Extensibility Guide | PUBLIC

Set Up Authentication for SAP S/4HANA Cloud Extensions

THE BEST RUN SAP

Content

1 Overview

> **i** Note
>
> This sample scenario is for learning purposes only. It is intended to give you an understanding of the various technical aspects related to extending SAP S/4HANA Cloud. The sample scenario may not always be available in a readily consumable state due to the continuous improvements being made in the underlying products or services. If this is the case, appropriate adaptations based on the latest documentation of the respective products or services are required.

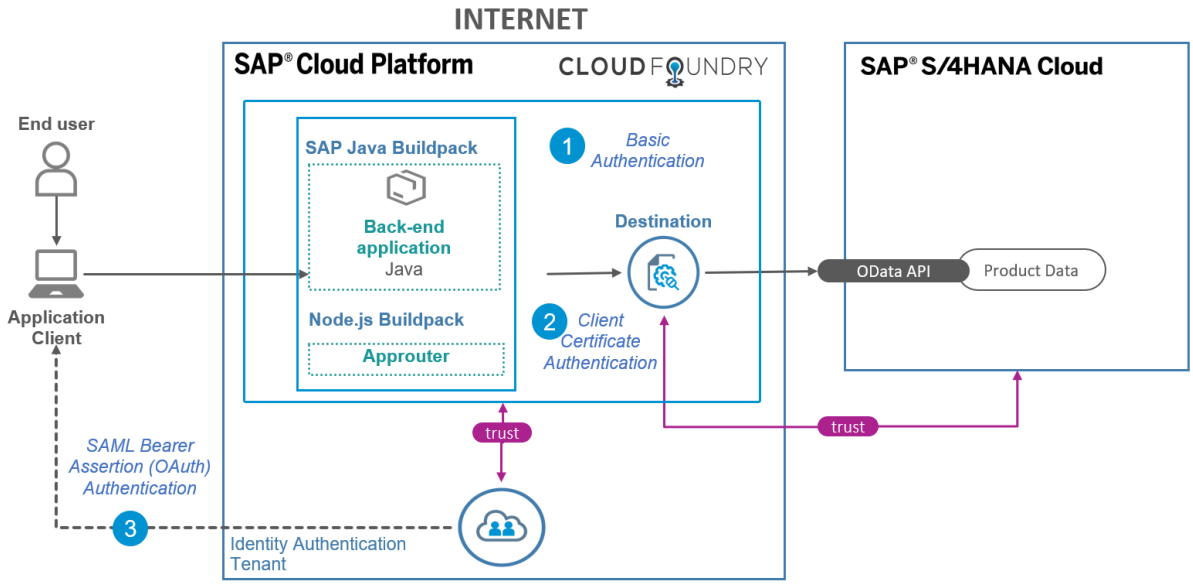
This scenario showcases how to configure the connectivity from your Cloud Foundry account to your SAP S/4HANA Cloud tenant when setting up side-by-side extensibility applications.

Simple approuter and Java applications are used as sample extensions that you can implement in your own landscape. You can check how you can to use different authentication methods when connecting an SAP Business Technology Platform Cloud Foundry account to your SAP S/4HANA Cloud: Basic, Client Certificate, or (OAuth) SAML Bearer Assertion.

This guide describes the typical steps you need to carry out in the SAP S/4HANA Cloud system and in your SAP Business Technology Platform Cloud Foundry account.

1.1 Technical Implementation Steps

This scenario showcases three different authentication techniques that can be implemented to establish a connection between the SAP S/4HANA Cloud system and SAP Business Technology Platform. To demonstrate this, the guide describes the steps that are necessary to implement these authentication methods in simple app router and JAVA applications.



The following table provides you with a brief overview of the major steps in this scenario:

	Step	Details
1	Basic Authentication	• Implementation on SAP S/4HANA Cloud: communication setup (user, system, arrangement) • Implementation on SAP Business Technology Platform: destination maintenance
2	Client Certificate Authentication	• Implementation on SAP S/4HANA Cloud: communication setup (user, system including the client certificate, arrangement) • Implementation on SAP Business Technology Platform: destination maintenance
3	SAML Bearer Assertion (OAuth) Authentication	• Trust configuration setup • Implementation on SAP S/4HANA Cloud: communication setup (user, system including the client certificate, arrangement) • Implementation on SAP Business Technology Platform: destination maintenance

2 Preparation

2.1 Prerequisites

To be able to perform the steps in this document, you need to make sure that the following prerequisites have been met:

Prerequisites	Details
SAP S/4HANA Cloud system	You have access to an SAP S/4HANA Cloud system. Make sure that the following business catalogs are assigned to the roles that your user has. This ensures that your user has the necessary authorizations and can access the respective SAP Fiori launchpad apps. • `SAP_CORE_BC_EXT` (for the key user) • `SAP_CORE_BC_COM` (for communication management)
SAP Business Technology Platform	You have an SAP Business Technology Platform account. For more information on SAP Business Technology Platform accounts, refer to SAP Business Technology Platform Accounts. **i Note** For non-productive/testing purposes, you can use an SAP Business Technology Platform trial account. Find more information about how to get a trial account.
SAP Cloud Identity provider	You need to have admin access to your SAP Cloud Identity provider to walk through the OAuth implementation steps.

3 Basic Authentication

When using the basic authentication method, authentication is achieved through a user name and password. You create a communication user in the SAP S/4HANA Cloud system. When you maintain the HTTP destination in the SAP Business Technology Platform cockpit, you use the communication user to allow communication between SAP S/4HANA Cloud and SAP Business Technology Platform. In the following steps, you can create a communication user, system, and arrangement. Furthermore, after configuring the HTTP destination using basic authentication, you expose an OData API that can be consumed on SAP Business Technology Platform through side-by-side sample apps.

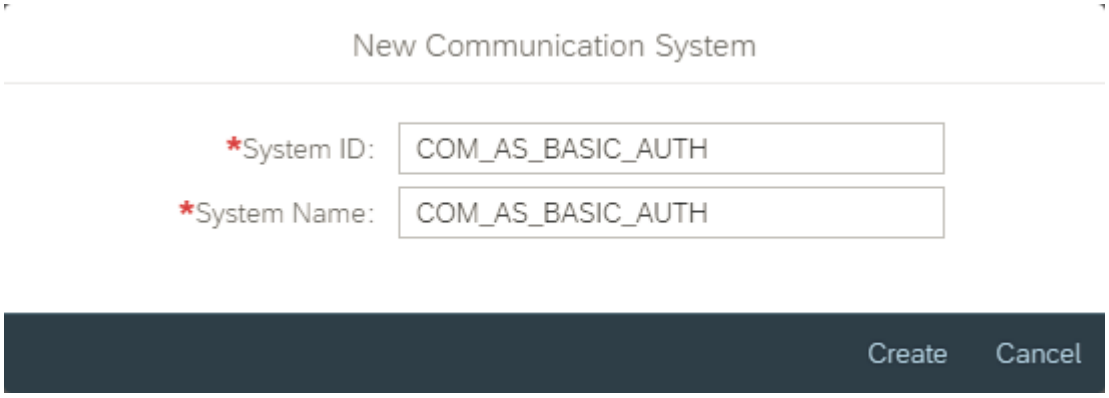
3.1 Implementation Steps in the SAP S/4HANA Cloud System

To allow inbound communication to the SAP S/4HANA Cloud tenant, you need to create a communication arrangement first. The communication arrangement defines which system (communication system) and which user (communication user) can call which APIs (communication scenarios).

3.1.1 Create Communication System and User

Procedure

1. Access the SAP Fiori launchpad.
2. Go to the *Communication Systems* app.
3. Choose *New*.
4. Enter a system ID (such as **`COM_AS_BASIC_AUTH`**). The *System Name* field will be filled automatically.



5. Choose *Create*.
6. On the *Communication System* screen, enter a host name. As this communication system is only used for inbound calls, you don't need to specify a host name. Enter **`localhost`** as value.
7. Make an entry (such as **`My System`**) in the *Logical System* field.
8. In the *Users for Inbound Communication* section, choose *Add* (the + icon) to create a new communication user.
9. In the dialog box, choose *New User*.

> **i Note**
>
> Alternatively, you can create a communication user in the *Maintain Communication Users* app.
>
> If you have already created a user, select the user in the *User Name* field via the value help icon.

