METER_READING_TYPE	
MeterReadingStatuses	0	MeterReadingStatus	SAM2205_METER_READING_STATUS	
MobileStatusMappings	0	MobileStatusMapping	SAM2205_MOBILE_STATUS_MAPPING	
MovementTypes	0	MovementType	SAM2205_MOVEMENT_TYPE	
NotificationTypes	0	NotificationType	SAM2205_NOTIF_TYPE	
OrderTypes	0	OrderType	SAM2205_ORDER_TYPE	
PartnerFunctions	0	PartnerFunction	SAM2205_PARTNER_FUNC	
PartnerDetProcs	0	PartnerDetProc	SAM2205_PARTNER_DET_PROC	
PlannerGroups	0	PlannerGroup	SAM2205_PLANNER_GROUP	
Plants	0	Plant	SAM2205_PLANT	
PMAuthorizationGroups	0	PMAuthorizationGroup	SAM2205_PM_AUTH_GROUP	

Entity Set Name	Sync Priority	Entity Type	Mobile oData Object	Lead Entity
PMCatalogCodes	0	PMCatalogCode	SAM2205_CATALOG_CODES	
PMCatalogProfiles	0	PMCatalogProfile	SAM2205_CATALOG_PROFILE	
Priorities	0	Priority	SAM2205_PRIORITY	
PRTControlKeys	0	PRTControlKey	SAM2205_PRT_CONTROL_KEY	
Regions	0	Region	SAM2205_REGION	
SAPUsers	0	SAPUser	SAM2205_USER_DATA	
SystemStatuses	0	SystemStatus	SAM2205_SYSTEM_STATUS	
UsageUoMs	0	UsageUoM	SAM2205_GENERIC_UOM	
UserGeneralInfos	0	UserGeneralInfo	SAM2205_USER_INFO	X
UserLocation	0	UserLocation	SAM2205_USER_INFO	X
UserSystemInfos	0	UserSystemInfo	SAM2205_USER_INFO	X
UserPreferences	0	UserPreference	SAM2205_USER_INFO	X
UserStatuses	0	UserStatus	SAM2205_USER_STATUS	
VarianceReasons	0	VarianceReason	SAM2205_CONF_VARIANCE_REASON	
WorkCenters	0	WorkCenter	SAM2205_WORKCENTER	
MeterReadingReasons	0	MeterReadingReason	SAM2205_METER_READING_REASON	
PolRegStructElements	0	PolRegStructElement	SAM2205_POL_REG_STRUCT	

5 SAP Service and Asset Manager client Setup

5.1 Deploying the SAM application in Business Application Studio

Procedure

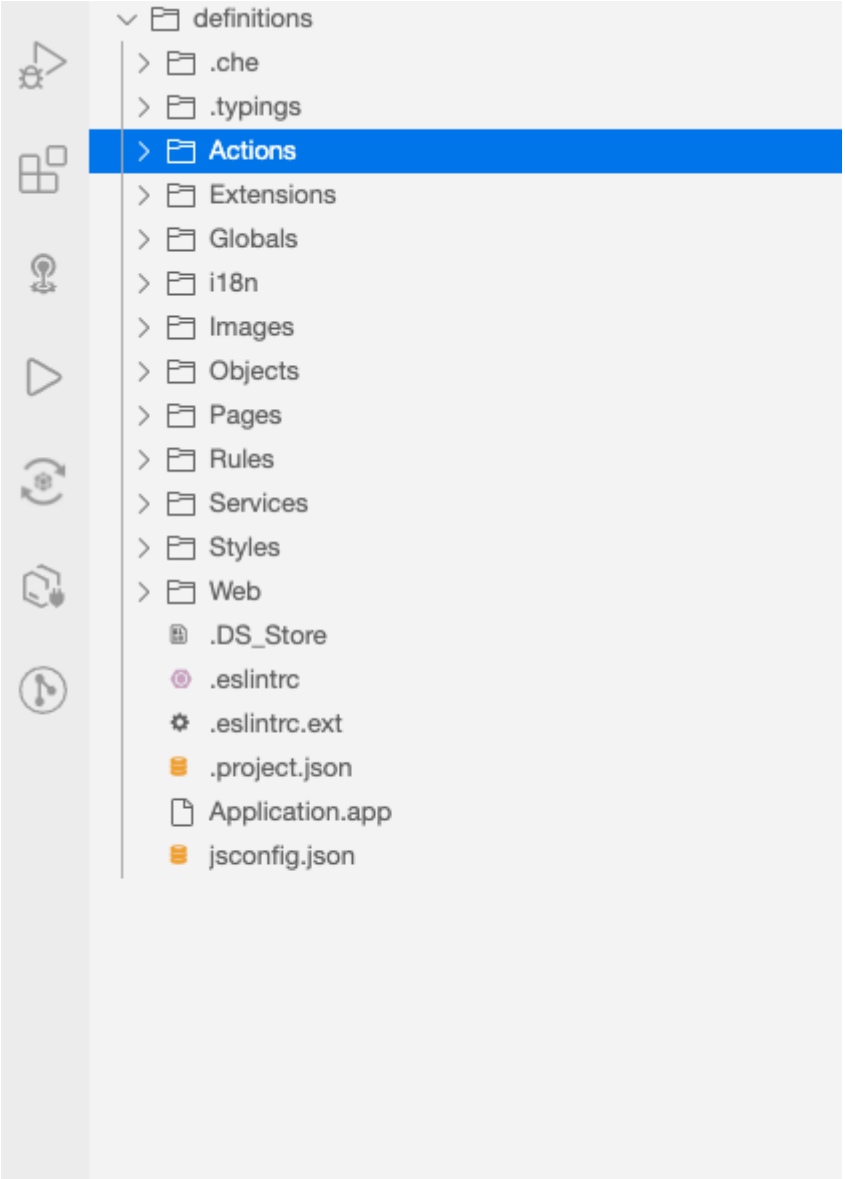
1. Import SAP Service Asset Manager project into the SAP Business Application Studio.
   a. Launch the *Dev* space in SAP Business Application Studio.
   b. Click *Import* on the welcome page, or select *Import Project* from the command palette.
   c. Select the desired ZIP or TAR file.
   d. Select *Open*.