10. On the *Create Communication User* screen, enter a user name (such as **`BASIC_AUTH_API_USER`**) and a description.
11. Enter a password.
12. Choose *Create*.
13. On the *Communication System* screen, the new user is inserted automatically in the *User Name* field dialog box. The authentication method is *User Name and Password*.
14. Choose *OK*.
15. In the *Users for Outbound Communication* section, choose *Add* (the + icon) to create a new user.
16. Choose *User Name and Password* and enter the user name and password. Since the outbound user is not needed in this scenario, use **`dummy`** as user name and password.
17. Choose *Save*.
18. Check that the status is *Active*.

3.1.2 Create Communication Arrangement

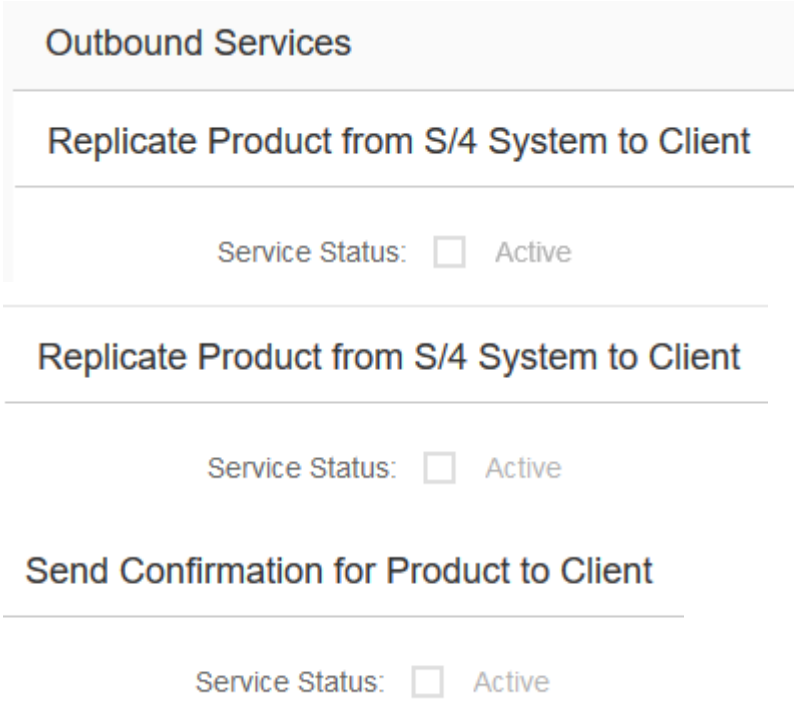
Procedure

1. Access the SAP Fiori launchpad.
2. Go to the *Communication Arrangements* app.
3. To create a new communication arrangement, choose *New* .
4. Create a communication arrangement for the standard product API and select *SAP_COM_0009* from the list.

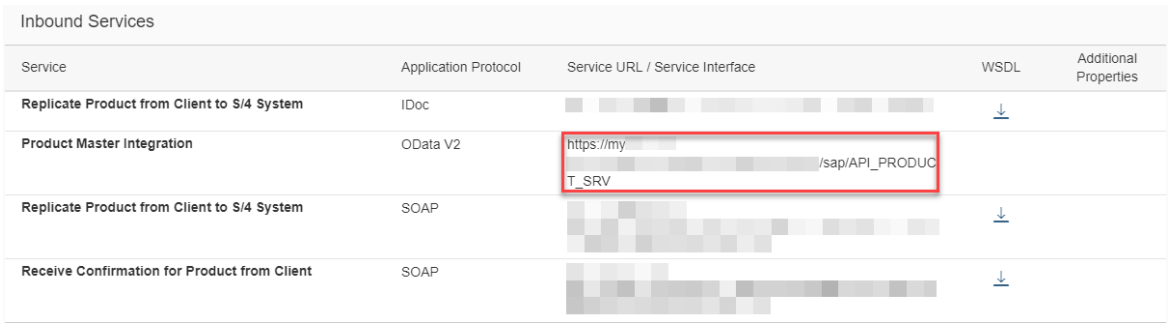
> **i Note**
>
> The `SAP_COM_0009` communication scenario (*Product Integration*) is the basis of this walkthrough example. (The Java application will consume the exposed API for Product Master Data later on. If you use a different scenario, the sample app won't work).

5. To differentiate from the other arrangements of this scenario, adapt the *Arrangement Name* (for example, `SAP_COM_0009_BASIC`).

6. Choose *Create*.
7. In the *Common Data* section, use the value help icon to select the *Communication System* that was created in the Create Communication System and User [page 7] section (for example, `COM_AS_BASIC_AUTH`).
8. The technical user that was created in the previous step is added automatically to the *Inbound Communication* section.
9. As outbound services aren't required, deactivate all of them in the *Outbound Services* section.



10. Although an outbound user isn't required for this scenario, add the dummy outbound user to the *Outbound Communication* section. Otherwise, an error message is displayed when saving.
11. Choose *Save*.
12. Check that the communication arrangements have been activated (*Active* status must be visible).
13. Note down your OData service URL because you'll need it later. You can find the service URLs in the *Inbound Services* section of the communication arrangement.



3.2 Implementation Steps on SAP Business Technology Platform

Destinations are used to allow your application to establish outbound communication to a remote system (in this case, the SAP S/4HANA Cloud system). To create a destination, enter a name, the URL of the SAP S/4HANA Cloud system, the authentication type, and some other configuration data.

3.2.1 Configure a Destination for the Sample Application

> **i Note**
>
> You can you use the destination for the Java application.

Procedure

1. Access SAP Business Technology Platform.
2. Go to *Connectivity* > *Destination*.
3. Choose *New Destination*.
4. Maintain the properties as follows:

Property	Value
Name	`ErpQueryEndpoint` **i Note** The SAP Cloud SDK considers the destination name *ErpQueryEndpoint* as the default name for HTTP-based communication. Don't change this destination name for the Java app.
Type	`HTTP` **i Note** *Type* specifies the communication protocol.

Property	Value
Description	**<description>** **i Note** Enter a meaningful description of the purpose of this destination.
URL	**<the base URL to your SAP S/4HANA Cloud system; note the "-api", https://myXXXXXX-api.s4hana.ondemand.com>** **i Note** *URL* specifies the URL of the target SAP S/4HANA Cloud system. Only provide the protocol, the host name, and the port (no relative path).
Proxy type	**Internet** **i Note** *Proxy type* specifies whether the communication is a direct HTTP call or tunneled via the SAP Cloud connector.
Authentication	**BasicAuthentication** **i Note** *Authentication* specifies which authentication method is used.
User	**BASIC_AUTH_API_USER** **i Note** Enter the name of the inbound communication user and not the generated name. In this case, enter: **BASIC_AUTH_API_USER**.

Property	Value
Password	`<the password of the communication user>` **i Note** *Password* specifies the password of the inbound communication user.

i Note

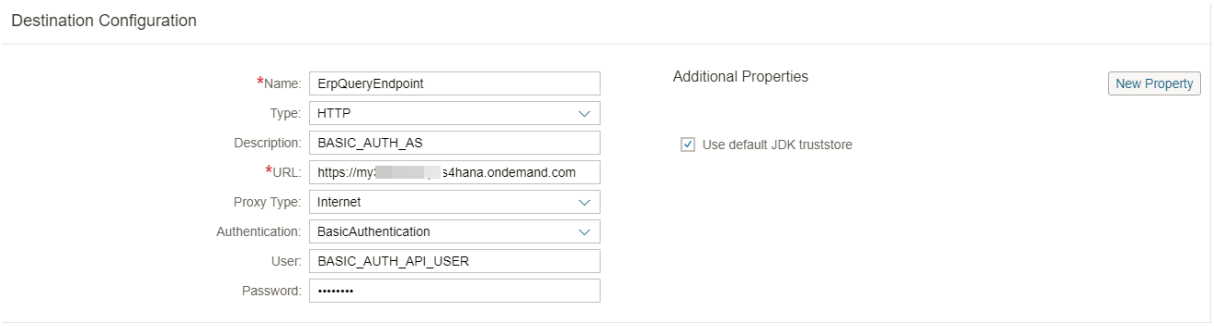
If you're working with your SAP Business Technology Platform trial account, add the following properties to the destination to ensure that the connection to SAP S/4HANA Cloud works.

proxyHost	proxy-trial.od.sap.biz
`proxyPort`	`8080`

5. Choose *Save*.

Result

This is what the destination looks like:



3.3 Test Basic Authentication for the Side-By-Side Application

Please refer to the section Testing the Extension Scenario [page 34].

4 Client Certificate Authentication

In this section, you can configure a client certificate authentication.

To do this, you need a client certificate that is signed by a trusted certificate authority (CA). When you create a communication user in the SAP S/4HANA Cloud system, you upload the public key and add the relevant keystore to the HTTP destination in the SAP Business Technology Platform cockpit.

In our use case, you expose an OData API that can be consumed on SAP Business Technology Platform using side-by-side sample apps. To do this, you configure the HTTP destination by using client certificate authentication. You can use the app router and Java sample apps to check the setup from end to end.

4.1 Prerequisites

Before proceeding with this step, make sure that you have a client certificate signed by a trusted certificate authority (CA). If you don't have one, you can skip the client certificate authentication steps.

For more information, refer to SAP Business Technology Platform: Keys and Certificates.

You can find a list of the trusted CAs in the SAP S/4HANA Cloud system using the *Maintain Certificate Trust List* application. For more information, refer to General Functions for the Key User.

4.2 Implementation Steps in the SAP S/4HANA Cloud System

To allow inbound communication to the SAP S/4HANA tenant, you need to create a communication arrangement first. The communication arrangement defines which system (communication system) and which user can call which APIs (communication scenarios).

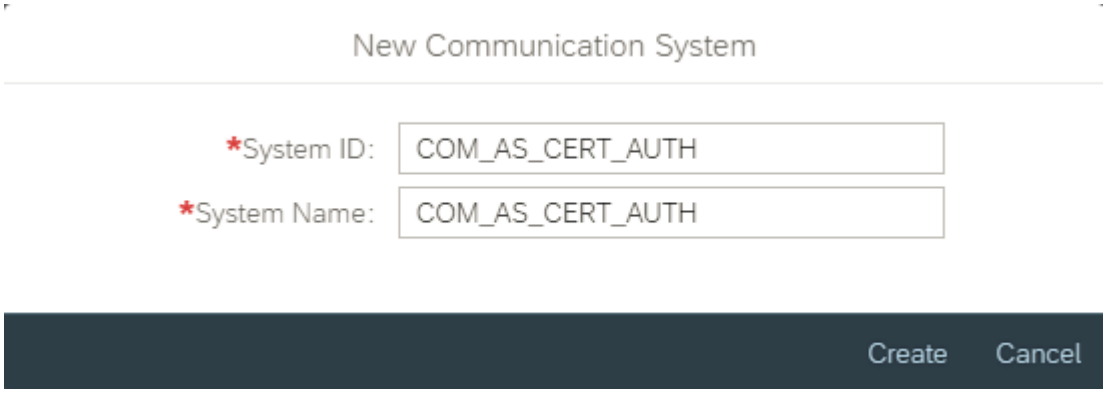
4.2.1 Create Communication System and User

Procedure

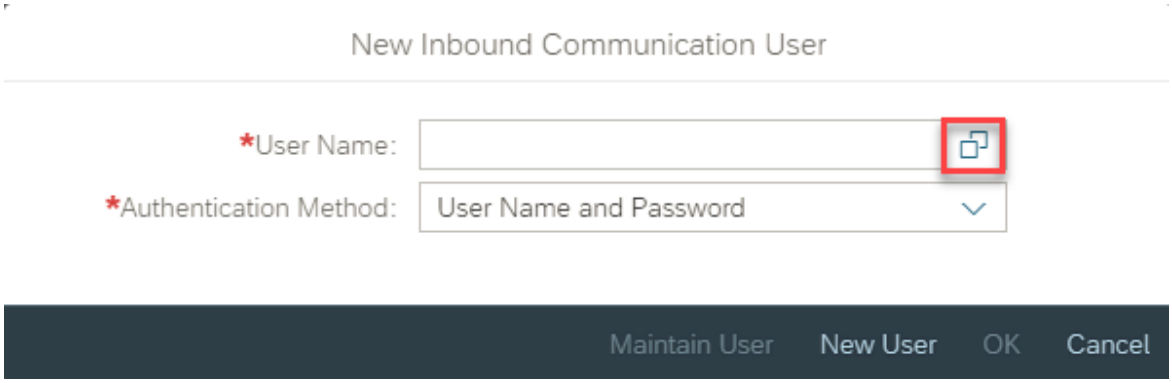
1. Access the SAP Fiori launchpad.
2. Go to the *Maintain Communication Users* app.

   > **i Note**
   >
   > In contrast to the basic authentication setup, creating a communication user in the communication system can lead to technical errors. For this reason, create a communication user before creating a communication system.

3. Choose *New*.
4. Enter a user name (such as **`CERT_AUTH_API_USER`**).
5. Enter a description.
6. Upload the certificate (`*.cer` file) in the *Certificate* section.
   If you have a `*pem` file, you can skip these steps. To export the certificate on a Windows system:
   1. Import the certificate to your Windows system so that you can use certmgr.msc certificates snap-in for Microsoft Management Console to export the public key.
   2. Open the *Run* menu by typing `Win` + `R`.
   3. Type in **`certmgr.msc`** and choose *OK*.
   4. Choose *Personal* > *Certificates*.
   5. Double-click the *Certificate*.
   6. Choose *Details* > *Copy to File*.
   7. Select *Next*.
   8. Choose *Base-64 encoded X.509 (CER)* > *Next*.
   9. Enter a file name and *Save* the certificate.
7. Choose *Create*.
8. Go to the *Communication Systems* app.
9. Choose *New*.
10. Enter a system ID (for example **`COM_AS_CERT_AUTH`**). This entry is adopted for the *System Name* field.



11. Choose *Create*.
12. On the *Communication System* screen, enter a host name. As this communication system is only used for inbound calls, you don't need to specify a host name. Enter **`localhost`** as value.
13. Make an entry (such as **`My System`**) in the *Logical System* field.
14. In the *User for Inbound Communication* section, choose *Add* (the + icon) to create a new communication user.
15. In the dialog box, select the value help icon.



16. Using the *Maintain Communication Users* app, search for the communication user that you created (in this case, *CERT_AUTH_API_USER*).
17. Select the user and choose *OK*. The authentication method is maintained as *SSL Client Certificate*.
18. In the *User for Outbound Communication* section, choose *Add* (the + icon) to create a new user.
19. Choose *User Name and Password* and enter the user name and password. Since the outbound user is a prerequisite but not required in this particular scenario, use **`dummy`** as user name and password.
20. Choose *Save*.
21. Check that the status is *Active*.

4.2.2 Create Communication Arrangement

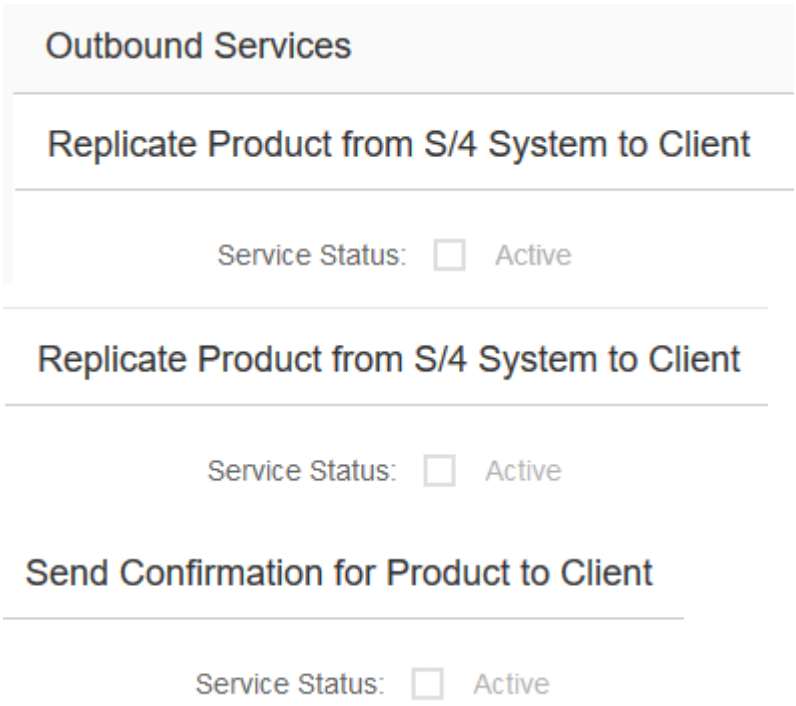
Procedure

1. Access the SAP Fiori launchpad.
2. Go to the *Communication Arrangements* app.
3. Choose *New* to create a new communication arrangement.
4. Create a communication arrangement for the standard *Product API* and select *SAP_COM_0009* from the list.