      The project is added to your dev space in a new workspace or as part of a multi-root workspace, depending on the workspace preferences that you defined.

   > **i** Note
   >
   > If you've an SAP Web IDE project or a project you worked on in another IDE, you can save it as a ZIP or a TAR file and import it to SAP Business Application Studio.

2. Get familiar with the generated project structure.

   This is how the project structure looks like within the workspace.

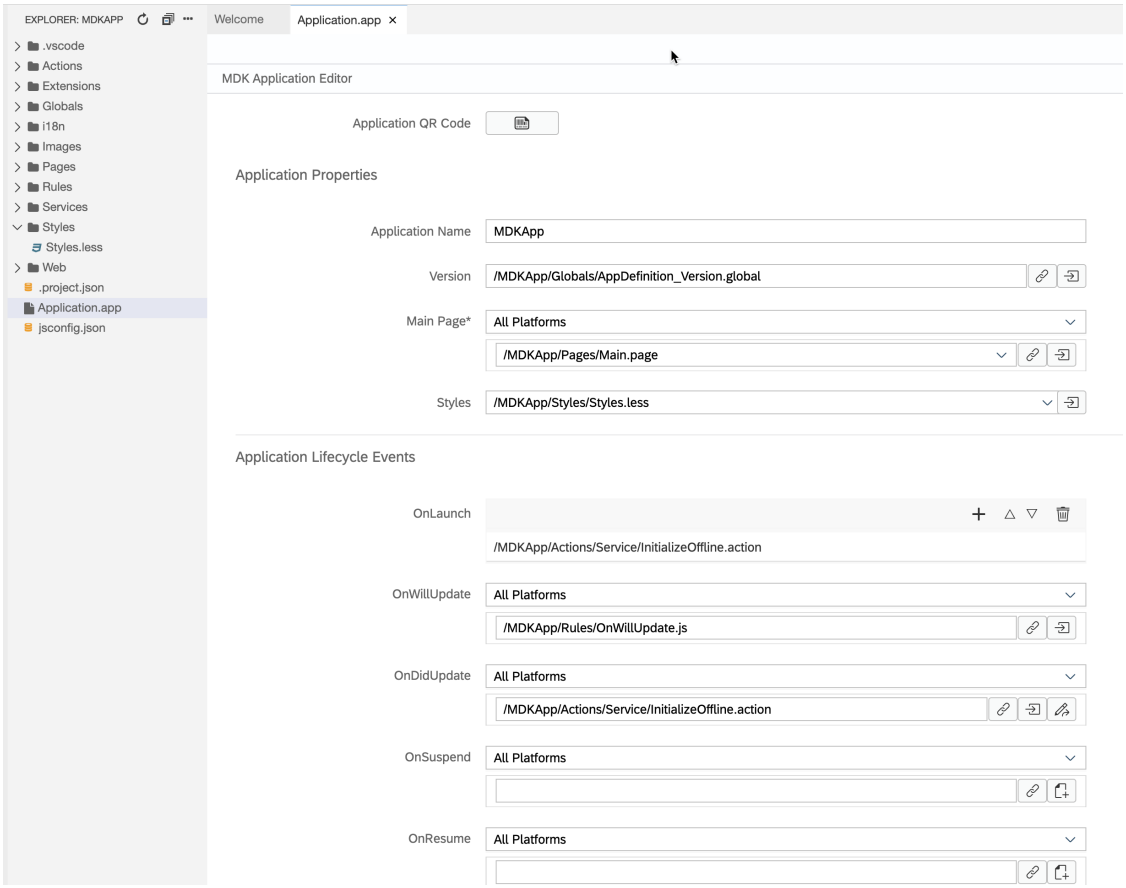


These are the metadata definitions available in the editor and the format in which these metadata definitions are stored in the editor:

- `InitializeOffline.action`: For Mobile applications, this action binds the application to the Mobile Services Offline OData server and downloads the required data to the offline store on the mobile device. For Web applications, it will initialize the service to be consumed in online mode.
- `DownloadOffline.action` and `UploadOffline.action`: These actions are applicable to Mobile client only. Using app initialization, data is downloaded to the offline store. If you want to have the application download any updated data from the backend server or upload changed data to the backend server, these actions will be needed.
- Success & Failure Message action: Here are some messages showing up in the app on a successful or failure of data initialization, sync etc.
- `Main.page`: This is the first page of your MDK application that is shown. For this application you will use this as a launching page to get to application functionality.

- `OnWillUpdate.js`: This rule is applicable to Mobile client only. MDK applications automatically download updates and apply them to the client without the end-user needing to take any action. The OnWillUpdate rule empowers the user to run business logic before the new definitions are applied. This allows the app designer to include logic to prompt the user to accept or defer applying the new definitions based on their current activity. For example, if the end-user is currently adding new customer details or in the middle of a transaction, they will be able to defer the update. The app will prompt again the next time it checks for updates.
- Web: In this folder, you can provide web specific app resource files and configurations.
- `Application.app`: this is the main configuration file for your application from within SAP Business Application Studio. Here you define your start page (here in this tutorial, it is main.page), action settings for different stages of the application session lifecycle, push notifications, and more.

a. Open the application settings in the application editor by clicking the Application.app.



3. Deploy the application.

   So far, you have learned how to build an MDK application in the SAP Business Application Studio editor. Now, you will deploy this application definition to Mobile Services and Cloud Foundry to consume it as Mobile and Web application respectively.

   a. Right-click `Application.app` and select `MDK: Deploy`.
   b. Select deploy target as `Mobile & Cloud`.