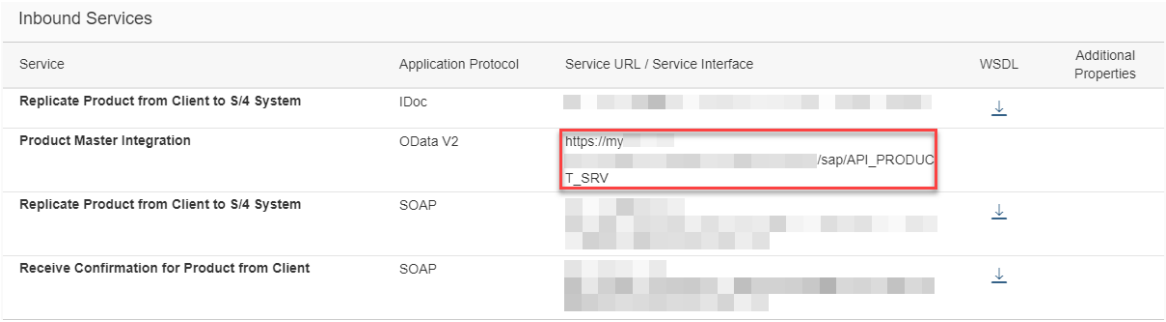
> **i Note**
>
> To keep things simple and showcase the functionality of the certificate-based authentication with the sample Java app, the standard `SAP_COM_0009` scenario is used. If you use a different one, the sample apps don't work.

5. Adapt the *Arrangement Name* if required (for example, **SAP_COM_0009_CERT** to differentiate between the ones of this scenario).
6. Choose *Create*.
7. In the *Common Data* section, select the communication system that was created in the section Create Communication System and User [page 14] (such as *COM_AS_CERT_AUTH*) using the value help icon.
8. The communication user that was created in the previous step is automatically added to the *Inbound Communication* section.
9. If this communication scenario has outbound services, deactivate all of them by clearing the following checkboxes.

> **i Note**
>
> Although an outbound user is not required in this scenario, you must create an outbound user for the `SAP_COM_0009` communication arrangement. Otherwise, an error message is displayed later.



10. Choose *Save*.
11. Check that the communication arrangements have been activated (*Active* status must be visible).
12. Note down your OData service URL for *Product Master Integration* in the *Inbound Services* section because you'll need this URL later. You can find the service URLs in the *Inbound Communication* section of the communication arrangement.



4.3 Implementation Steps on SAP Business Technology Platform

Destinations are used to allow the outbound communication of your application to a remote system (in this case, the SAP S/4HANA Cloud system). To create a destination, enter a name, the URL of the SAP S/4HANA Cloud system, the authentication type, and some other configuration data.

For more information on destination maintenance, refer to Connectivity and Destination APIs

4.3.1 Configure a Destination for the Sample Application

Procedure

1. Access SAP Business Technology Platform.
2. Go to *Connectivity* > *Destinations*.
3. Choose *New Destination*.

4. Maintain the properties as follows:

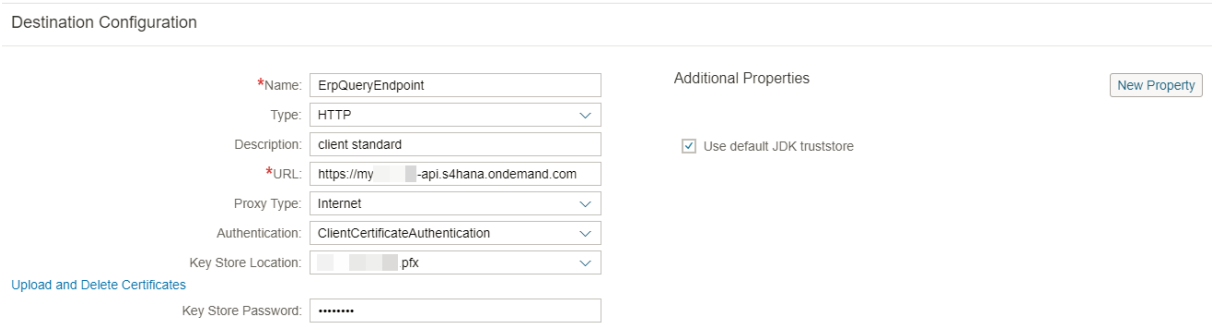
Property	Value
Name	`ErpQueryEndpoint` **i Note** SAP Cloud SDK considers the destination name *ErpQueryEndpoint* as the default name for HTTP-based communication. Don't change this name to test the destination with the Java application.
Type	`HTTP` **i Note** *Type* specifies the communication protocol.
Description	`<description>` **i Note** Enter a meaningful description of the purpose of this destination.
URL	`<the base URL to your SAP S/4HANA Cloud system; note the "-api", https://myXXXXXX-api.s4hana.ondemand.com>` **i Note** *URL* specifies the URL of the target SAP S/4HANA Cloud system. Only provide the protocol, the host name, and the port (no relative path).
Proxy type	`Internet` **i Note** *Proxy type* specifies whether the communication is a direct HTTP call or tunneled via the SAP Cloud Connector.

Property	Value
Authentication	`ClientCertificateAuthentication` **i Note** *Authentication* specifies which authentication method is used.
Keystore Location	Choose *Upload and Delete Certificates*. In the *Certificates* dialog box, choose *Upload Certificate* and select the PFX file you received previously. **i Note** *Keystore* specifies the keystore that has the relevant certificates.
Password	`<the password of the key>` **i Note** *Password* specifies the password that protects the keystore.

5. Choose *Save*.

Result

This is what the destination looks like:



4.4 Test the Certificate-Based Authentication for the Side-By-Side Application

Please refer to section Testing the Extension Scenario [page 34].

5 SAML Bearer Assertion (OAuth2.0) Authentication

SAP Business Technology Platform provides support for applications to use the SAML bearer assertion flow for consuming OAuth-protected resources. Thus, applications don't need to be created to handle some of the complexities of OAuth and can reuse existing identity providers for user data. Users are authenticated by using SAML against the configured trusted identity providers. The SAML assertion is used to request an access token from an OAuth authorization server. This access token is injected automatically in all HTTP requests to the OAuth-protected resources.

5.1 Configure Trust and Federation with UAA Using BTP Identity Authentication Service

Use

Configure the trust configuration of the SAML 2.0 identity provider in your subaccount using the cockpit.

In this specific case, the SAP S/4HANA Cloud system and the SAP Business Technology Platform subaccount must have mutual trust established and use the same identity provider.

By configuring trust in a subaccount and using the same identity provider, you ensure that your SAP S/4HANA Cloud business user can log on to and access the side-by-side application.

Prerequisites

- You have an SAP Business Technology Platform account.
- You have an SAP S/4HANA Cloud system and an SAP Business Technology Platform Identity Authentication service tenant that is already connected to the SAP S/4HANA Cloud system. For more information, refer to SAP Business Technology Platform Identity Authentication Service.
- You have a user with administration authorization for the tenant's administration console for the SAP Business Technology Platform Identity Authentication service.
- A separate "subaccount" for apps is used that is protected by the identity provider (IdP), because the IdP has been configured for a complete "subaccount".

Procedure

Carry out the implementation steps outlined in Establish Trust and Federation with UAA Using SAP Business Technology Platform Identity Authentication Service.

In case you're using another SAML2 Identity Provider, please follow the instructions in Establish Trust and Federation with UAA Using Any SAML Identity Provider.

Result

You've established a trust between the SAP Business Technology Platform Identity Authentication service and SAP Business Technology Platform. Your business user can now log on to and access the side-by-side application.

5.2 Create a Signing Certificate from the SAP Business Technology Platform Account

The SAML assertion sent to SAP S/4HANA Cloud is signed using the private key of the local service provider. To enable SAP S/4HANA Cloud to verify this signature, the relevant certificate is required.

Procedure

1. Access SAP Business Technology Platform.
2. Navigate to *Connectivity* > *Destinations*.
3. Download the certificate by choosing *Download Trust*.
4. Choose *Save*.

> **i Note**
>
> Please remember where you stored the file because you'll need it later.

5.3 Implementation Steps in the SAP S/4HANA Cloud System

To allow inbound communication to the SAP S/4HANA tenant, you need to create a communication arrangement first. The communication arrangement defines which system (communication system) and which user can call which APIs (communication scenarios).