      2.1 Select the organization to which you want to connect. (we can skip this step in quick guide)

      2.2 Select the desired space within the organization. (we can skip this step in quick guide)

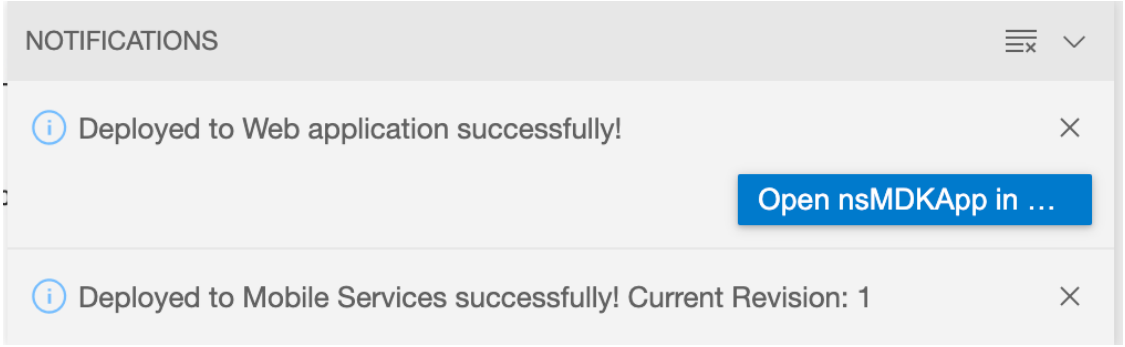
2.3 Enter the application version (we can skip this step in quick guide)

2.4 Include Source map file for debugging? (we can skip this step in quick guide)

2.5 Select Web runtime Version (we can skip this step in quick guide)

c. MDK editor will deploy the metadata to Mobile Services (for Mobile application) followed by to Cloud Foundry (for Web application).

d. You should see successful messages for both deployments. (same image).

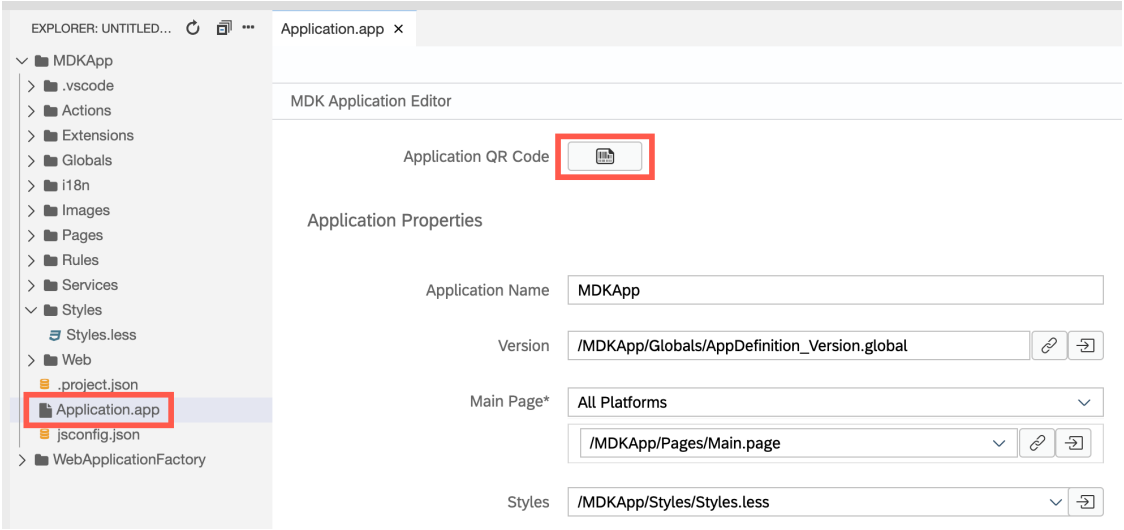


> **i Note**
>
> First web deployment takes 2-3 minutes as it creates five service instances for the application, you can find these details in space cockpit. These are the following:
>
> • XSUAA
>
> • destination
>
> • connectivity
>
> • HTML Repo host
>
> • HTML repo runtime

4. Display the QR code for onboarding the Mobile app.

SAP Business Application Studio has a feature to generate QR code for onboarding the mobile app.

a. Click the Application.app to open it in MDK Application Editor and click Application QR Code icon.



The On-boarding QR code is now displayed.

b. Leave the Onboarding dialog box open for the next step.

5. Run the app.
6.

5.2 Deploying the SAM application in Web IDE

If the extension or installation of additional components onto SAP Service and Asset Manager is required, then the SAP Web IDE is a required deployment.

Find more information at the documentation for the SAP Web IDE Full-Stack.

To deploy an application from the SAP Web IDE to the SAP BTP services instance:

1. Ensure that the *mobileservices* destination in the SAP Business Technology Platform subaccount has the value *mobile* added to the *WebIDEUsage* property.
2. Ensure that the *Mobile Services App Development Tools* is enabled by checking the *Settings* tab in the *Extensions* section in the SAP Web IDE.
3. Download the SAP Service and Asset Manager metadata from the SAP Marketplace. Import the application into the SAP Web IDE.
4. Once the metadata is imported, load the metadata to the MDK perspective.
   Right-click on the app, and select *MDK Deploy and Activate*. Deploy the app to the mobile application.
5. Once the app is loaded, use the connection link builder in the SAP Web IDE to build an onboarding link for the mobile device using the information retrieved from the building of the mobile application.
6. Once the link is built, send the link to the mobile device with SAP Service and Asset Manager installed.
7. Connect to the mobile app, sign in, and update the mobile application when prompted.

5.3 Installing the SAP Service and Asset Manager Client

Procedure

For information on how to install the SAP Service and Asset Manager client, see Installing the SAP Service and Asset Manager Client.

5.4 Building the SAP Service and Asset Manager Application Overview

Use the following information as a reference when building your application using the procedure Building the SAP Service and Asset Manager Application Client [page 65].

Structure of .mdkproject

- **BrandedSettings.json:** Runtime configurations such as security settings, URLs for connecting to the SAP Business Technology Platform Mobile Services, and more
- **MDKProject.json:** Build time configurations such as the application name, version, and bundle ID
- **App_Resources:** Any custom resources used by the application, such as all of your action bar images that are customized for your application.
- **demo:** To make an OData service available in demo mode, include the `.udb` and `.rq.udb` files for that service in this directory
  Currently the SAP Service and Asset Manager application can't take the demo UDBs created by the most recent SAP Business Technology Platform SDK and use them with an older SAP Business Technology Platform SDK version.
  Therefore, use the Android UDBs for demo mode, for both the iOS and Android applications. Failure to do so results in the inability to perform Update or Create OData actions, such as:
  - Changing the mobile status of any work orders
  - Creating reminders
- **extensions:** Include any extensions used by the application in this directory
- **metadata:** Built in metadata for the application

Configuring the MDKProject.json File

The `MDKProject.json` file contains settings that you can only configure before running the `create-client.command`:

- **AppName:** Determines the name of the application project and the app as it appears on a mobile device
- **AppVersion:** The client project application version
- **BaseProject:** The `metadata` subdirectory under the `.mdkproject` structure that contains the main application metadata. The main application metadata is the MDK application, which includes one or more component MDK applications. The component applications are only required if you are customizing your own client.
- **BundleID:** Uniquely identifies the resulting MDK client application on the device. Only one instance of a bundle ID can be installed on a device at a time. If you attempt to install a second application using the same bundle ID, it will overwrite the existing application.
- **Externals:** A list of NPM nodules that should not be included in the application bundle. Use this option for dependencies you expect to be in the environment when the application is built.
  Note that the modules `file-system` and `ui/dialogs` are automatically used as externals as they are already included in the client application.

- **URLScheme:** Allows you to specify a custom URL scheme that opens the client. If the URL includes connection settings such as URL parameters, these settings override the settings used by the client. Defaults to *mdkclient*.

Application Version and Notes on the Settings App

The Mobile Development Kit client tracks several versions, which you can view in the iOS *Settings* menu. These versions are identified as the *application* version, the *definitions* version, and the *frameworks* versions for the frameworks used in the client build.

When generating a client project, you can specify the application version. Specifying the application version allows you to version the client itself, which can be useful if you change extension controls or other branded settings. To specify the application version, specify the `AppVersion` property in the `MDKProject.json` file before running `create.client.command`.

To further customize the entry of your application in the iOS *Settings* menu, you can manually edit `<ProjectDirectory>/app/App_Resources/iOS/Settings.bundle/Root.plist` after the script has completed. You can add new entries, but do not remove existing entries or the application may not function correctly.

For more information, see Implementing an iOS Settings Bundle .

5.5 Building the SAP Service and Asset Manager Application Client

Deploy SAP Service and Asset Manager from the out of the box configuration to set the cloud endpoint authentication URL and the OData service URL. You can also set other configuration values.

Prerequisites

- Verify that your system is set up to build the SAP Service and Asset Manager application by running the MDK Dependencies Installer. This tool detects all the components to install or update, allowing you to update or install them instantly.
  For more information and instructions on how to obtain the MDK Dependencies Installer, see the Building Your MDK Client procedure.
- Complete the Creating an Application in SAP Cloud Platform mobile services procedure.

Context

> **!** Restriction
>
> Develop any customization on the app as a separate component in a Mobile Development Kit project. Developing customizations as a component makes it easier to maintain customizations during upgrades,

as it isolates custom code. Isolating your custom code eliminates the chance of overwriting when you implement a new release.

Use the following procedure to build and brand the SAP Service and Asset Manager application.

Procedure

1. Locate the required installation files from the SAP Download Center and navigating to:

   Software Downloads > *By Category* > *SAP Mobile Solutions* > *SAP Asset Manager*

   > **i Note**
   >
   > For detailed information on SAP Service and Asset Manager and Mobile Development Kit version compatibility, see 2933065.
   >
   > See 2970982 for detailed information when building an iOS client using iOS 14.

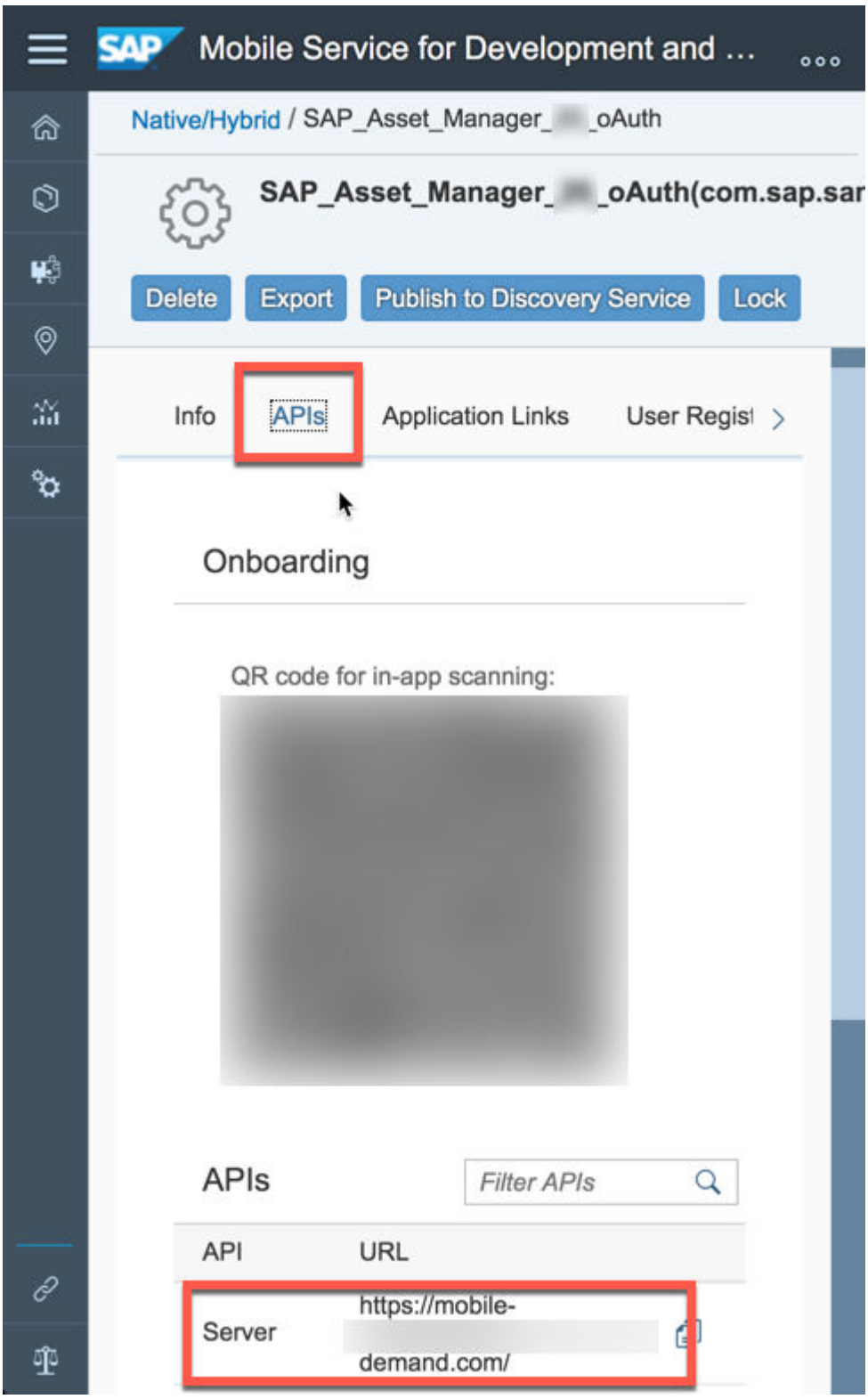
2. Download the files found in the `MDK PLUG-IN SDK` folder. Select the correct version for your installation. Select either iOS or Android, depending on your installation platform.
3. Download the files from either *SAP ASSET MANAGER* for iOS devices or *MOB ASSET MANAGER AND* for Android devices. Select the correct application version folder. In the version folder, download the SDK file and the metadata files found in the following subfolders:
   - `ASSET MANAGER METADATA`: SAP Service and Asset Manager
   - `ASSET MGR BRANDING SDK`: Branding SDK for SAP Service and Asset Manager
4. Create a folder to contain the installation files (`SAPAssetManager`).
5. Extract the SAP Service and Asset Manager branding SDK:
   a. Unzip the `ASSET MGR BRANDING SDK` folder.
   b. Copy the `SAPAssetManager/SAM.mdkproject` folder to the `SAPAssetManager` folder.
   c. Set up the SAP Service and Asset Manager Mobile Development Kit project folders:
      1. In the `SAPAssetManager/SAM.mdkproject` folder, create a folder named `metadata`.
      2. In the `SAPAssetManager/SAM.mdkproject` folder, create a folder named `extensions`.
6. Extract the SAP Service and Asset Manager metadata:
   a. Unzip the `ASSET MANAGER METADATA` file.
   b. Copy all contents of the ZIP file to `SAPAssetManager/SAM.mdkproject/metadata`.
   c. If present, delete the `SIGNATURE.SMF` file from the `SAPAssetManager/SAM.mdkproject/metadata` folder.
7. Extract the MDK plug-in SDK in one of the following ways, based on whether you're building either an iOS or an Android client:

iOS Client	Android Client
1. **Unzip the iOS subfolder found in the** `MDK PLUG-IN SDK` **zip file.**	1. Unzip the Android subfolder found in the `MDK PLUG-IN SDK` zip file.

iOS Client	Android Client
2. **Choose your architecture from one of the following folders:** ○ **Release-iphoneos** ○ **Release-iphonesimulator** ○ **Release-fat (contains both the iphoneos and the iphonesimulator architectures)** 3. **Copy the following folders from your selected architecture folder to the** `SAPAssetManager/SAM.mdkproject/extensions` **folder:** ○ `extension-PDF` ○ `extension-BarcodeScanner` ○ `extension-FieldDataCapture` ○ `extension-MapFramework` ○ `extension-HierarchyFramework`	2. Copy the following folders from your `Universal` folder to the `SAPAssetManager/SAM.mdkproject/extensions` folder: ○ `extension-PDF` ○ `extension-BarcodeScanner` ○ `extension-FieldDataCapture` ○ `extension-MapFramework` ○ `extension-HierarchyFramework`

8. Configure the connection to SAP Business Technology Platform Mobile Services:
   a. Retrieve the following information to establish a connection between the SAP Service and Asset Manager application and the SAP Business Technology Platform Mobile Services:
      - **AppId:** Set to the ID under the Mobile Development Kit that you created in *Step 6* of the Creating an Application in SAP Cloud Platform mobile services procedure.
      - **ClientID:**
        - **Neo:** Set to the oAuth client ID that you created in *Step 12* of the Creating an Application in SAP Cloud Platform mobile services Creating an Application in SAP Business Technology Platform Mobile Services - Neo [page 26] procedure.
        - **Cloud Foundry:** Set to the oAuth client ID that you created in *Step 11* of the Creating an Application in SAP Cloud Platform mobile services Creating an Application in SAP Business Technology Platform Mobile Services - Cloud Foundry [page 32] procedure.

- **SapCloudPlatformEndpoint:** Find the Endpoint setting inside the application list of APIs on the *Mobile Services* under the *Server* API:



> **Note**
>
> By default, the *Server API* has a / at the end of the endpoint URL. Do not add this / into your connection settings.

- **Neo:**
  - **AuthorizationEndpointURL:** Set to the oAuth authorization endpoint URL that you created in *Step 12* of the Creating an Application in SAP Business Technology Platform Mobile Services - Neo [page 26] procedure.
  - **RedirectURL:** Set to the callback URL that is automatically generated with the creation of the oAuth client in *Step 12* of the Creating an Application in SAP Business Technology Platform Mobile Services - Neo [page 26] procedure.
  - **TokenURL:** Set to the token URL that is automatically generated with the creation of the oAuth client in *Step 12* of the Creating an Application in SAP Business Technology Platform Mobile Services - Neo [page 26] procedure.
- **Cloud Foundry:**
  - **AuthorizationEndpointURL:** Set to the oAuth authorization endpoint URL that you created in *Step 11* of the Creating an Application in SAP Business Technology Platform Mobile Services - Cloud Foundry [page 32] procedure.
  - **RedirectURL:** Set to the callback URL that is automatically generated with the creation of the oAuth client in *Step 11* of the Creating an Application in SAP Business Technology Platform Mobile Services - Cloud Foundry [page 32] procedure.
  - **TokenURL:** Set to the token URL that is automatically generated with the creation of the oAuth client in *Step 11* of the Creating an Application in SAP Business Technology Platform Mobile Services - Cloud Foundry [page 32] procedure.

b. Choose your client configuration:

   To preconfigure your client to connect to your mobile application, add the information retrieved in the above step to the *ConnectionSettings* block. When adding additional entries, include a comma after the existing *EnableOverrides* entry. An example is included in the table for reference when you're adding entries to your file.