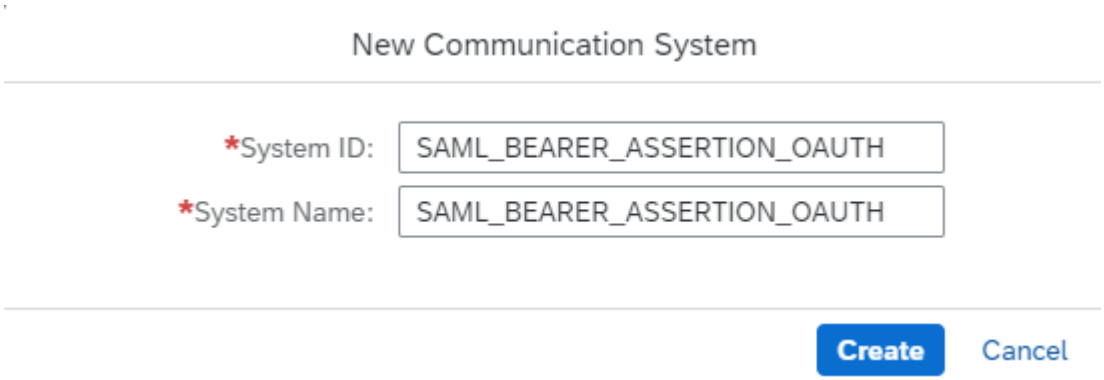
5.3.1 Create Communication System and User

Procedure

1. Access the SAP Fiori launchpad.
2. Go to the *Maintain Communication Users* app.

   > **i Note**
   >
   > In contrast to the basic authentication scenario, creating a communication user in the communication system can lead to technical errors. For this reason, create a communication user before creating a communication system.

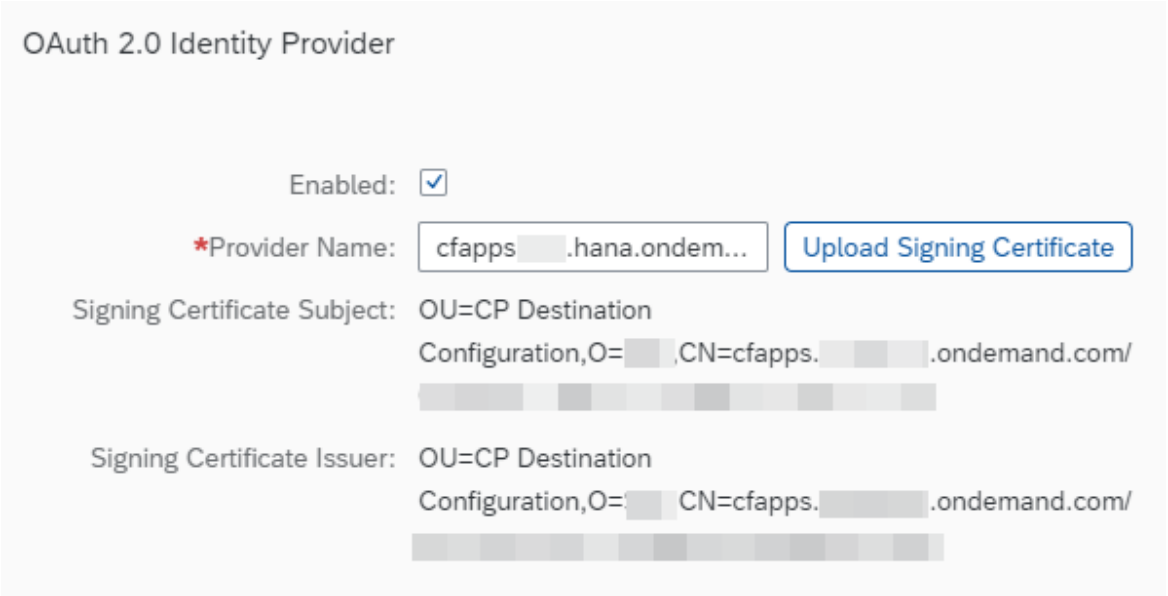
3. Choose *New*.
4. Enter a user name (such as **`SAML_BEARER_ASSERTION`**).
5. Enter a description.
6. Choose *Propose Password* or create one yourself.
7. Go to the *Communication Systems* app.
8. Choose *New*.
9. Enter a system ID (such as **`SAML_BEARER_ASSERTION_OAUTH`**).
   This entry will be adopted for the *System Name* field.



10. Choose *Create*.
11. On the *Communication System* screen, enter a host name. As this communication system is only used for inbound calls, you don't need to specify a host name. Enter **`localhost`** as value.

12. In the *Logical System* field, make an entry (such as **`My System`**) .
13. Under *OAuth 2.0 Identity Provider*, select the checkbox next to the *Enabled* label.
14. Upload the certificate that was created from SAP Business Technology Platform.
15. Enter the provider name that looks like **`<cfapps.<your_region>.hana.ondemandcom/<guid_of_your_subaccount>`**.

> **i** Note
>
> You can use it from the CN value of the signing certificate subject or issuer.



16. In the *User for Inbound Communication* section, choose *Add* (the + icon) to create a new communication user.
17. In the dialog box, use the value help icon to select the user.
18. Search for the communication user you created (in this case, *SAML_BEARER_ASSERTION*) using the *Communication Users* app.
19. Select the user and choose *OK*.
20. In the *User for Outbound Communication* section, choose *Add*(the + icon) to create a new user.
21. Choose *User Name and Password* and enter the user name and password. Since the outbound user is a prerequisite but not required in this particular scenario, use **`dummy`** as user name and password.
22. Choose *Save*.
23. Check that the status is *Active*.

5.3.2 Create Communication Arrangement

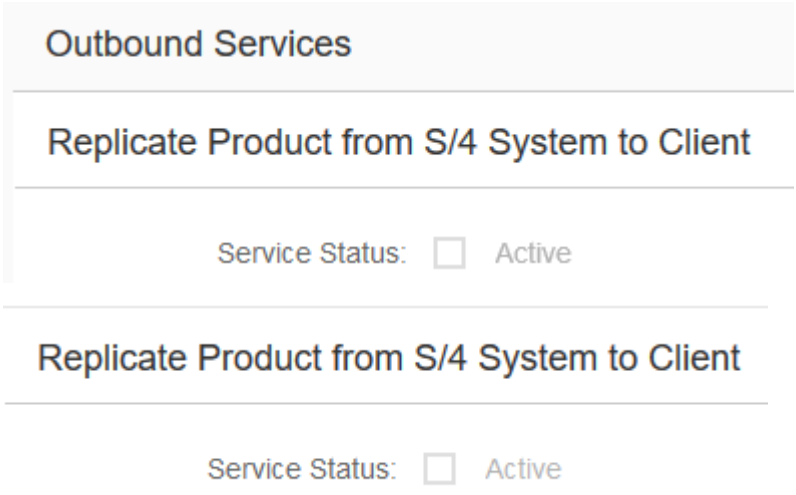
Procedure

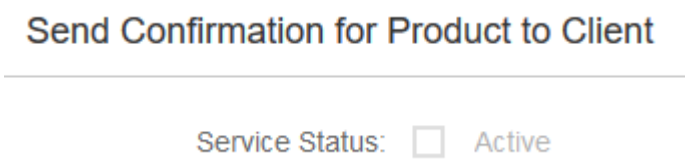
1. Access the SAP Fiori launchpad.
2. Go to the *Communication Arrangements* app.
3. To create a new communication arrangement, choose *New* .
4. Create a communication arrangement for the standard *Product API* and select *SAP_COM_0009* from the list.

> **i Note**
>
> To keep things simple and showcase the functionality of the OAuth-based authentication with approuter and Java application, the standard `SAP_COM_0009` scenario is used.

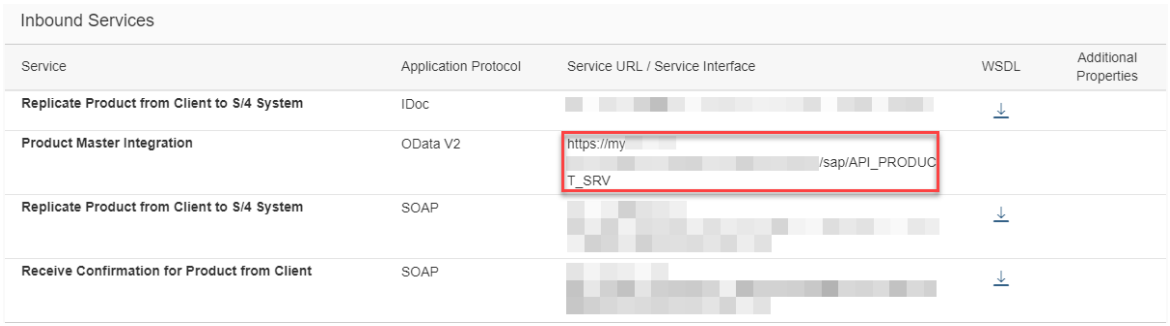
5. Adapt *Arrangement Name* if required.
6. Choose *Create*.
7. In the *Common Data* section, use the value help icon to select the *Communication System* that was created in the section Create Communication System and User [page 23] (such as `COM_AS_OAUTH_AUTH`).
8. The technical user that was created in the previous step is added automatically to the *Inbound Communication* section.
9. Please check whether the *Supported Authentication Method* is “Authentication with OAUTH 2.0” for the inbound communicaction user.
10. If this communication scenario has outbound services, deactivate all of them.

> **i Note**
>
> Although an outbound user is not required in this scenario, you must choose an outbound user for the `SAP_COM_0009` communication arrangement. Otherwise, an error message is displayed later.





11. Choose *Save*.
12. Check that the communication arrangements have been activated (the status *Active* must be visible).
13. Note down your OData service URL because you'll need it later. You can find the service URLs in the *Inbound Communication* section of the communication arrangement.



5.3.3 Maintain Business User in the SAP S/4HANA Cloud System

The principal propagation relies on the equivalence of user master data attributes in both SAP S/4HANA Cloud and SAP Business Technology Platform Identity Authentication service (or another identity provider used in your SAP Business Technology Platform subaccount).

In this section, you can create user master data for the business user and configure which user master data attribute is common.

Procedure

1. Access your SAP S/4HANA Cloud system.
2. Open *Maintain Business User*.
3. Make sure that the following business role is maintained for your user:
   Master Data Specialist - Product Data (`SAP_BR_PRODMASTER_SPECIALIST`)

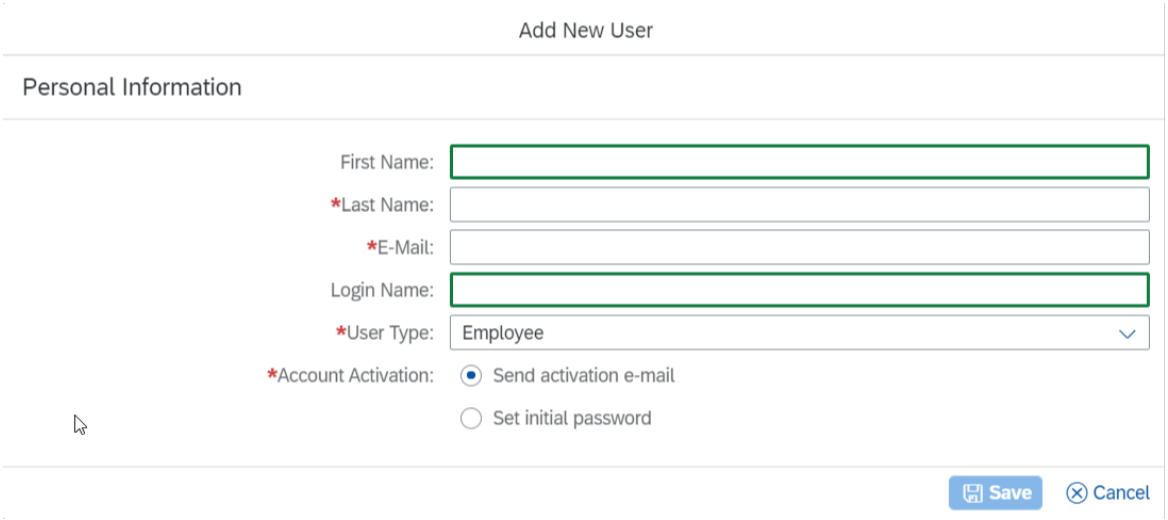
> **i Note**
>
> With this role assignment, the business user can invoke the product master API.
>
> Make sure that the e-mail address is maintained for the user. You can maintain a new employee in the *Maintain Employees* SAP Fiori app.

5.4 Maintain Business User in SAP Business Technology Platform Identity Authentication Tenant

Procedure

> **i Note**
>
> This step is only mandatory if you're working with a different IDP from the one that is used by the SAP S/4HANA Cloud tenant. If the same IDP is used, then the user is already there.

1. Log on to the chosen SAP Business Technology Platform Identity Authentication Tenant.
2. Go to *Users and Authorizations* > *User Management*.
3. To create a new user with the same e-mail address as specified for the business user that was previously created in the SAP S/4HANA Cloud system, choose *Add User*.



4. The user will receive an e-mail with the registration confirmation link from IDP to the e-mail address that was used during the registration. Please follow the confirmation link, otherwise the user will not be able to authenticate.

> **i Note**
>
> For *Account Activation*, select *Set Initial Password*. When you use the newly created user in SAP Identity provider for the first time, you're prompted to reset the password.

5.5 Implementation Steps on SAP Business Technology Platform

Destinations are used for the outbound communication of your application to a remote system (in this case, the SAP S/4HANA Cloud system). To create a destination, enter a name, the URL of the SAP S/4HANA Cloud system, the authentication type, and some other configuration data.

5.5.1 Configure a Destination for the Sample Application

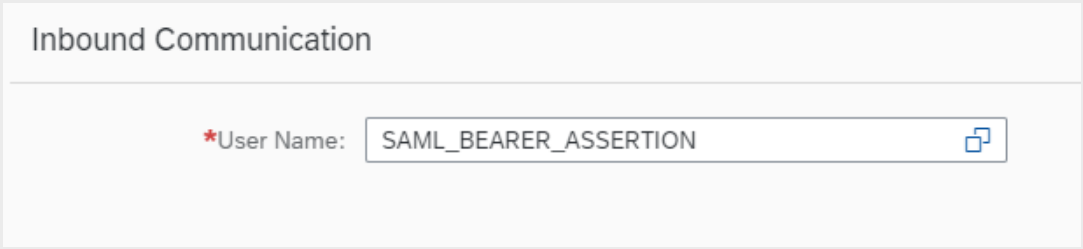
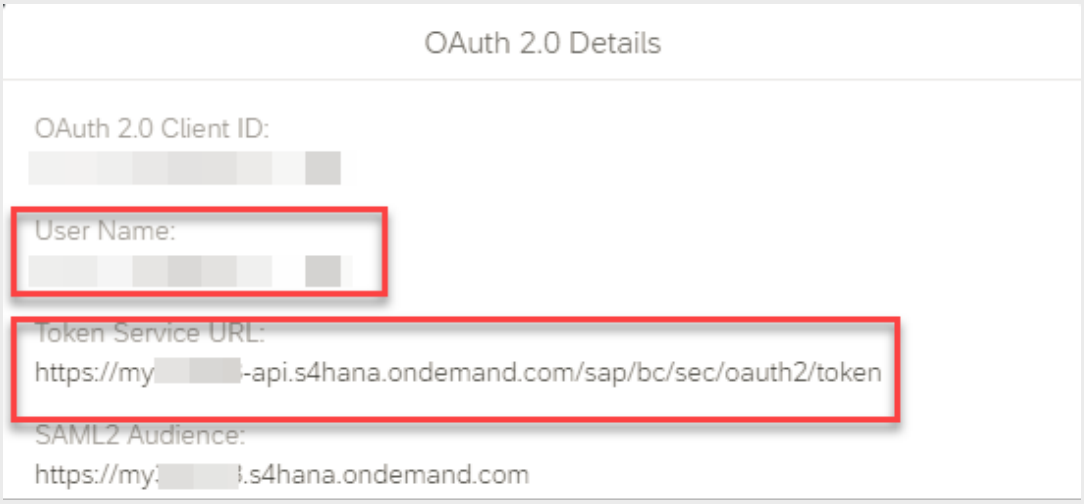
Procedure

1. Access SAP Business Technology Platform.
2. Go to *Connectivity* > *Destinations*.
3. Choose *New Destination*.
4. Maintain the properties as follows:

Property	Value
Name	`ErpQueryEndpoint` **i Note** The SAP Cloud SDK considers the destination name *ErpQueryEndpoint* as the default name for HTTP-based communication. Don't change this name to test the destination with the sample OBJ application.
Type	`HTTP` **i Note** *Type* specifies the communication protocol.

Property	Value
Description	**<the name of your communication arrangements>** **i Note** Enter a meaningful description of the purpose of this destination.
URL	**<the base URL to your SAP S/4HANA Cloud system; note the "-api", https://myXXXXXX-api.s4hana.ondemand.com>** **i Note** *URL* specifies the URL of the target SAP S/4HANA Cloud system. Only provide the protocol, the host name, and the port (no relative path).
Proxy type	**Internet** **i Note** *Proxy type* specifies whether the communication is a direct HTTP call or tunneled via the SAP Cloud Connector.
Authentication	**OAuth2SAMLBearerAssertion** **i Note** *Authentication* specifies which authentication method is used.
Audience	**<the base URL to your SAP S/4HANA Cloud system; https://myXXXXXX.s4hana.ondemand.com>** **i Note** *Audience* specifies the target audience of the issued SAML assertion. Use the host name of your SAP S/4HANA Cloud system.