   If you want to use SAP Service and Asset Manager against different back-end mobile applications (ex: DEV and QA), leave the *ConnectionSettings* as is, and build an onboarding URL for users using the values found in *Step 6a*. See the example in the table for further information on how to connect a client using either of the methods.

   > **Note**
   >
   > If you are using the onboarding URL approach, note that all users must recieve an email with the onboarding URL to connect applications to SAP Business Technology Platform Mobile Services.

c. OPTIONAL: Enable certificate authentication in the Mobile Development Kit:

   In the `BrandedSettings.json` file, `ConnectionSettings` section, set `AllowCerts` to `true`.

Preconfigured Client	Onboarding URL

To configure the client to connect to a specific application, update the following values in the `BrandedSettings.json` file:
Sample Code
`…`
`"ApplicationDisplayName": "<Name of application>",`
`"ConnectionSettings": {`
`"EnableOverrides": true,`
`"AppId": "<Insert AppID value here>",`
`"ClientId": "<Insert ClientID value here>",`
`"SapCloudPlatformEndpoint": "https://<Insert cloud platform endpoint URL here>",`
`"AuthorizationEndpointUrl": "https://<insert authorization endpoint URL here>",`
`"RedirectUrl": "https://<insert redirect URL here>",`
`"TokenUrl": "https://<insert token URL here>",`
`"AllowCerts": <true/false>`
`},`
Using the example as a guide, insert your own connection-specific values where they belong.
Save any changes you make.
	You can generate an onboarding link to overwrite the values discussed in this substep on a device. Use the following format in a URI:
	Sample Code
	`samclient://?AppId=<Insert AppID value here>`
	`&ClientId=<Insert ClientID value here>`
	`&SapCloudPlatformEndpoint=<Insert cloud platform endpoint URL here>`
	`&AuthorizationEndpointUrl=<insert authorization endpoint URL here>`
	`&RedirectUrl=<insert redirect URL here>`
	`&TokenUrl=<insert token URL here>`
	`&ServiceTimeZoneAbbreviation=<insert timezone abbreviation here>`
	Save any changes you make.

9. Edit the project settings:
   a. Open the `SAPAssetManager/SAM.mdkproject/MDKProject.json` file in a text editor. Edit app information such as:
      - Application name on the home screen
      - App version
      - Bundle ID to uniquely identify the application on the device
      - URL scheme for onboarding URLs
   b. Save any changes you make.
10. Set up the Mobile Development Kit Client SDK:
    a. Unzip Mobile Development Kit `MDK PLUG-IN SDK`.
       1. Run the Mobile Development Kit dependencies installer and confirm that your system is ready.
       2. Unzip `MDKClient_SDK.zip` to the new `SAPAssetManager` folder.
    b. To install the necessary dependencies, open a Terminal prompt in the `SAPAssetManager/MDKClient_SDK` directory. Run either `./install.command` on a Mac or `./install.cmd` on a Windows PC. Note that you can build both iOS and an Android on a Mac. You can only build Android if you're using a Windows PC.

> **Note**
>
> An internet connection is required. If you're connecting to the internet through a proxy, configure your settings before running the `./install.command` or `./install.cmd` command.

11. Create the SAP Service and Asset Manager client:
    a. Open a Terminal prompt in the `SAPAssetManager/MDKClient_SDK` directory.
    b. Run the `create-client.command` command if you are on a Mac. Run the `create-client.cmd` command if you are on Windows.
    c. You can either specify command line arguments to point to the `SAM.mdkproject` and the type of client (either device or simulator) you are building, or the script prompts you for the information.

> **Sample Code**
>
> on a Mac

```
$ ./create-client.command
? Enter the path of the .mdkproject directory. ../SAM.mdkproject
Using ../SAM.mdkproject
Using /Users/.../sdk for out directory
? Would you like to build for iOS or Android or All? ios
Building client for ios
? Would you like to build for device or simulator of iOS? device
Building client for device of iOS
Removing existing directory /Users/.../sdk/AssetManager
Creating application AssetManager
```

> **Sample Code**
>
> on a Windows PC

```
>create-client.cmd
? Enter the path of the .mdkproject directory. ..\SAM.mdkproject
Using ..\SAM.mdkproject
Using C:\...\mdk for out directory
Building client for Android
Removing existing directory C:\...\mdk\AssetManager
Creating application AssetManager
```

Results

After `create-client.command` for iOS or `create-client.cmd` for Windows finishes, you're ready to run the client either on the mobile device or on a simulator.

Next Steps

Open a Terminal prompt in the resulting client directory (ex: `SAPAssetManager/MDKClient_SDK/SvcAssetMgr`). Run either the `tns run ios` or the `tns run android` command to start the application, based on your platform.

For iOS installations only, continue to the Allowing Custom URI Schemes [page 72] procedure, which allows SAP Service and Asset Manager to open custom URIs.

If a custom SAP Service and Asset Manager build is needed for Cloud Foundry, see the Creating a Build Job for Customized MDK Clients procedure.

Optional: If you are using custom map icons, follow the instructions in the Customizing Map Icons [page 75] procedure.

5.5.1 Allowing Custom URI Schemes

Prerequisites

You have built and branded the SAP Service and Asset Manager application. For more information, see the following topics:

- Building the SAP Service and Asset Manager Application Overview [page 64]
- Building the SAP Service and Asset Manager Application Client [page 65]

Context

> **i** Note
>
> The following procedure is applicable for iOS installations only. You do not need to perform this procedure if you are installing SAP Service and Asset Manager on Android.