Property	Value
Client Key	**SAML_BEARER_ASSERTION** **i Note** Enter the name of the inbound communication user and not the generated name. This can be obtained from the SAP S/4HANA Cloud system. Please refer to the screenshot below. *Client Key* represents the OAuth client registered in the SAP S/4HANA Cloud system.
Token Service URL	**<https://myXXXXXX-api.s4hana.ondemand.com/sap/bc/sec/oauth2/token>** **i Note** The *Token Service URL* can be obtained from the SAP S/4HANA Cloud system. Please refer to the screenshot below. *Token Service URL* specifies the URL of the token endpoint (the HTTP endpoint where the SAML assertion is sent to).
Token Service User	**SAML_BEARER_ASSERTION** **i Note** Enter the name of the inbound communication user and not the generated name. This can be obtained from the SAP S/4HANA Cloud system. Please refer to the screenshot below. *Token Service User* specifies the user name for authentication to the token endpoint.
Token Service Password	**<password of the inbound communication user>** **i Note** *Password* specifies the password for authentication to the token endpoint.
System User	**<Leave blank>**

> **Note**
>
> To get more information about the *Client Key*, *Token Service URL*, and *Token Service User*, carry out the following steps:
>
> 1. In the SAP S/4HANA Cloud system, navigate to *Communications Arrangement*.
> 2. Search for the communication arrangement that you created.
> 3. Navigate to *Inbound Communication* and choose *OAUth2.0 Details*.
>
> 
>
>
> 4. The following dialog box is displayed.
>
> 
>
>
> *Client Key* and *Token Service User* are the same as the *User Name*.
> *Token Service URL* is exactly as displayed in the screenshot.

5. Maintain additional properties:

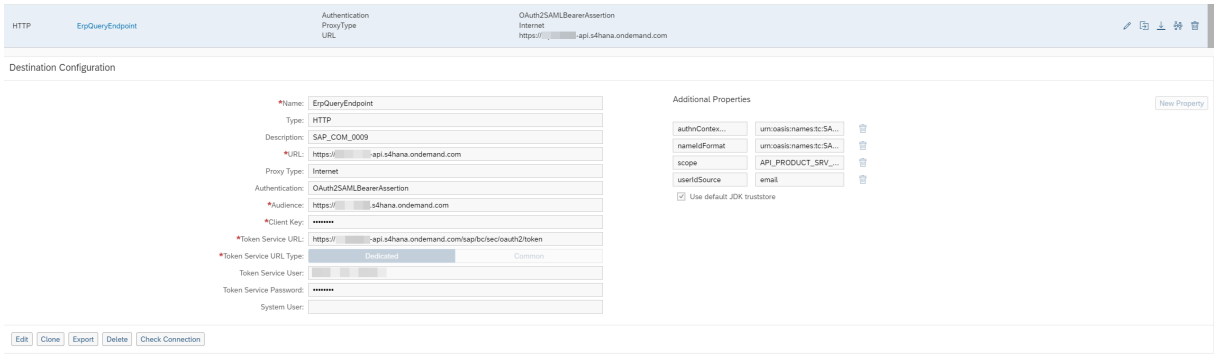
Parameter	Value
authnContextClassRef	`urn:oasis:names:tc:SAML:2.0:ac:classes:X509` > **Note** > *authnContextClassRef* specifies the requested authentication context.

Parameter	Value
nameIdFormat	`urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress` **i Note** *nameIdFormat* specifies which format your principal that was sent to SAP S/4HANA Cloud has.
scope	`API_PRODUCT_SRV_0001` **i Note** *scope* specifies the OAuth scope of the OData service you want to invoke.
userIdSource	`email` **i Note** *userIdSource* specifies which attribute of your SAP Business Technology Platform user master data is propagated as principal to SAP S/4HANA Cloud.

6. Select *Use default JDK truststore*.
7. Choose *Save*.

Result

This is what the destination looks like:



5.6 Test the OAuth SAML Bearer Assertion Authentication for the Side-by-Side Application

Please refer to section Testing the Extension Scenario [page 34]. Pay special attention to Maintain User Roles [page 41] and don't forget to assign the necessary role to your business user in the relevant Identity Provider.

6 Testing the Extension Scenario

In this step, you can download a sample app that showcases the functionality of the scenario that has been set up. The apps are side-by-side applications created with Java and approuter, and show how communication with an SAP S/4HANA Cloud system can be achieved.

Prerequisites

For the Java sample apps to be implemented, make sure that these prerequisites have been met.

- You've installed a JDK 8, which is available on the Oracle Download page . Consider the JAVA_HOME environment variable.
- You've downloaded and installed Maven 3.0, which is available on the Apache Maven Project Download page .
- You've downloaded and installed the Git Command Line Interface, which is available on the Git Download page .

> **i Note**
>
> Make sure that all binaries are maintained in your environment PATH variable. If you're behind a web proxy, configure the proxy settings accordingly (for example, settings.xml for Maven and environment variable HTTPS_PROXY for Git).

6.1 Download the Sample App

Procedure

1. Access the sample app repository on GitHub .
2. Download and extract the ZIP file that you get from the GitHub repository.
   Alternatively, you can clone the repository via the command line.

```
git clone https://github.com/SAP/s4hana-ext-authentication
cd s4hana-ext-authentication
```

6.2 Test Using a Java and Approuter Sample Application

The app is a simple Java app that fetches and displays products from SAP S/4HANA Cloud.

In this step, you're using Maven to package the application (for more information on Maven, refer to Apache Maven Project). The package compiles all sources, runs tests, and bundles all assets in a web application archive file (WAR file). This archive is imported to SAP Business Technology Platform.

6.2.1 Build the Application

1. After you've downloaded and unzipped the repository of the app from GitHub, go the folder of the application.
2. Open the command console.
3. In the command console, switch to the project directory of the Java app (where you downloaded the project).
4. Enter:

   ```
   mvn clean package
   ```

5. After Maven has downloaded all dependencies and successfully compiled the Java source files, you receive a success message:

   ```
   [INFO] BUILD SUCCESS
   ```

   > **i** Note
   >
   > If the build fails, check the troubleshooting section in the readme of the repository.

6. In the application target folder, you can find the WAR file that you created, for example `application/target/authentication-application.war`.

6.2.2 Create an Authorization and Trust (xsuaa) Service Instance

Purpose

You're providing the necessary services to the back-end application.

Prerequisites

- You have a Cloud Foundry space with entitlements to Authorization & Trust Management (xsuaa) and Destination service.
- Your SAP Business Technology Platform user is assigned to the Space Developer and Space Manager role in the Cloud Foundry space where you're going to deploy the application.

Procedure

1. Open the SAP Business Technology Platform cockpit and log on.
2. Navigate to a space where you want to deploy the application.
3. From the navigation area, choose *Services* > *Service Marketplace*.
4. Choose *Authorization & Trust Management* (xsuaa).
5. From the navigation area, choose *Instances*.
6. Choose *New Instance*.
7. Select *Application* from the dropdown list for *Plan* and choose *Next*.
8. To specify parameters, choose *Browse* and select the `xs-security.json` file from the folder where you stored the source code of the application. Choose *Next*.
9. Choose *Next* to skip the *Assign Application* step.
10. Provide a unique *Instance Name* for your service. Note down the instance name because you'll need it later to deploy the application.

Result

You've created an instance for services that are needed for the application. To check this, navigate to your space and choose *Services* > *Service Instances*.

6.2.3 Create a Destination Service Instance

Purpose

You're providing the necessary services to the back-end application.

Prerequisites

1. You've completed the previous steps.

2. You've installed the Cloud Foundry Command Line Interface (CLI) tool.

Download the latest Command Line Interface (CLI) tool from the Cloud Foundry website .

Procedure

1. Launch the command line interface of your operating system from the folder where you stored the source codes of the application. The following commands should be issued in the command line interface of your operating system.

2. To test the Cloud Foundry command line (CLI), type in the following command:

```
cf
```

3. Determine the API endpoint URL of your Cloud Foundry subaccount using Regions and API Endpoints Available for the Cloud Foundry Environment.

4. Using the Cloud Foundry command line (CLI), specify the API endpoint of the Cloud Foundry region where you want to deploy your application: `cf api https://<api-endpoint-of-your-cloud-foundry-region>`

5. Log on to SAP Business Technology Platform using the following command: `cf login`.

6. Enter your e-mail and password.

7. Select your organization and space.

8. Use the following CLI commands to create a service instance: `cf create-service destination <service-plan> <service-name>`

For example:

```
cf create-service destination lite my-destination
```

Result

You've created an instance for services that are needed for the application. To check this, navigate to your space and choose *Services* > *Service Instances*.

6.2.4 Adapt the Manifest File

Purpose

You're providing the necessary services to the back-end application.

Prerequisites

You've completed the previous steps.

Procedure

1. Navigate to the folder where you stored the source code of the project.
2. Open the `manifest.yml file` in a text editor of your choice.
3. Replace the placeholders with real values.

> ⚠ Caution
>
> The real values must not be surrounded by angle brackets (< and >). Make sure that you replace all placeholders (some of them appear twice).

Placeholder	Value
<xsuaa-service-instance>	The name of the Authorization & Trust Management (xsuaa) service instance created in the previous step. This value is mandatory.
<destination-service-instance>	The name of the destination service instance created in the previous step. This value is mandatory.

Placeholder	Value
<backend-service-url>	The URL for the back-end microservice. You need to construct this URL. The URL should be in the correct domain for your region, for example, for Europe (Frankfurt), the valid URL should have format **<your-url-without-dots>.cfapps.eu10.hana.ondemand.com**. For more information, refer to Regions and Hosts. This is a service URL to be used by the approuter, which is not intended to be accessed directly. This value is mandatory.
<main-url>	This is the URL to access the application. This value is mandatory. You need to construct this URL. The URL should be in the correct domain for your region, for example, for Europe (Frankfurt), the valid URL should have the format **<your-url-without-dots>.cfapps.eu10.hana.ondemand.com.**

4. Save the `manifest.yml file`.

6.2.5 Download the Dependencies

Purpose

You're downloading all the necessary dependencies to prepare the application for deployment.

Prerequisites

- You've completed the previous steps.
- *Node.js Package Manager* (NPM) is installed. It can be installed as a part of Node.js .

Procedure

1. Launch `Node.js` command prompt. All the following commands should be issued in it.
2. Run the following command to configure NPM to use the SAP NPM registry: **`npm config set @sap:registry https://npm.sap.com`**
3. Navigate to the folder where you stored the source code of the project.
4. Navigate to the approuter subfolder.
5. Run the following command: **`npm install`**
6. Wait until the operation is completed.

Result

All dependencies are downloaded, and the application is ready to be deployed.

6.2.6 Deploy the Application

Prerequisites

• You've completed the previous steps.

• There's enough memory quota in the target Cloud Foundry space to run the application. The amount of memory needed for each microservice is specified in the relevant memory entry of the matching `manifest.yml file`.

• Cloud Foundry CLI is installed.

Procedure

Refer to section Create a Destination Service Instance [page 36] and make sure that you are targeting the right Cloud Foundry space. Launch the command line and issue the command **`cf target`**, and check the results. In case you've been logged out from Cloud Foundry space, repeat steps 1–7 to log in, and target your Cloud Foundry space again. Then run the following command to deploy the application:

`cf push` and wait until the operation is completed.

6.2.7 Maintain User Roles

Purpose

You're granting legitimate users access to the application.

The application offers one role:

• *Viewer*: Users that are assigned to the Viewer role can access the application

Prerequisites

- All previous steps are completed.
- Your user has administration rights in this subaccount and or global account in SAP BTP.
- The roles defined by your application developers in the application security descriptor are available in the SAP BTP cockpit.
- The users are stored in identity providers that are connected to SAP BTP.

Procedure

1. Open the SAP BTP cockpit.
2. Go to your global account and subaccount.
3. Choose *Security* > *Role Collections*.
4. To create a new role collection, choose + (Create New Role Collection).
5. Enter a new name and description. In this case, enter name as **`Viewer`**.
6. Save your changes.
7. To add roles, go to *Security* > *Role Collections* and choose the role collection *Viewer*.
8. Go to the *Roles* section and choose *Edit*.
9. To add a role to the role collection, choose the input field. The role selection screen opens.
10. To display the roles that are available, use the dropdown list or the F4 function key under *Role Name*. Choose the role you want to add.
    In this case, select your Authorization & Trust Management (xsuaa) service instance from the dropdown for *Application Identifier*. Select the row where *RoleName* is *Viewer* and *RoleTemplate* is also *Viewer*.
11. Choose *Add*.
12. Save your changes.
13. To add users to role collections, go to *Security* > *Role Collections* and choose the *Viewer* role collection.
14. Go to the *Users* section and choose *Edit*.
15. Enter the user ID of the user you want to assign to the *Viewer* role collection and choose the user.

16. Save your changes.

Result

The legitimate users are now able to access the application after it has been started.

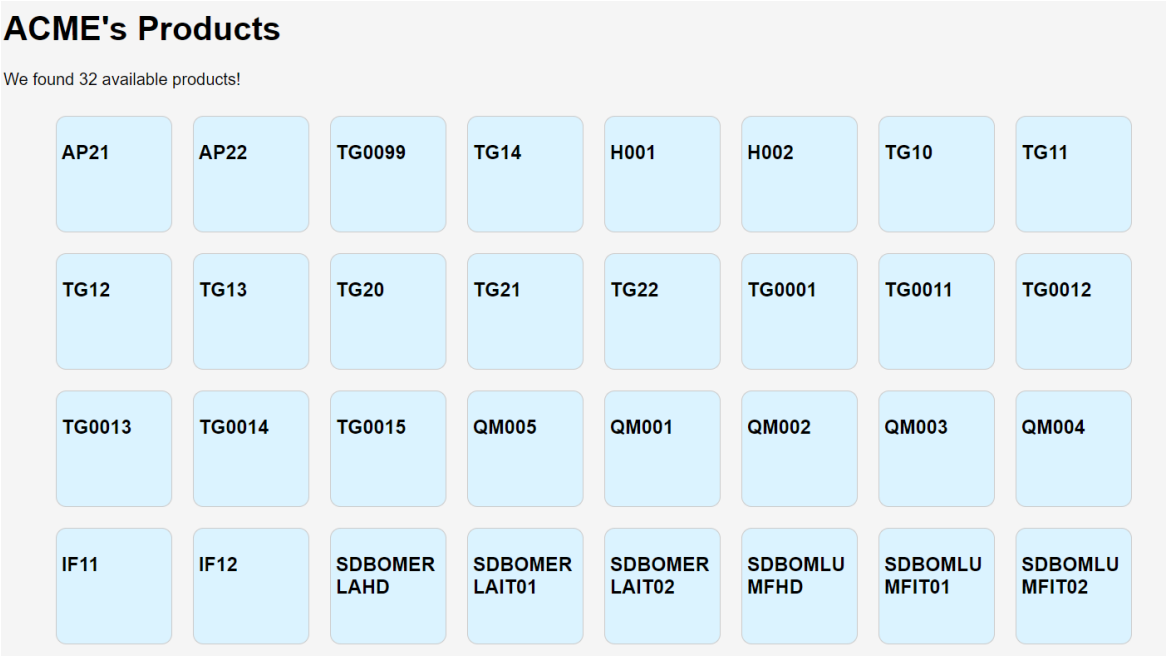
6.2.8 Run and Test Application

Prerequisites

To work your way through the app, you must have appropriate business data in your system.

Procedure

1. Access the *Sample Authentication* app in your SAP Business Technology Platform account. You can find the application URL under *Applications* > *App Router* > *Application Routes* link .
2. On the overview page of the application, you can see a list of products that are maintained in your SAP S/4HANA Cloud system.



> **Note**
>
> These products are read live from the SAP S/4HANA Cloud system. The sample code only shows products in the *L001* product group. This filter is hard coded and can be changed in the `application.properties` file.

7 Appendix

7.1 Issues

Please note that SAP does not offer any official support for the sample code (see the *SAP SAMPLE CODE LICENSE AGREEMENT* on GitHub). However, feel free to use the *Issues* section on GitHub if you have any problems. We recommend that you browse through the known issues section before creating a new issue.

7.2 Document History

Revision	Date	Change
1.0	2018-08-15	Document created.
1.1	2018-08-28	*Trust Configuration between SAP Cloud Platform and SAP Cloud Identity Provider* section and subsections replaced by Configure Trust and Federation with UAA Using BTP Identity Authentication Service [page 21] section.
2.0	2018-09-20	*SAP S/4HANA Cloud Calls an External Service Using Client Certificate Authentication* section and subsections added.
2.1	2019-04-01	Download the Sample App [page 34] updated.
2.2	2019-06-03	Prerequisites [page 6] updated.
2.3	2019-09-13	Issues [page 44] section added.
3.0	2020-02-11	Document updated.
3.1	2021-05-27	Maintain User Roles [page 41] updated.

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.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

www.sap.com/contactsap

THE BEST RUN SAP