By default, iOS allows third-party apps to specify a limited set of URI schemes:

- http:
- https:
- mailto:
- tel:
- sms:
- facetime:

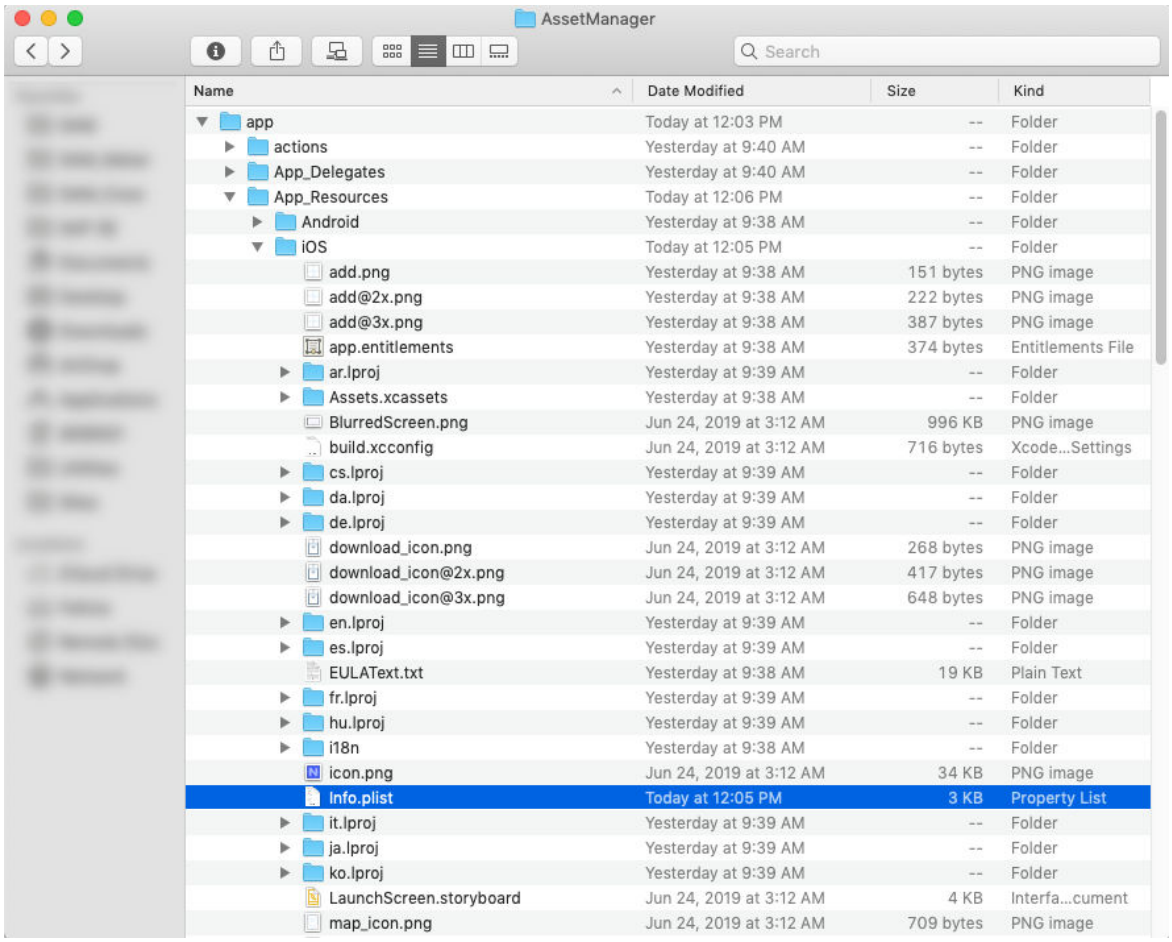
Other applications, such as Microsoft Edge, can support custom URI schemes. For example, if the application is installed, `microsoft-edge-https://www.google.com`, opens up Google in Microsoft Edge for iOS. However, it will only work correctly from a non-third-party iOS application such as Safari.

For additional information on iOS URI schemes, see the Launch Services Keys documentation from Apple.

To allow SAP Service and Asset Manager to open custom URIs, use the following procedure.

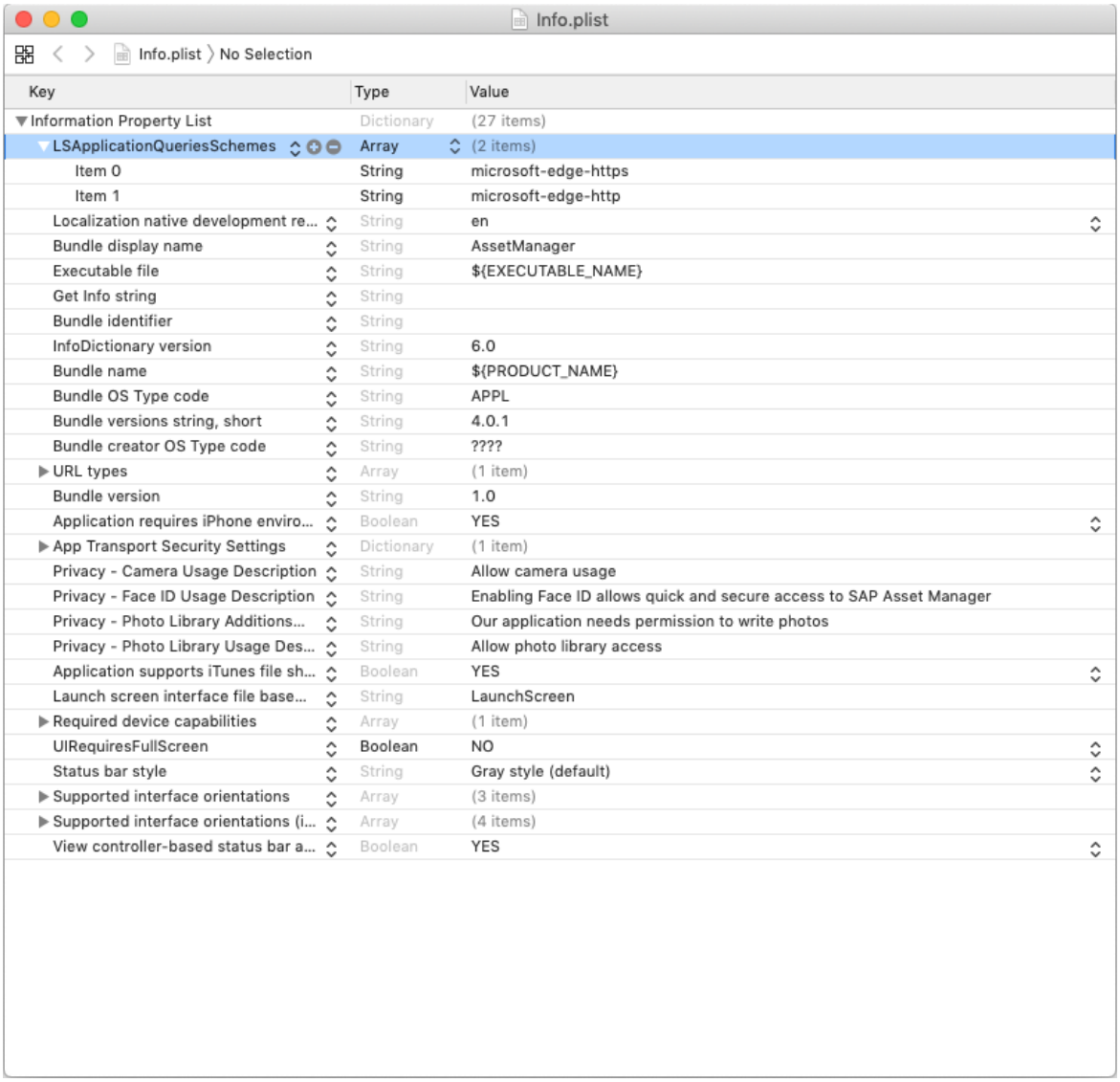
Procedure

1. Locate the `Info.plist` file, located at `${ASSET_MANAGER_ROOT}/app/App_Resources/iOS`, after building the SAP Service and Asset Manager client.



2. Open the `Info.plist` in Xcode, and add a new key named `LSApplicationQueriesSchemes` of type *Array*.
3. Add an entry for each custom URI scheme. Don't include any trailing colons or slashes.

   In the following example, the custom schemes `microsoft-edge-https` and `microsoft-edge-http` are added. If Microsoft Edge is installed on the mobile device, these custom schemes allow the client to open HTTP and HTTPS URLs in Microsoft Edge.



4. Rebuild and rerun the SAP Service and Asset Manager client.

Results

Any custom URL attachments associated with your newly added schemes now open as expected.

5.5.2 Customizing Map Icons

Prerequisites

Build the SAP Service and Asset Manager. See Building the SAP Service and Asset Manager Application Client [page 65] for more information.

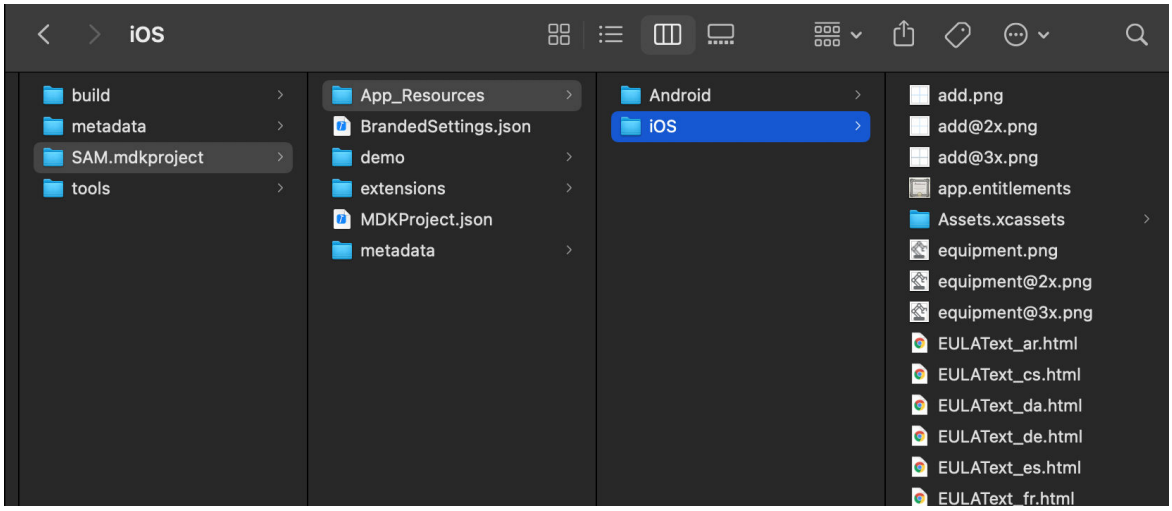
Context

> **!** Restriction
>
> Develop any customization on the app as a separate component in a Mobile Development Kit project. Developing customizations as a component makes it easier to maintain customizations during upgrades, as it isolates custom code. Isolating your custom code eliminates the chance of overwriting when you implement a new release.

Add custom map icons to the *App_Resources* folder located in *mdkproject*. You can then reference the custom map icons directly in the metadata.

Procedure

1. Navigate to the `SAM.mdkproject` folder and add your custom icons to the appropriate Android or iOS folder.



2. Refer to the custom icons directly by name in the SAP Service and Asset Manager metadata. For example, if the custom icons are named `workorder_red.png`, `wo_green.png`, and `wo_orange.png`, set the metadata as follows:

Sample Code

```
"Symbol":{
    "marker":"wo_red.png",
    ...
    "onSelect":{
        "marker":"wo_red_selected.png",
        ...
    }
}
```

3. Set the custom image in a rule to control the image based on business logic:

Sample Code

```
"Symbol":{
    "marker":".SAPAssetManager/Rules/Maps/Icons/WorkOrderIcon.js",
    ...
    "onSelect":{
        "marker":"/SAPAssetManager/Rules/Maps/WorkOrderIconSelected.js",
        ...
}
```

An example of a rule is as follows:

Sample Code

```
export default function WorkOrderIcon(context){
    if(priority=='high,){
        return "wo_red.png";
    }else if(priority=='medium'){
        return "wo_orange.png";
    }else if(priority=='low'){
        return "wo_green.png";
    }
    return "wo.png";
}
```

4. Build the client.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP