QuantityWare

QAPI Connection Guide

QuantityCloud Phase 1

Version History

Version	Date	Description
00	2025-04-16	Initial Version
01	2025-05-06	Added details of the setup of and authorizations for the RFC user in the SAP system
02	2025-05-20	Clarified HTTP method for connection in 9.3 Corrected XML example in 9.4 Added PHP CURL example in 9.5
03	2025-08-20	Edited for Knowledge Base
04	2025-09-19	Amended for iFlow and Destination naming conventions.

Contents

1. Introduction

QAPI provides access to the QuantityWare BCS quantity conversion functionality to systems external to SAP, via an SAP BTP API.

It is the first phase of the wider QuantityCloud project, and more information can be found at https://www.quantityware.com/faqs/qapi.

The QAPI functions reside on the SAP Oil, Gas, and Energy system in which QuantityWare BCS has been installed on either an S/4 HANA or ECC system. This system is owned and run by the customer and contains the required configurations.

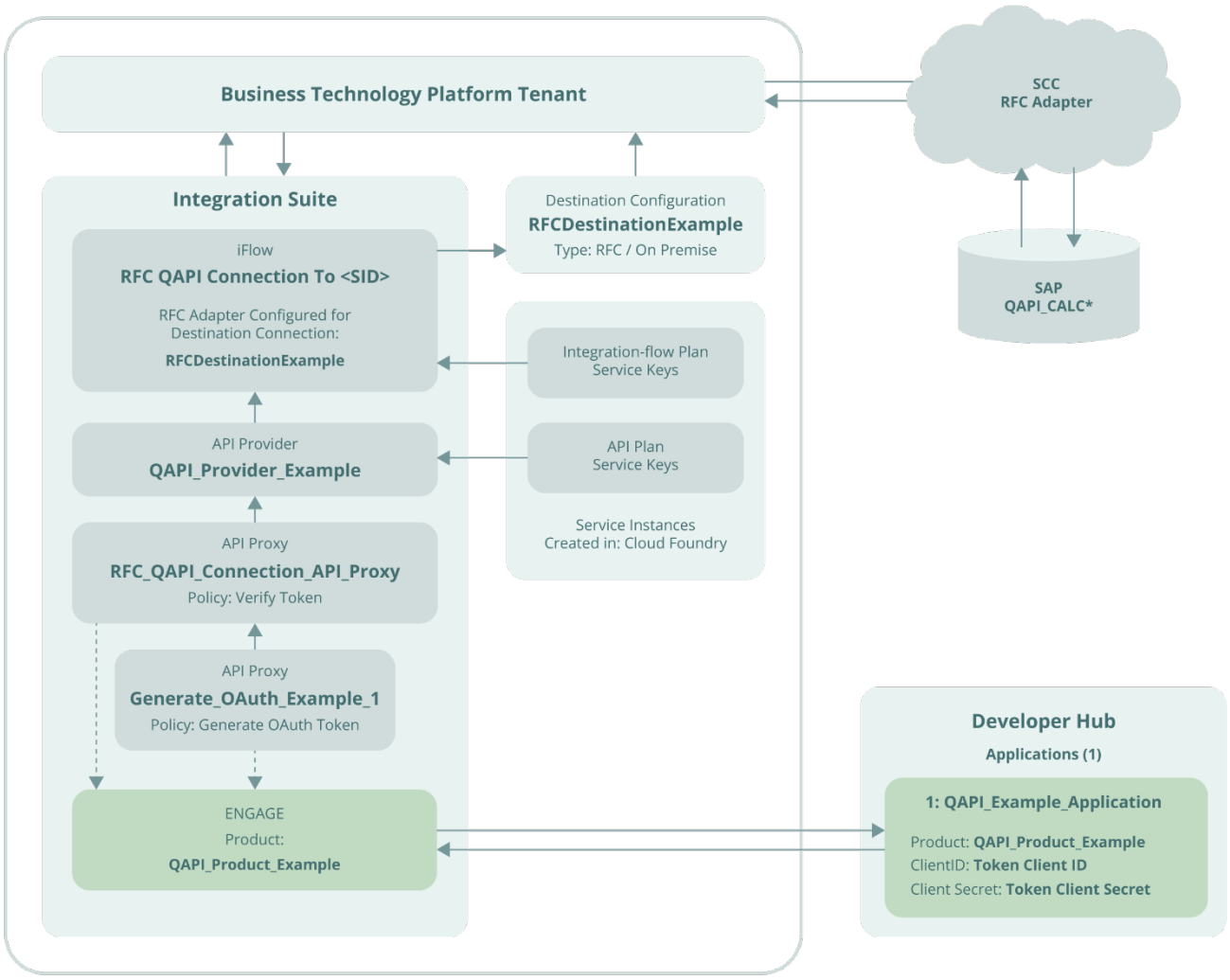
We need to expose the functions to the web, which requires configuration of several elements within SAP Cloud Connector, the BTP Cockpit, and Integration Suite.

This document describes an example configuration set up, which we have used to successfully interact with the QAPI functions with external systems via the web.

2. Configuration Overview

Below we describe the end-result of completing all the configuration in this document. Your exact configuration may vary.

2.1. Configuration Diagram



2.2. Configuration Components

2.2.1. Integration Flow (iFlow)

The iFlow (**RFC QAPI Connection To <SID>**), is nothing more than a sender / receiver configuration, which will incorporate an RFC Adaptor to return a set of data from the On-Premise SAP RFC ABAP Modules / Function.

2.2.2. API Provider

The Api provider (**QAPI_Provider_Example**) will be created to securely expose the iFlows RESTful API so that it can be consumed externally.

2.2.3. Service Keys

The Service Keys are **Service Instances** that shall be created in Cloud Foundry, which will supply ClientID, Secret and a Return URL for Authentication.

- The **integration-flow Plan** service keys will be used to service the iFlow (**RFC QAPI Connection To <SID>**).
- The **API Plan** service keys will be used to service the API Provider (**QAPI_Provider_Example**)

2.2.4. API Proxies

The API Proxy (**RFC_QAPI_Connection_API_Proxy**) will be created using the API Provider (**QAPI_Provider_Example**), which in turn will expose the iFlow (**RFC QAPI Connection To <SID>**).

The API Proxies will be configured with a “ProxyEndpoint/PreFlow - Policy” that expects a Valid Token to Verify before it will pass the request to the EndPoint URL (i.e. call the iFlow).

A common API Proxy will be created (**Generate_OAuth_Example_1**). This Proxy will be configured with a “ProxyEndpoint/PreFlow - Policy” that Generates the Token required by the other two proxies.

2.3. API Packaging Procedure

A Product will be first created, where we group all the API Proxies required.

In our example, we created **QAPI_Product_Example**

This Product will include two API Proxies:

- **RFC_QAPI_Connection_API_Proxy**
- **Generate_OAuth_Example_1**

In Developer Hub, we will then create an Application, **QAPI_Example_Application**, based on the desired Product (**QAPI_Product_Example**), which in turn generates our Application Key and Secret for that Product.

2.4. Testing the Connection

In Postman, we will:

- Issue a call with the correct XML payload to the URL from API Proxy **QAPI_API_Example**
- Authenticate our request using the credentials generated in the Application and the Return URL in the API Proxy **Generate_OAuth_Example_1**
- Use the valid Token generated to authenticate our request with API Proxy **QAPI_API_Example**

3. RFC User Set Up

We first need to create the RFC user that will be used to access the QAPI functions from BTP.

3.1. Prerequisites

- The user is created on the destination SAP system and client
- Relevant authorizations have been added to the user

3.2. Create User

Enter transaction **SU01** in the destination SAP system's target client.

Create the user with the "**User Type**" of "**Communications Data**".

3.3. Grant Authorizations

Add the following authorizations to the user:

Authorization Object	RFC_TYPE	RFC_NAME	ACTVT
S_RFC	Function group	SYST	Execute
S_RFC	Function module	RFCPING	Execute
S_RFC	Function group	RFC_METADATA	Execute
S_RFC	Function module	RFC_METADATA_GET	Execute
S_RFC	Function group	/QTYW/QAPI	Execute
S_RFC	Function module	/QTYW/QAPI_CALCUATE, /QTYW/QAPI_CALC_CONTEXT_GET	Execute

Once complete, the authorizations should match those shown on the following page:

Authorization Object S_RFC	Manual			Authorization Check for RFC Access
Authorization T-SJ22000500	Manual			Authorization Check for RFC Access
RFC_TYPE	Manual		Function group	Type of RFC object to which access is to be allowed
RFC_NAME	Manual		SYST	Name (Whitelist) of RFC object to which access is allowed
ACTVT	Manual		Execute	Activity
Authorization T-SJ22000501	Manual			Authorization Check for RFC Access
RFC_TYPE	Manual		Function Module	Type of RFC object to which access is to be allowed
RFC_NAME	Manual		RFCPING	Name (Whitelist) of RFC object to which access is allowed
ACTVT	Manual		Execute	Activity
Authorization T-SJ22000502	Manual			Authorization Check for RFC Access
RFC_TYPE	Manual		Function group	Type of RFC object to which access is to be allowed
RFC_NAME	Manual		RFC_METADATA	Name (Whitelist) of RFC object to which access is allowed
ACTVT	Manual		Execute	Activity
Authorization T-SJ22000503	Manual			Authorization Check for RFC Access
RFC_TYPE	Manual		Function Module	Type of RFC object to which access is to be allowed
RFC_NAME	Manual		RFC_METADATA_GET	Name (Whitelist) of RFC object to which access is allowed
ACTVT	Manual		Execute	Activity
Authorization T-SJ22000504	Manual			Authorization Check for RFC Access
RFC_TYPE	Manual		Function group	Type of RFC object to which access is to be allowed
RFC_NAME	Manual		/QTYW/QAPI	Name (Whitelist) of RFC object to which access is allowed
ACTVT	Manual		Execute	Activity
Authorization T-SJ22000505	Manual			Authorization Check for RFC Access
RFC_TYPE	Manual		Function Module	Type of RFC object to which access is to be allowed
RFC_NAME	Manual		/QTYW/QAPI_CALCULATE, /QTYW/QAPI_CALC_CONTEXT_GET	Name (Whitelist) of RFC object to which access is allowed
ACTVT	Manual		Execute	Activity

3.4. Share Details

The username and password should be shared with the colleague performing the create the destination in BTP steps later in the setup process.

4. SAP Cloud Connector (SCC)

We next need to expose the on-premise SAP Server, and the QAPI RFC functions, to SAP BTP via the SAP Cloud Connector.

SAP Cloud Connector acts as a **secure tunnel (reverse proxy)** between your local network and the cloud, without exposing your internal systems directly to the internet. The following instructions will detail the required setup.

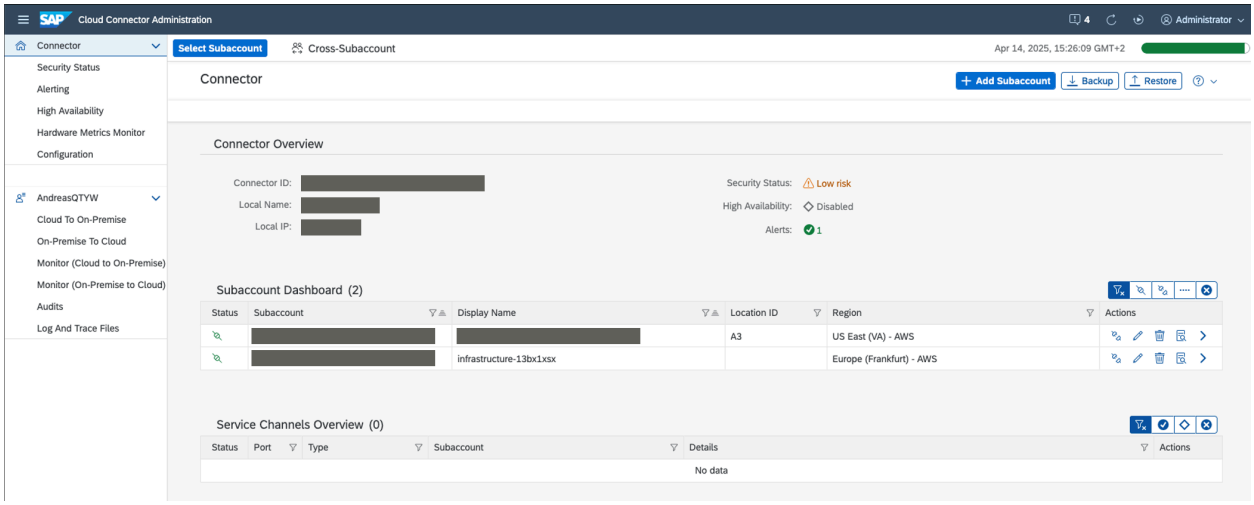
4.1. Prerequisites

- SAP Cloud Connector installed and connected to an SAP BTP subaccount.

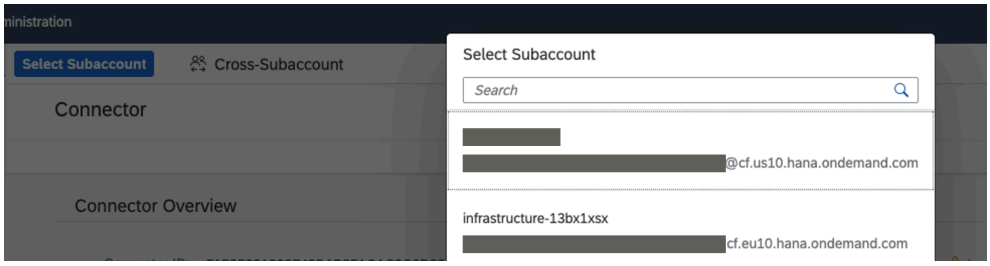
4.2. Create Mapping to Internal System

Firstly, login to your SAP Cloud connector.

You will see the Connector Overview, and a list of all configured subaccounts:

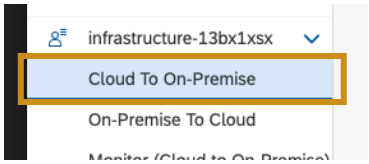


Click the "**Select Subaccount**" button in the top bar, and click the subaccount from the "Select Subaccount" menu:



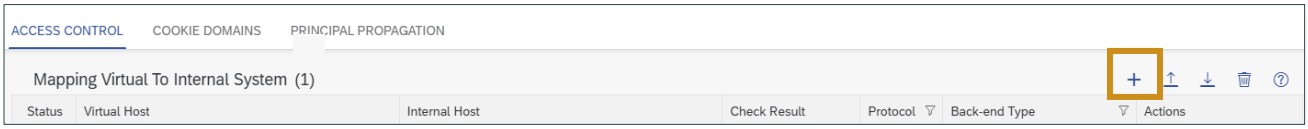
The Subaccount overview will be shown.

From the navigation menu, under the Subaccount name, click "**Cloud to On-Premise**".

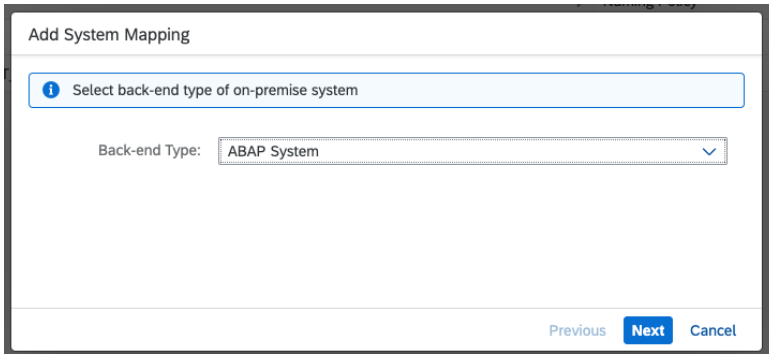


The Cloud To On-Premise overview will display.

In the "ACCESS CONTROL" tab, to the right of the "Mapping Virtual To Internal System" section, click the "**+**" button:



The Add System Mapping dialog will display:

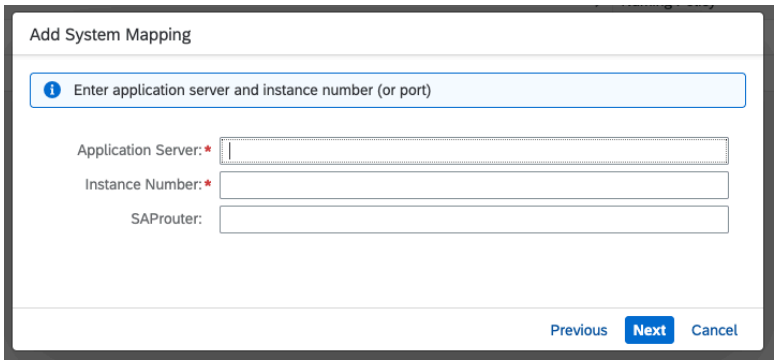


Leave the "**Back-end Type**" as "**ABAP System**" and click "**Next**".

Change the "**Protocol**" to "**RFC**" and click "**Next**".

Change "**Connection Type**" to "**Without load balancing**" (unless you are specifically intending to use load balancing) and click "**Next**".

The next panel will display:

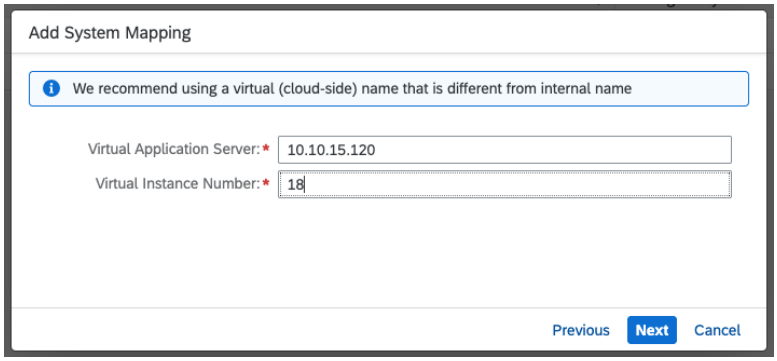


In the "**Application Server**" field, enter internal host name (or IP address) of the target SAP server.

In the "**Instance Number**" field, enter the target SAP instance on the target SAP server.

Click "**Next**".

The virtual application server panel will display, with the fields automatically filled to match the physical application server details entered in the previous step:

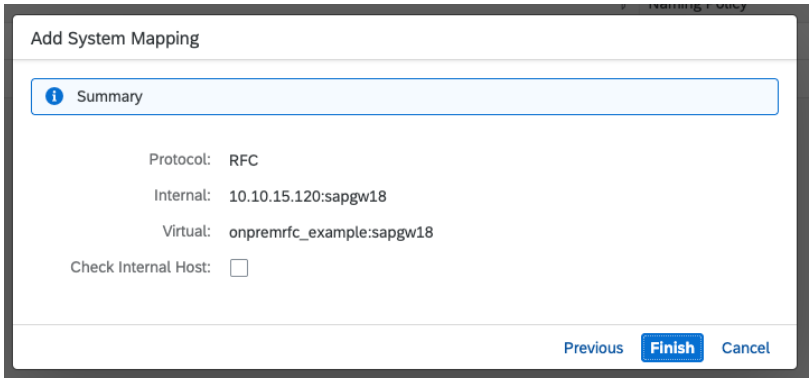


It is recommended to change the "**Virtual Application Server**" to something other than the internal name of the SAP server.

Click "**Next**".

Optionally, enter a description in the "**Description**" field, and click "**Next**".

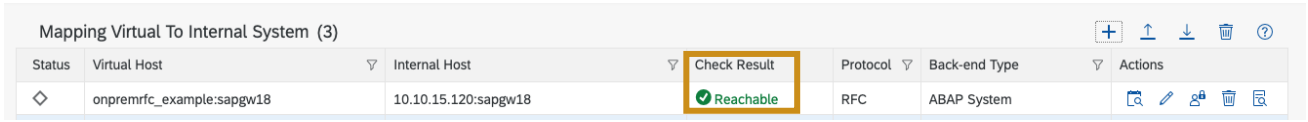
The Summary panel will display:



Select the "**Check Internal Host**" check box to perform a connection check immediately on finishing.

Click "**Finish**" to create the connection.

The system will be displayed in the "Mapping Virtual To Internal System" section, which should show as "**Reachable**":



If so, your mapping has been created successfully.

Configuration Example:

Property	Value	Note
Back-end Type	**ABAP System**	
Protocol	**RFC**	
Application Server	**10.10.15.120**	Your internal host name
Instance Number	**sapgw18**	"sapgw" is placed in front of the instance number you enter
Virtual Application Server	**onpremrfc_example**	Your externally accessible virtual host name
Virtual Instance Number	**sapgw18**	"sapgw" is placed in front of the virtual instance number you enter
System ID	**PRD**	Your system ID (optional)

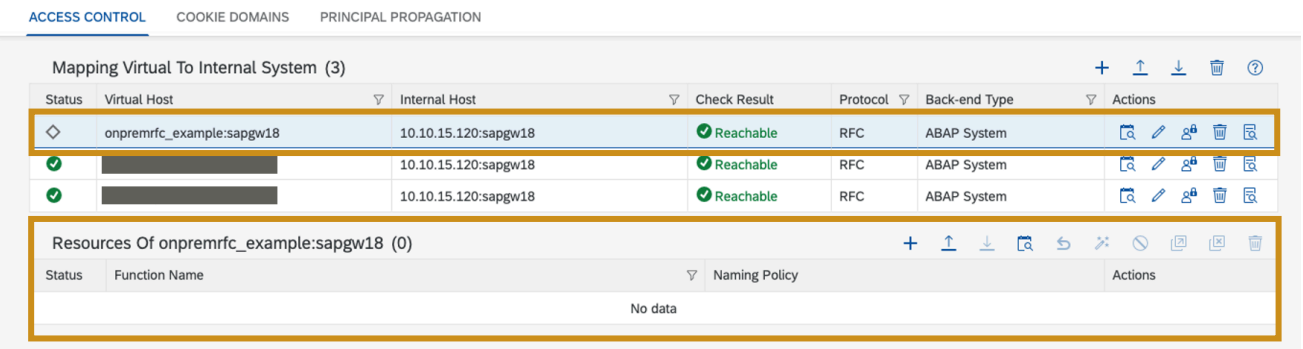
4.3. Make RFC Functions Available

Below we describe how to make the following required QAPI RFC functions available:

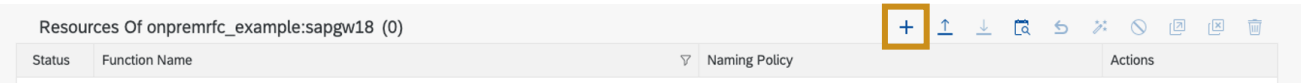
- **/QTYW/QAPI_CALCULATE**
- **/QTYW/QAPI_CALC_CONTEXT_GET**

In the "ACCESS CONTROL" tab, in the "Mapping Virtual To Internal System" section, click your created mapping to view its resources.

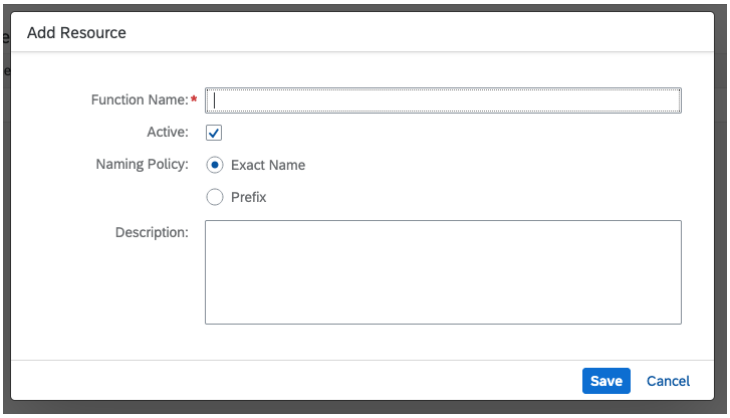
There will be no resources shown yet:



To the right of the "Resources of [target]" section, click the "**+**" button:



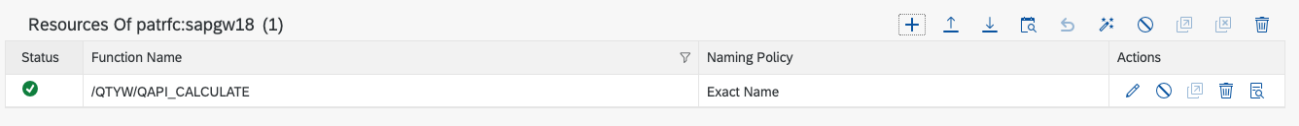
The Add Resource dialog will show:



In the "**Function Name**" field, enter: "**/QTYW/QAPI_CALCULATE**" (without quotes).

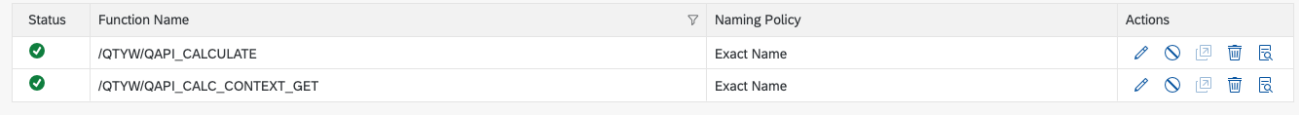
Leave the other settings as default and click "**Save**".

The RFC function will show in the "Resources of [target]" section:

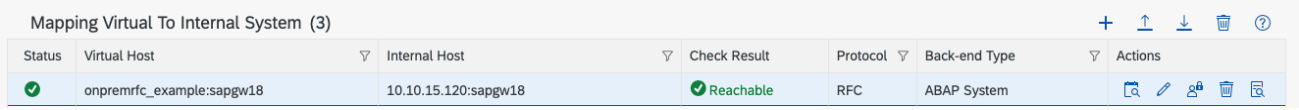


Repeat the above steps for "**/QTYW/QAPI_CALC_CONTEXT_GET**".

Both functions will now show in the "Resources of [target]" section:



You will notice that the status of your mapping in the "Mapping Virtual To Internal System" section will show a green status tick, which indicates it has mappings defined:



5. Business Technology Platform (BTP)

Here, we create the Destination route to our SAP Server via the Cloud Connector, directly in the subaccount, as well as build the necessary APIs (and service keys) required to access the QAPI functions using the Integration Suite.

SAP Business Technology Platform (BTP) is SAP's unified cloud environment where you can integrate, extend, and build applications securely. In this guide, BTP is used as the platform for creating and running the iFlow.

5.1. Prerequisites

- You have an active SAP BTP subscription
- The following services are enabled on your SAP BTP subaccount:
    - Integration Suite
    - Cloud Foundry (with Space created)
    - SAP Process Integration Runtime (with authorizations configured)

SAP BTP Documentation

If you do not currently have an SAP BTP subscription, SAP provide official documentation to guide you through the process of setting up a BTP trial account and the Integration Suite.

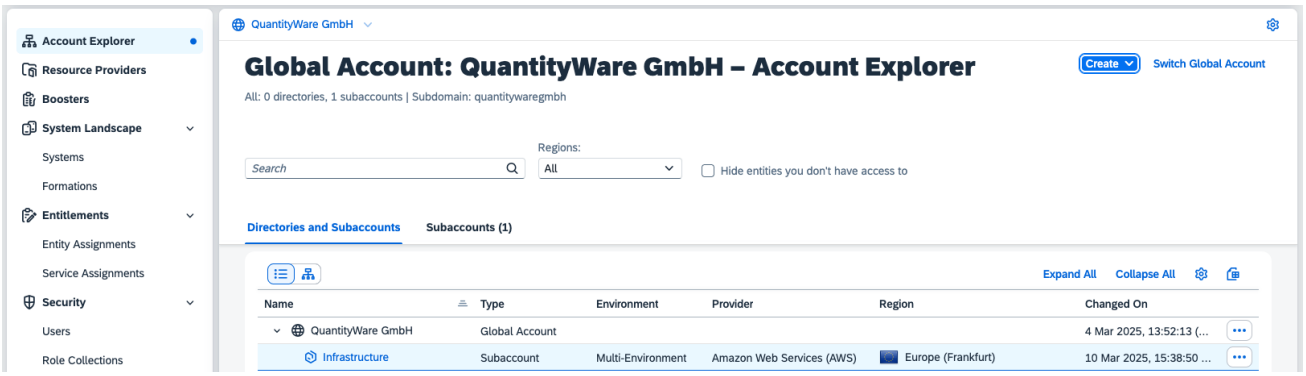
We have included links to the documentation below. Please note that these are external SAP resources and were not created or maintained by QuantityWare:

- Get an Account on SAP BTP Trial | SAP Tutorials
- Preparing Your SAP Integration Suite Tenant | SAP Tutorials

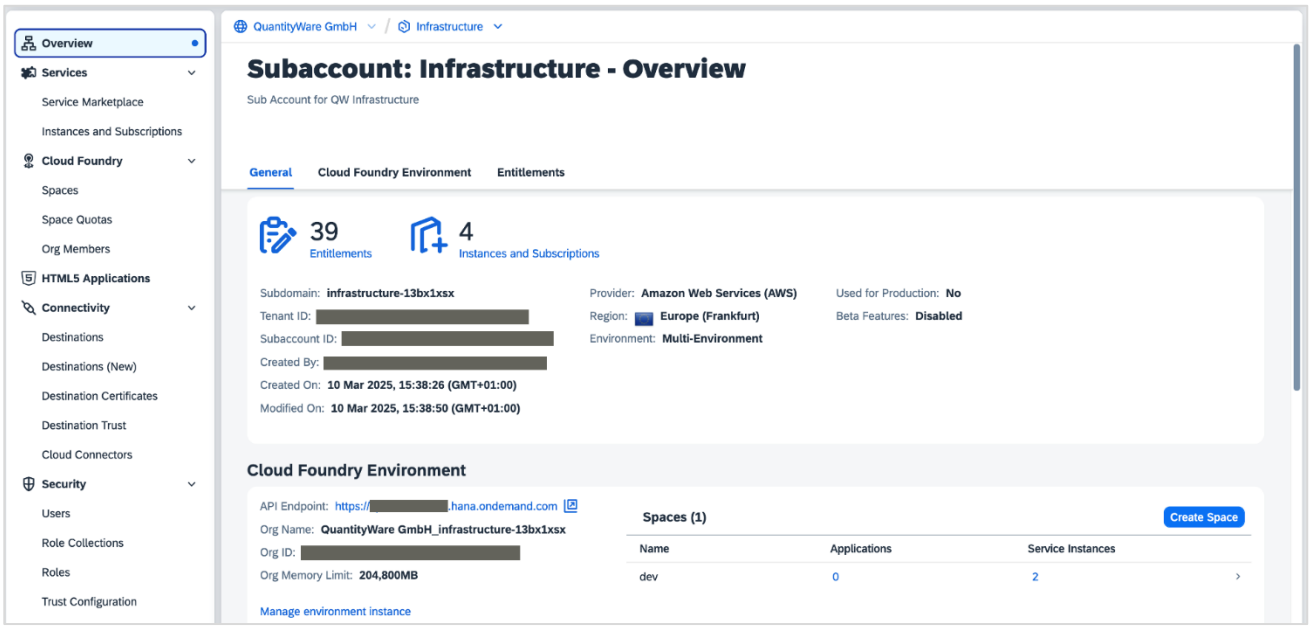
5.2. Create Destination in BTP

Log in to the SAP BTP Cockpit (e.g. https://emea.cockpit.btp.cloud.sap/cockpit).

The Global Account overview will be displayed:

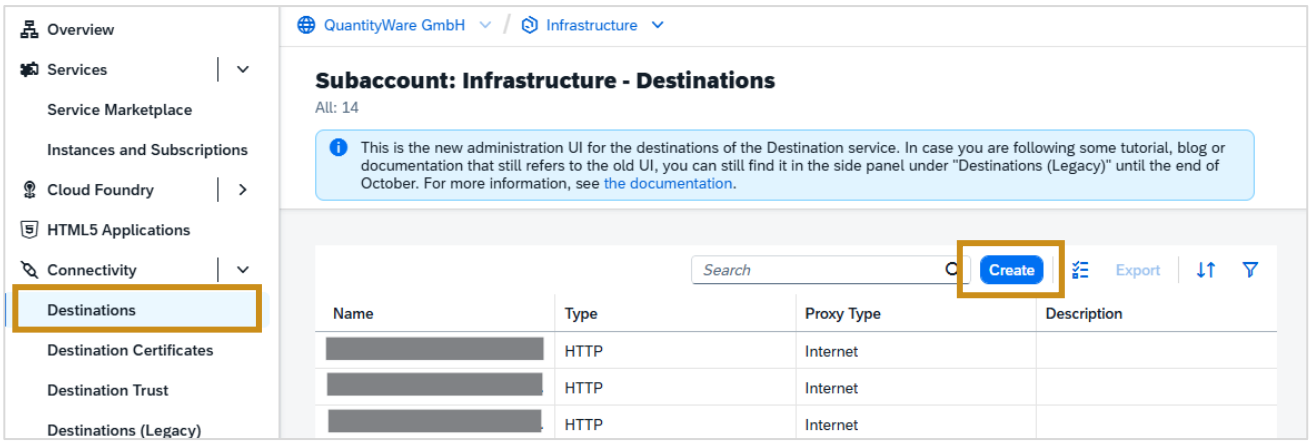


Click the relevant subaccount in the list, to view its Overview page:



From the navigation menu, in "**Connectivity**", click "**Destinations**".

The **Destinations** page will display, with any existing configured Destinations shown in the list:



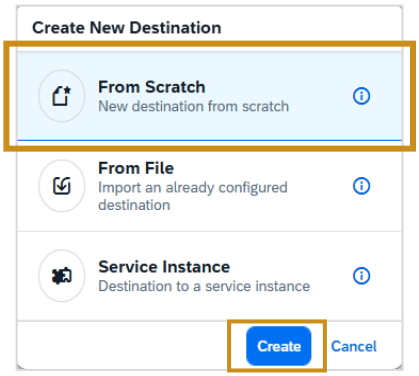
Click "**Create**".

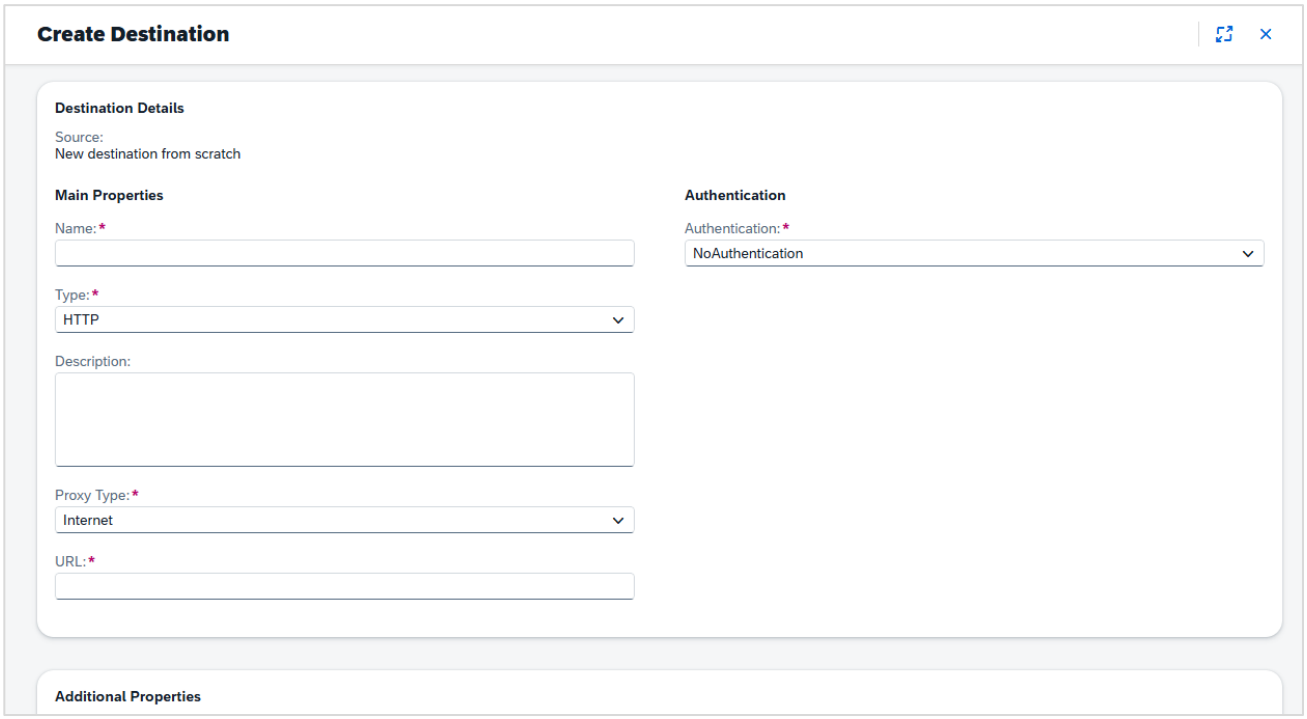
The **Create New Destination** options dialog will appear (see right):

Select "**From Scratch**"

Click "**Create**"

The **Create Destination** > **Destinations Details** Form will open to the right of the Destinations List:





NOTE: We will be connecting using a local user with access to the QAPI functions on the required client of our on-premise SAP system. If you wish to use a different authentication method, please enter the alternative information as required.

Fill in the form:

- In the **"Name"** field, enter a unique name for the Destination

Matching the QAPI iFlow Package

The QAPI iFlow package in the Accelerator Hub is configured to connect to a destination name: **OnPremRFCTarget_QW_QAPI_Test**

If you intend to import the QAPI iFlow package (in a later configuration step), **we recommend setting your destination to use the above name**. Note, however, that the iFlow's destination name can be changed after import.

If you intend to manually create your iFlow, please enter a name for the RFC target that matches your internal naming standards (e.g. OnPremRFCTarget_<SID>_<CLIENT>).

- In the "**Type**" field, select "**RFC**"

  A new section named "Target System Configuration" will be displayed below "Destination details".

- Ensure "**Proxy Type**" field is set to "**OnPremise**"
- Ensure "**Authorization Type**" field, is set to "**CONFIGURED_USER**"

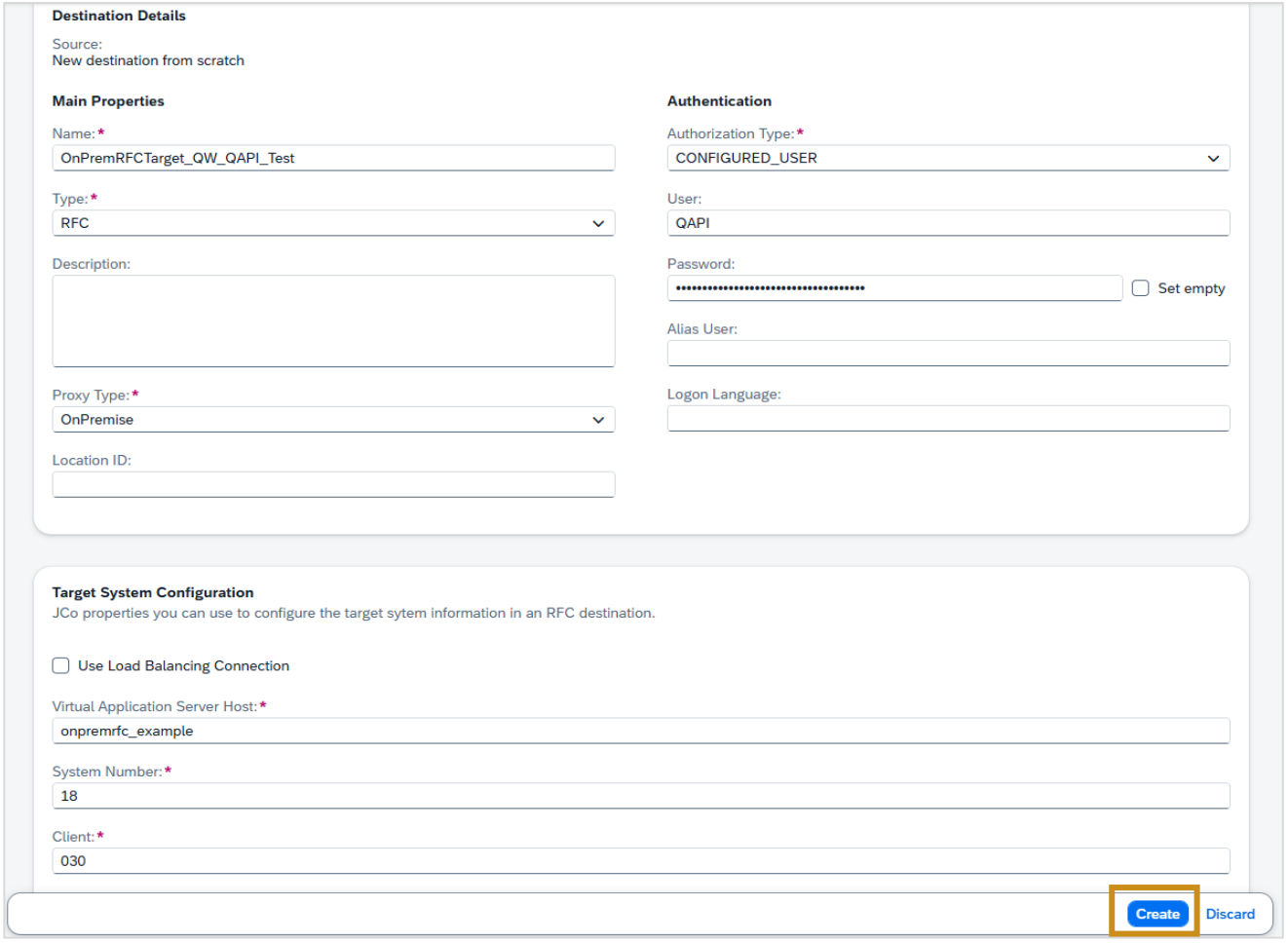
  The user details will be displayed below the "Authorization Type" field.

- In the "**User**" field, enter the name of the local user with RFC access to the QAPI functions on the client of your on-premise SAP system, created in RFC User Set Up
- In the "**Password**" field, enter the password of that user
- In the **Target System Configuration** section, fill in the following JCo properties:

Property Name	Example Value	Notes
Virtual Application Server Host	**onpremrfc_example**	The **Virtual Application Server** you created in SAP Cloud Connector.

System Number	**030**	The client of the on-premise SAP system to connect to.
Client	**18**	The **Virtual Instance Number** you created in SAP Cloud Connector.

Please see an example configuration below:

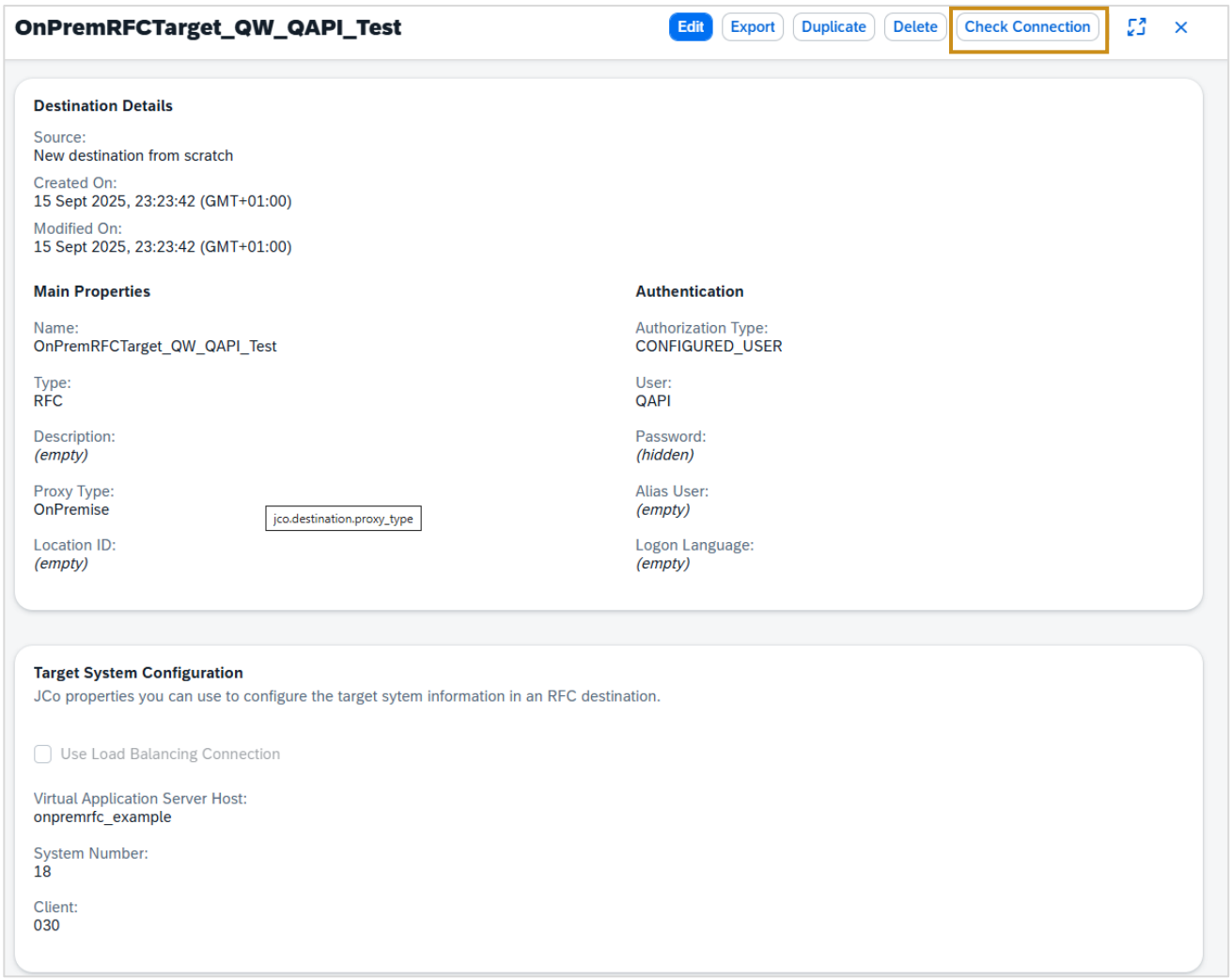


Click "**Create**".

The **Create Destination** form will close, and the newly created Destination will be listed in the Destinations list.

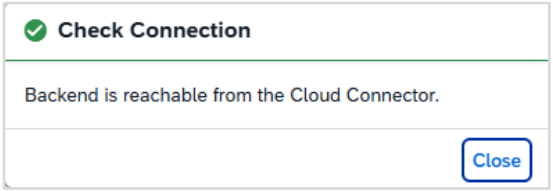
Click on the newly created Destination.

The **Destination Details** will be displayed for the Destination:



Click "**Check Connection**" in the top right. On smaller resolution screens, you may need to click the three horizontal dots ••• in the top right, then select "**Check Connection**" in the pop-up menu.

If it was configured correctly, you will see a successful message, for example:



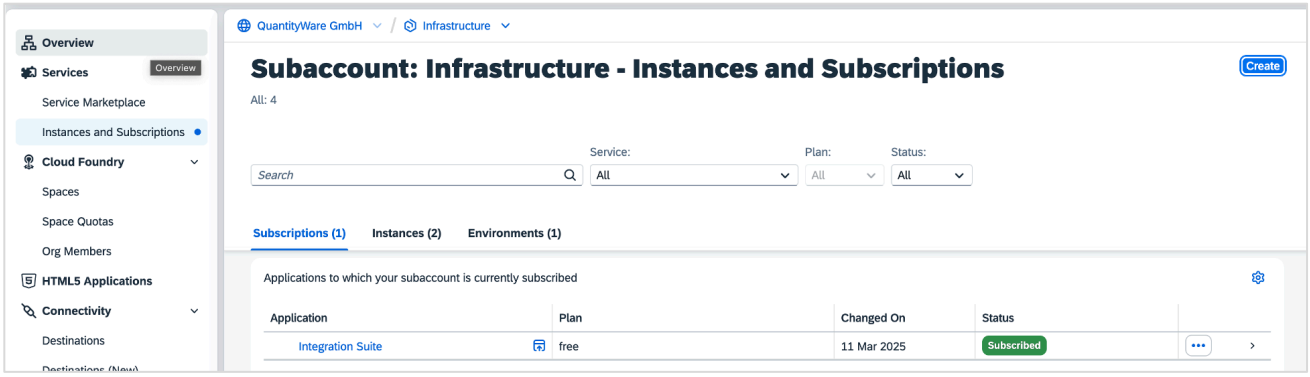
NOTE: This doesn't attempt to log in / authorize, so the user credentials are not checked – it only checks that BTP can connect to the destination's configured virtual host, defined in SAP Cloud Connector.

5.3. Generate Service Keys

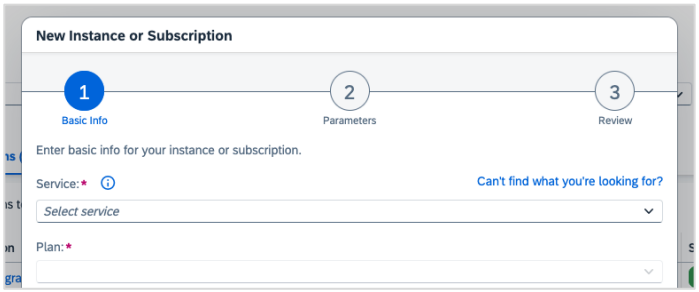
5.3.1. Create Integration Flow Instance + Keys

From SAP BTP Cockpit, navigate to the relevant subaccount.

From the navigation menu, under the "**Services**" section, click "**Instances and Subscriptions**" to view the Instances and Subscriptions page:



In the top right, click the "**Create**" button. The New Instance or Subscription dialog will display:

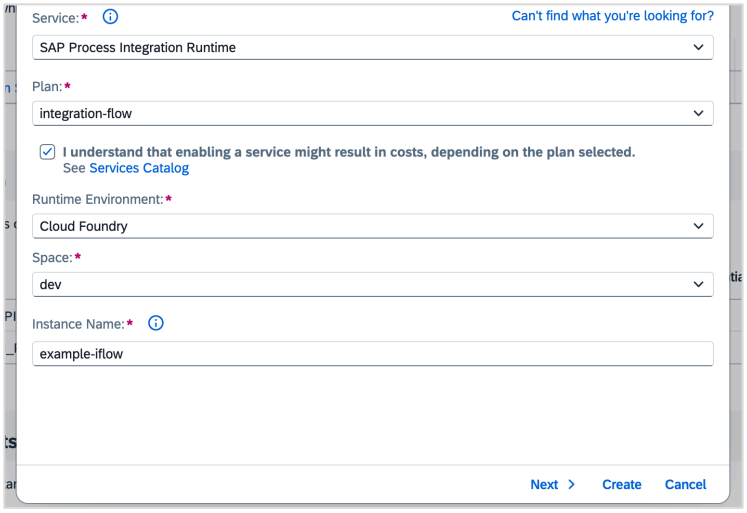


Fill in the form:

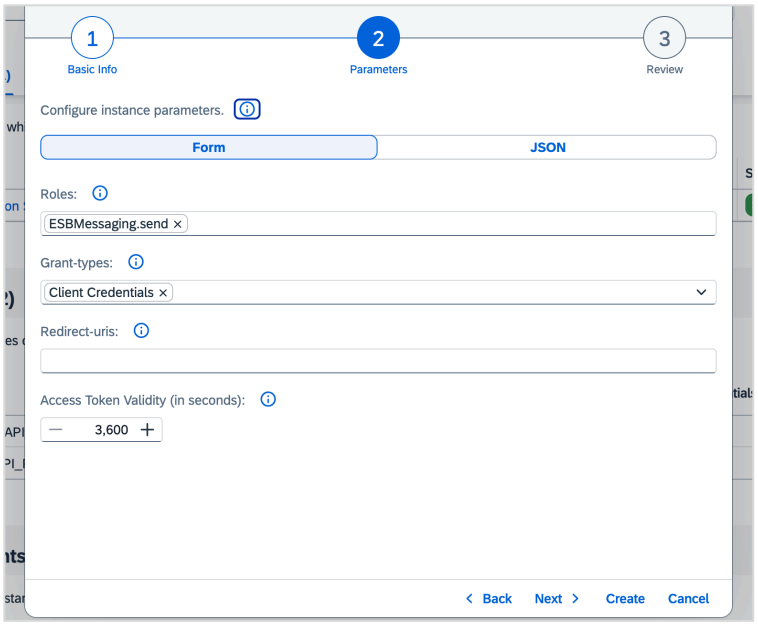
- In the "**Service**" field, select "**SAP Process Integration Runtime**".
- In the "**Plan**" field, select "**integration-flow**".

  Additional fields will be shown.
- Check the "**I understand that enabling...**" confirmation checkbox.
- In the "**Runtime Environment**" field, select "**Cloud Foundry**".
- In the "**Space**" field, select the Cloud Foundry space you wish to create the instance within.
- In the "**Instance Name**" field add your instance's name.

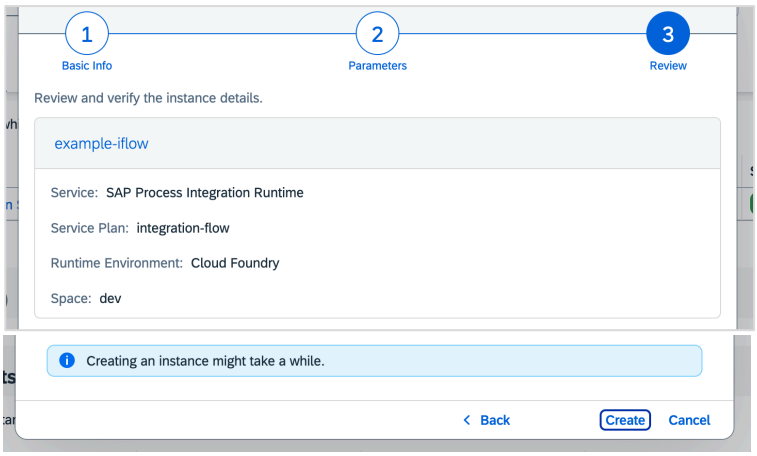
Example configuration:



Click "**Next**" to load the next panel:



Make no changes and click "**Next**" to load the next panel:

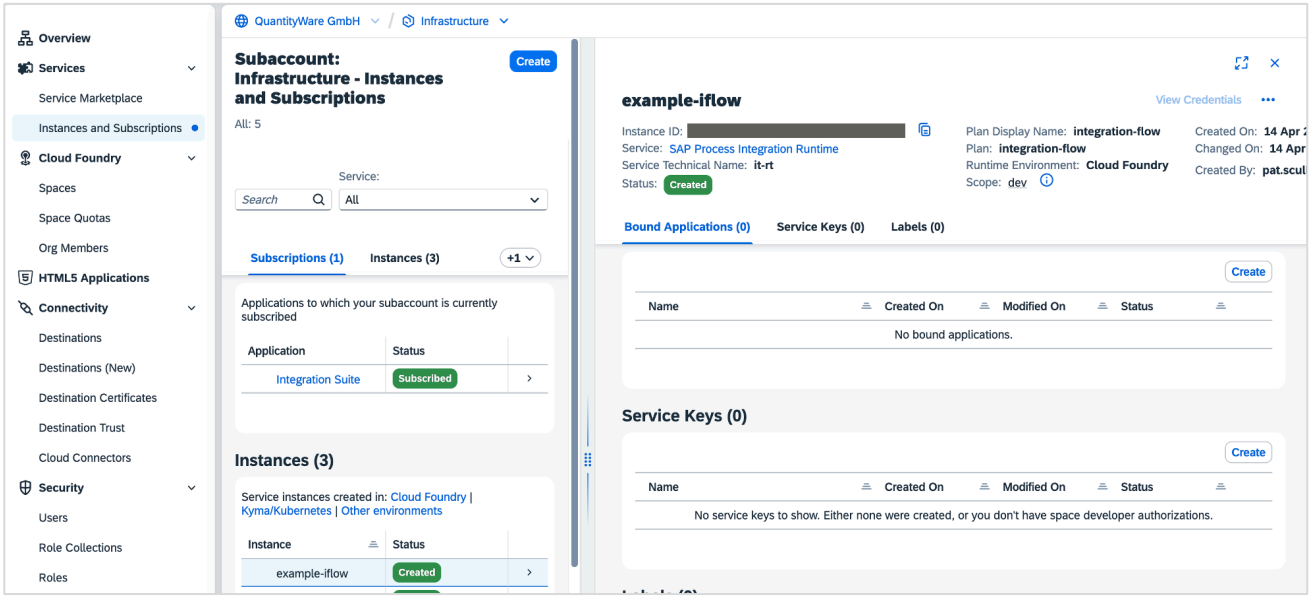


Click "**Create**".

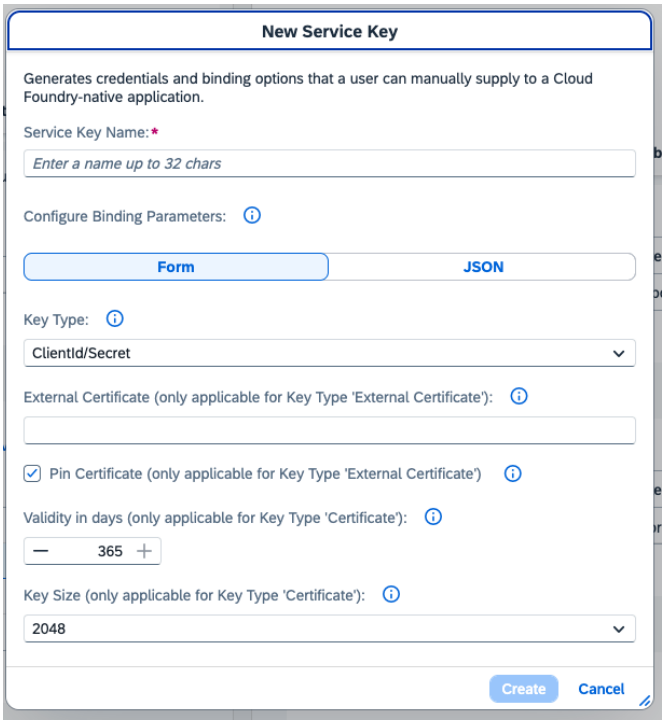
It may take some time to create. Initially you will see "Creation in Progress":

Once created, it will show "Created":

Click the arrow on the right of the new instance, to show the details:



In the "Service Keys (0)" section on the right, click the "**Create**" button. The New Service Key dialog will display:

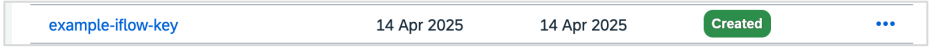


In the "**Service Key Name**" field, enter the name of your service key (e.g. "**example-iflow-key**").

Leave all other fields as default and click "**Create**".

It may take some time to create. Initially you will see "Creation in Progress":

After some time (it may take up to a minute), the status will update to "Created":

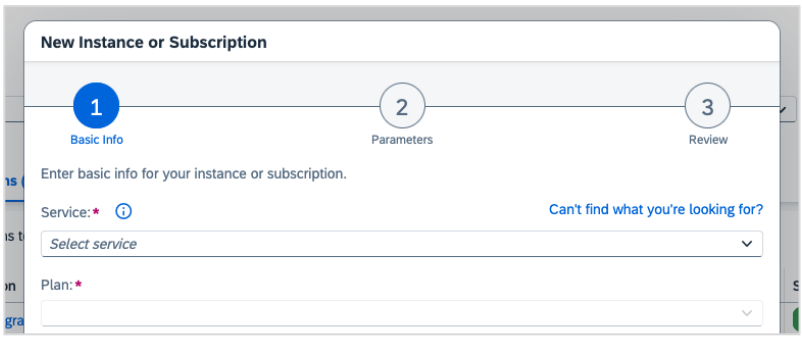


Close the side bar to show the full-width Instances table.

5.3.2. Create an API Instance + Keys

From the Instances and Subscriptions page, in the top right, click the "**Create**" button.

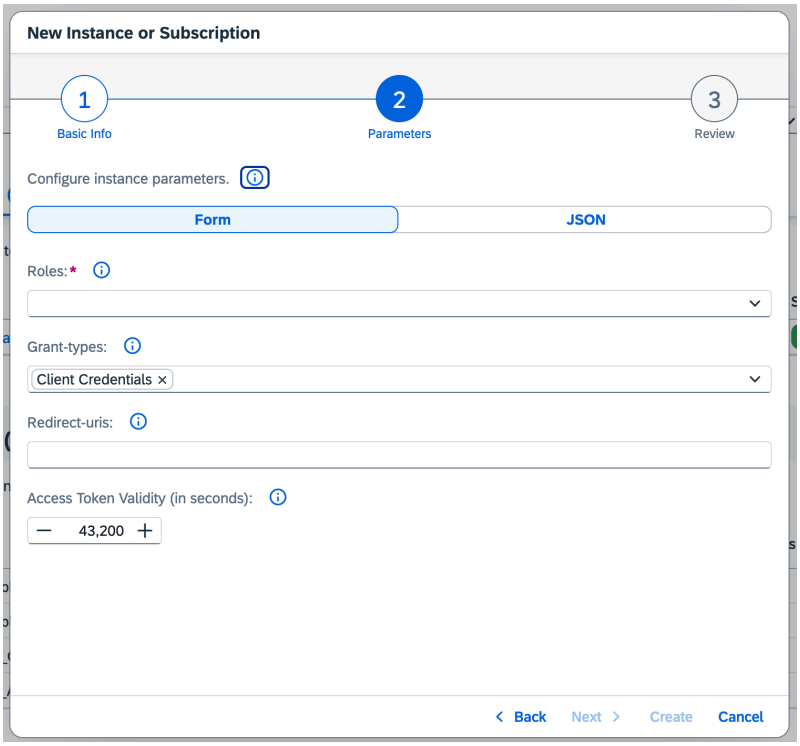
The New Instance or Subscription dialog will display:



Fill in the form:

- In the "**Service**" field, select "**SAP Process Integration Runtime**".
- In the "**Plan**" field, select "**api**". Additional fields will be shown.
- Check the "**I understand that enabling...**" confirmation checkbox.
- In the "**Runtime Environment**" field, select "**Cloud Foundry**".
- In the "**Space**" field, select the existing Cloud Foundry space you wish to create the instance within.
- In the "**Instance Name**" field add your instance's name.

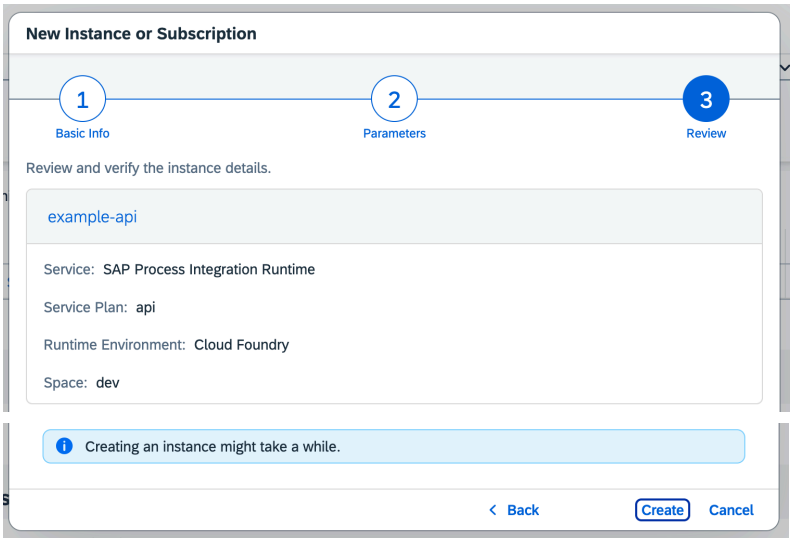
Click "**Next**" to load the next panel:



In the "**Roles**" field, select the following roles:

- **AuthGroup_Administrator**
- **AuthGroup_IntegrationDeveloper**

Leave the other fields as default and click "**Next**" to load the next panel:



Click "**Create**". It may take some time to create, as per the integration-flow Instance.

Click the arrow on the right of the new instance to show the details.

In the "Service Keys (0)" section on the right, click the "**Create**" button. The New Service Key dialog will display.

Fill in the form:

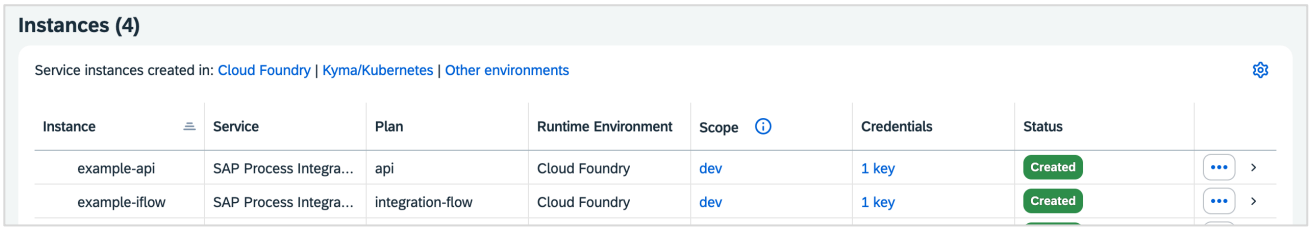
- In the "**Service Key Name**" field, enter the name of your service key (e.g. "**example-api-key**").
- Leave all other fields as default

Click “**Create**”.

It may take some time to create. Initially you will see “Creation in Progress”:

After some time (it may take up to a minute), the status will change to “Created”:

Close the side bar to show the full-width Instances table. You will see both Instances created:



6. Integration Suite

Integration Suite is used to design, publish and manage the APIs required to access the QAPI functions outside of SAP system environments.

This section will detail the following:

- Configuration/import and deployment of the QAPI iFlow with RFC adapter
- Creating the API Provider, exposing the endpoint of the iFlow
- Create the API Proxies to generate the frontend URL

Policies are also utilised for generating and verifying Access Tokens to secure the API Request. The API Proxy routes the request to the actual backend API, as defined by the API Provider.

6.1. Import / Create iFlow

The iFlow logic required for the API can either be **imported** directly from the QAPI iFlow Package, or **created manually**.

QAPI iFlow Package

We provide a pre-configured package from a ZIP file containing the iFlow logic.

Once imported, the artifacts (iFlow schematics) are editable, allowing you to amend the pre-configured connection details for both HTTP and RFC Adapters, as required.

The package contains two iFlow artifacts:

1. **RFC QAPI Connection to Test SID** - Pre-configured for use with your SAP Test System
2. **RFC QAPI Connection to Production SID** - Pre-configured for a Production system if you choose to use it rather than create your own (which is described in a later chapter)

6.1.1. Import iFlows from the QAPI Package

In this section, we will detail the process for importing the preconfigured iFlows in the QAPI package on the SAP Business Accelerator Hub.

Obtain the QAPI package from the SAP Business Accelerator hub.

QAPI Package Available Q4 2025

The QAPI Package will be available in the SAP Business Accelerator hub in Q4 2025.

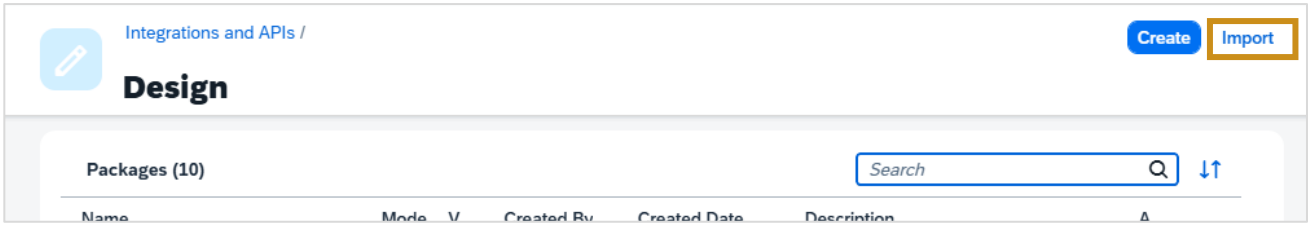
Once done, steps for downloading the package will be added to this document.

Login to the BTP Cockpit.

From the navigation menu, under the "**Services**" section, click "**Instances and Subscriptions**", then click "**Integration Suite**":

Select "**Design**" from the navigation menu, then "**Integrations and APIs**"

Click the "**Import**" button:

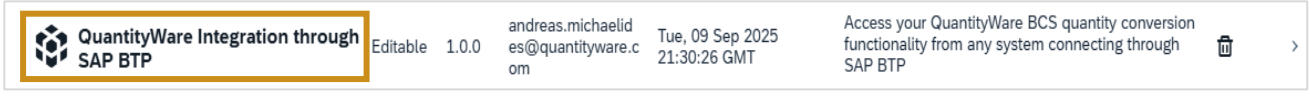


Your web browser's Open dialog will be displayed.

Locate your file and open it to upload it into the Integration Suite.

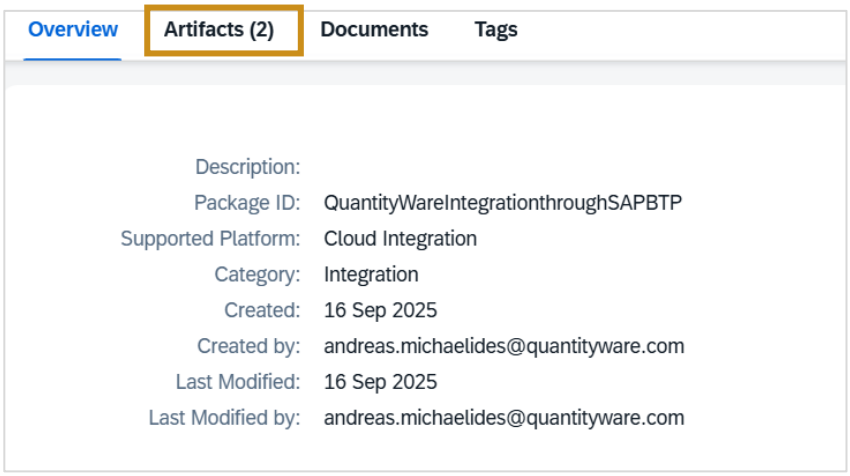
A new entry, "**QuantityWare Integration through SAP BTP**", will appear in the list of packages.

Click the package name:



The “QuantityWare Integration through SAP BTP” package information will be displayed.

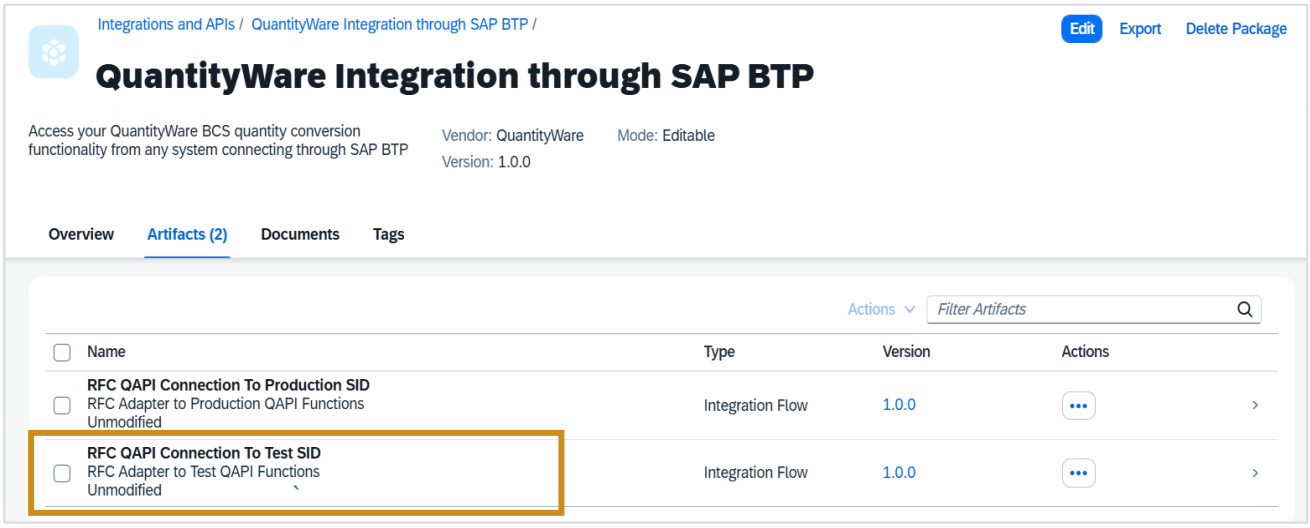
Click on the “**Artifacts (2)**” tab:



The artifacts list will be displayed.

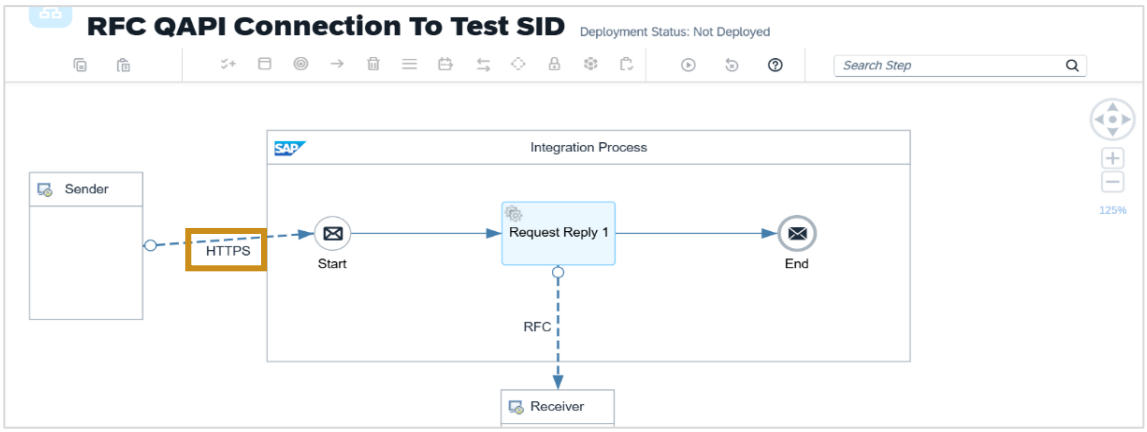
In this document we will focus on the **Test** artifact.

Click on “**RFC QAPI Connection To Test SID**”:



A diagram representing the schematic of the **RFC QAPI Connection To Test SID** artifact will be displayed.

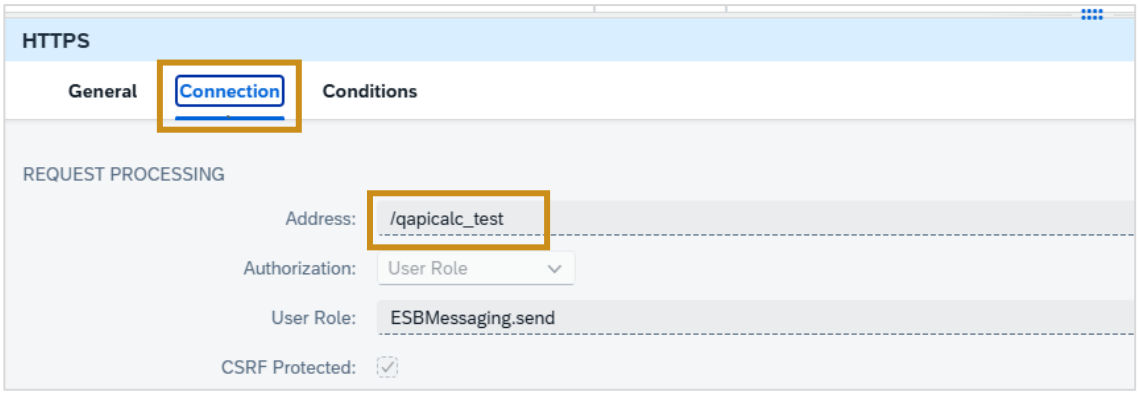
Double-click on the “**HTTPS**” adapter in the diagram:



The "HTTPS" details section will be shown at the bottom of the page. If you have not already, drag the bottom bar on the page up to view the details.

In the **HTTPS** details section, click the "**Connection**" Tab.

The "Request Processing" information will be shown:



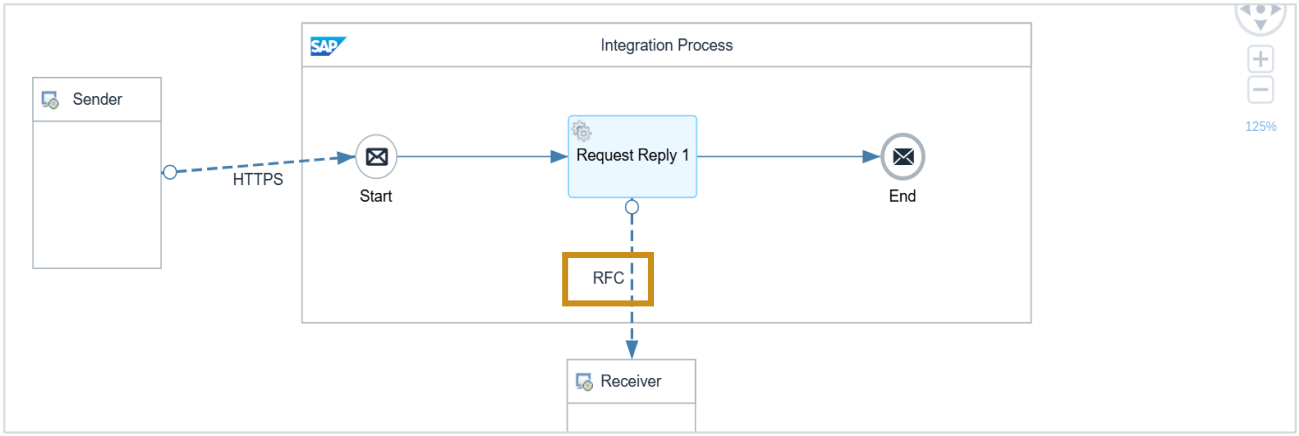
Address is Endpoint URL

The address shown above will make up the endpoint of the iFlow URL.

For this illustration, **/qapicalc_test** will be used to access the QAPI ABAP functions on the TEST SAP Server, though any endpoint name can be used.

We suggest using a descriptive name for this (e.g. **/qapicalc_<SID>**, where <SID> corresponds to the target SAP System ID, or **/qapicalc_prod** for your productive QAPI iFlow URL endpoint).

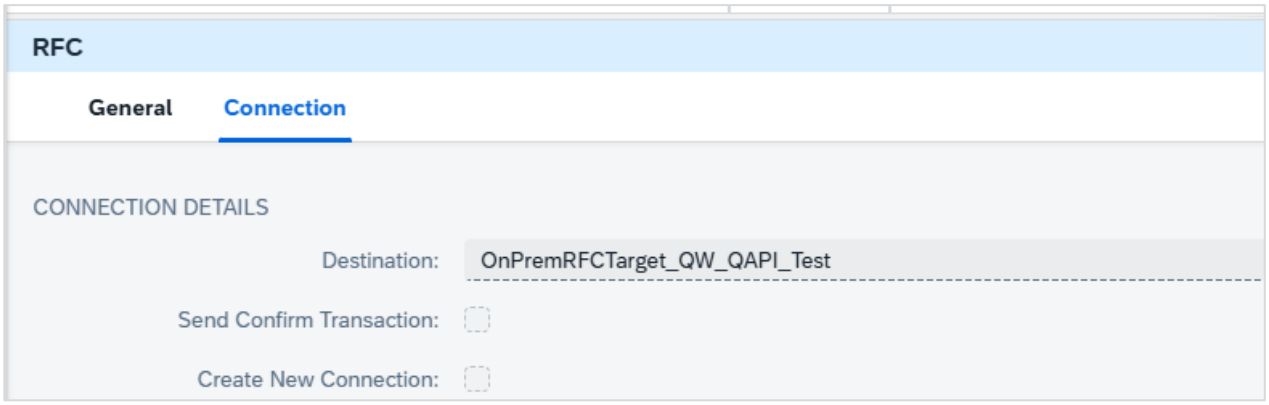
In the diagram, double-click on the "**RFC**" Adapter:



A "RFC" details section will be shown at the bottom of the page.

In the RFC details section, click the "**Connection**" Tab.

The "Connection Details" information will be shown:



Destination

Observe the Destination: **OnPremRFCTarget_QW_QAPI_Test**. This pre-configured Destination name is for illustration, connecting to a Test SAP Server.

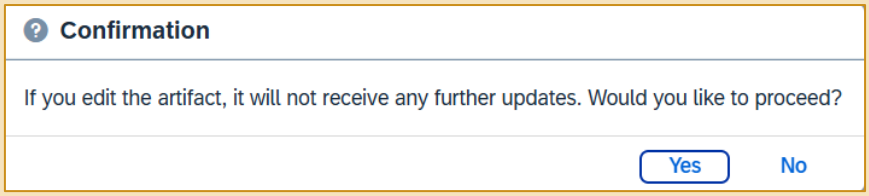
This should match the name of the destination you created in defined in Create Destination in BTP.

Need to Change the Destination?

If the destination in the iFlow does not match the name of the destination created in Create Destination in BTP, you can edit the iFlow's destination name here.

Click the "**Edit**" button

Click "**Yes**" in the confirmation dialog below:



Update the destination name

Click "**Save**"

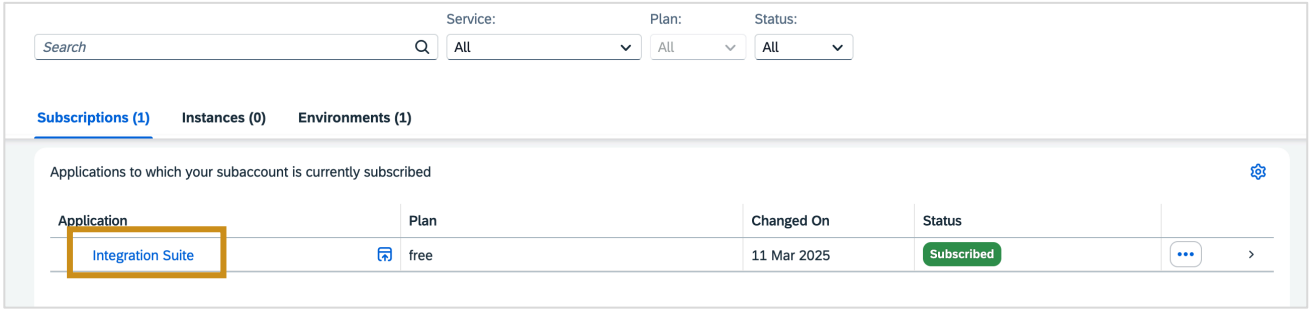
Deploy the iFlow by following the instructions in Deploy the iFlow.

6.1.2. Create an iFlow with the RFC Adapter

In this section, we will showcase the basic iFlow diagram required for the API. No extra processing is carried out in the integration flow other than a request via the RFC adapter.

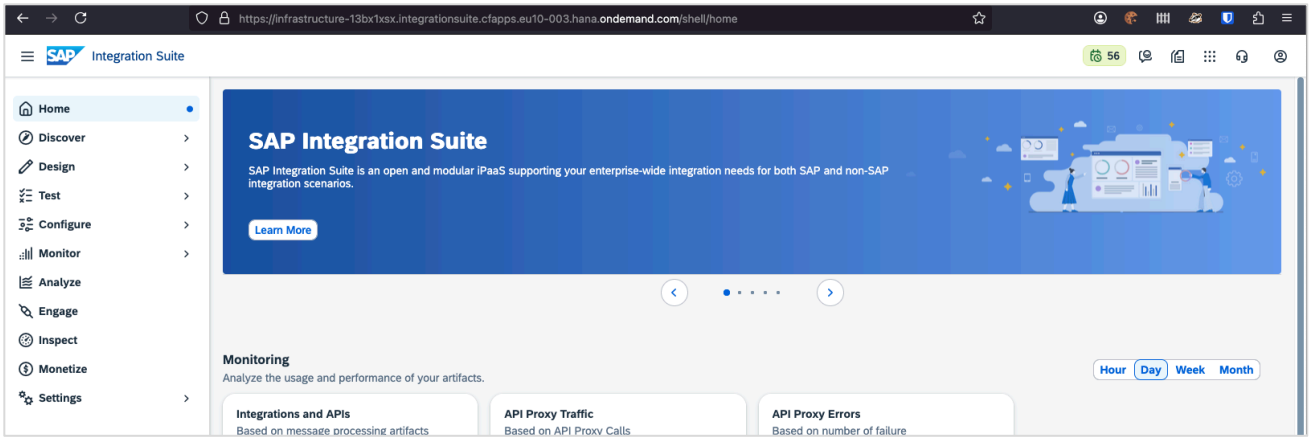
Login to the BTP Cockpit.

From the navigation menu, under the "**Services**" section, click "**Instances and Subscriptions**", then click "**Integration Suite**":



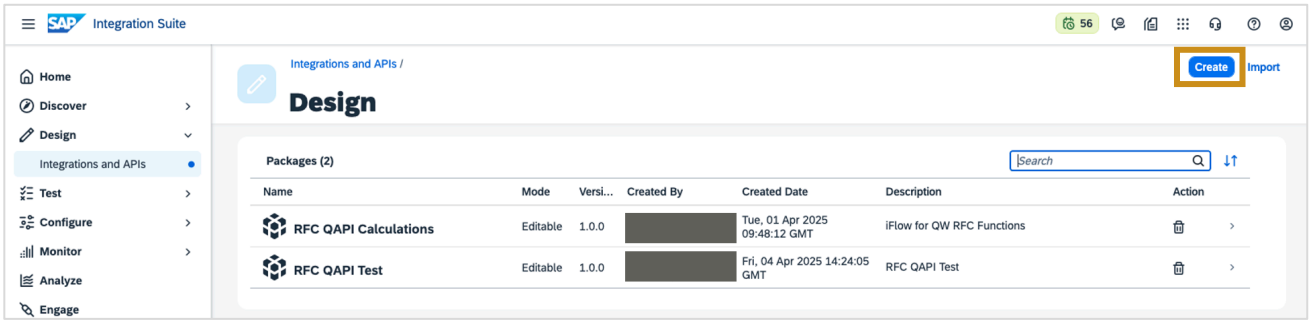
This will load the Integration Suite in a new browser tab.

NOTE: You may need to log in with your SAP Universal ID. Once done, the Integration Suite will display:

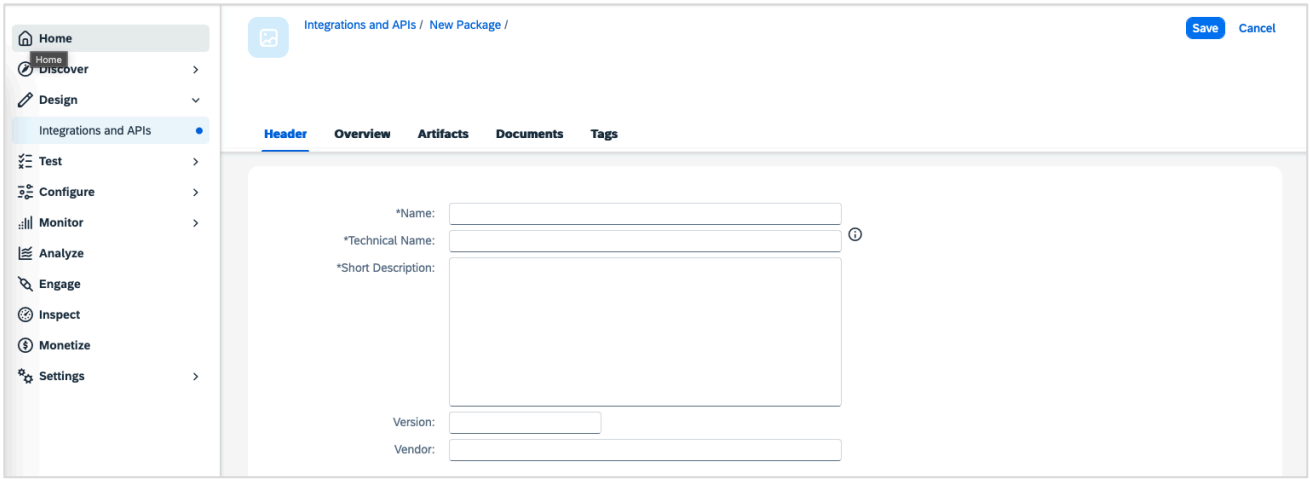


From the navigation menu, click "**Design**", then "**Integrations and APIs**". The Integrations and APIs Design page will display.

Click the "**Create**" button in the top right:



The New Package page will display:

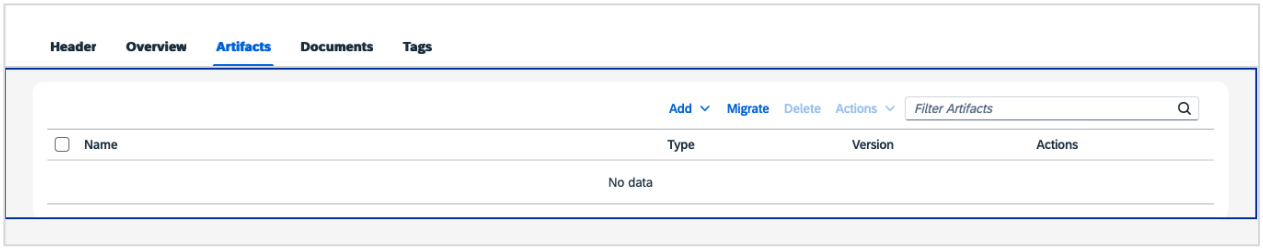


Fill in the form:

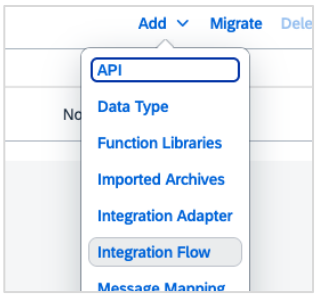
- In the "**Name**" field, enter the name for the package (e.g. "QuantityWare Integration through SAP BTP")
- The "**Technical Name**" field should auto-populate with a sanitised name based on the value in the "Name" field. You can edit this if desired.
- In the "**Description**" field, enter a short description that describes the purpose of the package (e.g. "Connection to QAPI RFC").

Click the "**Save**" button in the top right. This will lock the Technical Name and set an initial version of v1.0.0.

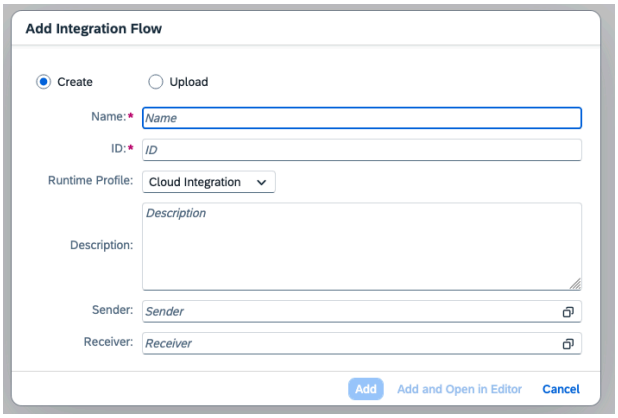
Click the "**Artifacts**" tab to show the (empty) Artifacts list:



Click the "**Add**" button. In the pop-up menu select "**Integration Flow**":



The Add Integration Flow dialog will be displayed:

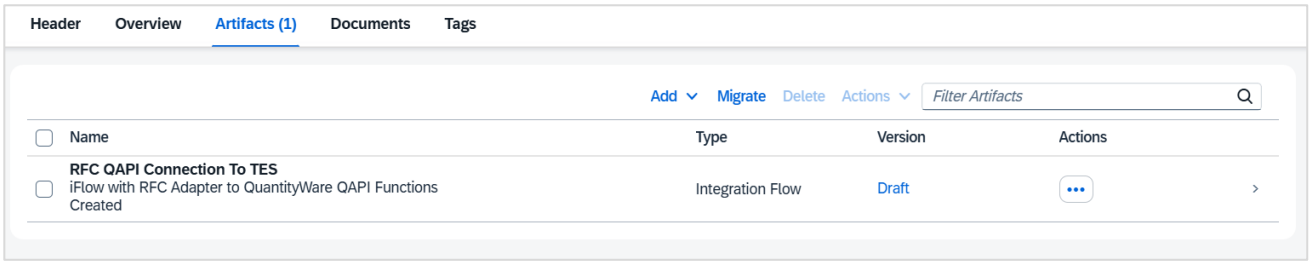


Fill in the form:

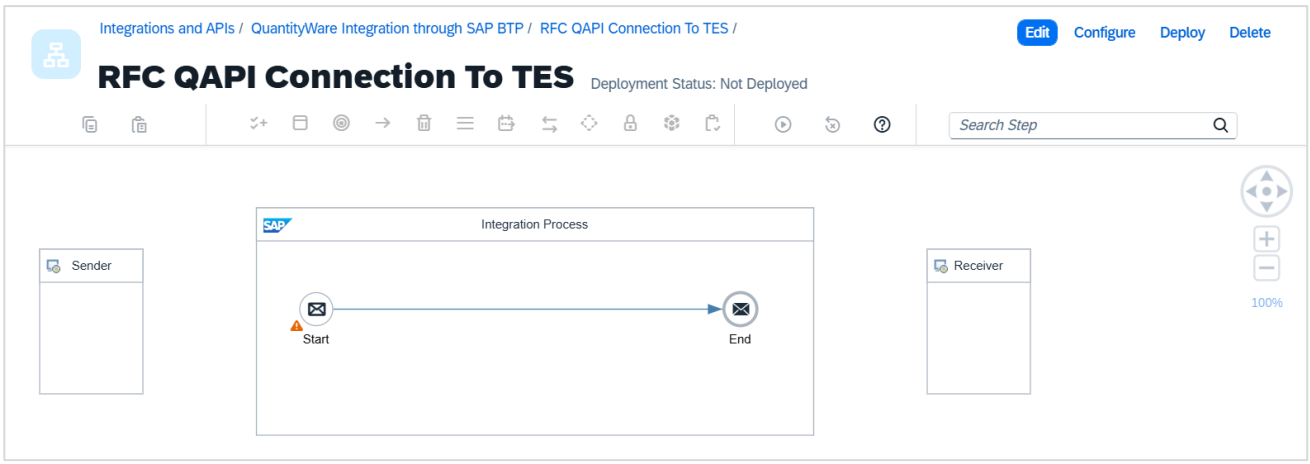
- In the "**Name**" field, enter the name if your integration flow (e.g. "RFC QAPI Connection To <SID>" – we have used "RFC QAPI Connection To TES")
- In the "**Description**" field, enter a description of your integration flow (e.g. "iFlow with RFC Adapter to QuantityWare QAPI Functions")
- The "**ID**" field should auto-populate with a sanitised name based on the value in the "Name" field. You can edit this if desired
- Leave all other fields as default

Click "**Add**".

The new Integration Flow will appear in the Artifacts list:



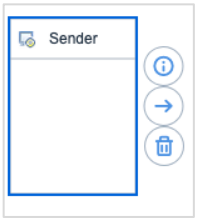
Click on your Integration Flow. The Integration Flow Editor page will display:

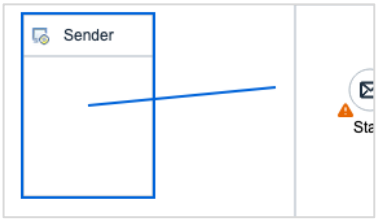


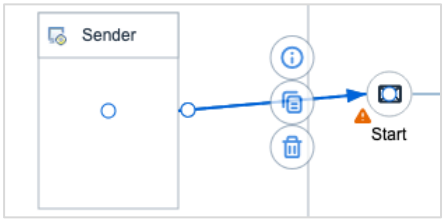
Click the "**Edit**" button at the top of the page to enter edit mode.

Create a connection from the "Sender" to the "Start" stage:

1) Click the "**Sender**" block...
2) ...click the arrow to its left, and drag it...
3) ...to the "**Start**" item in the "Integration Process" block:

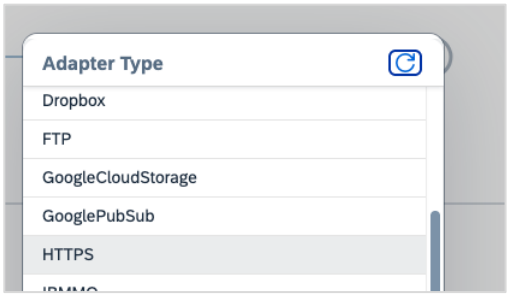
1: 

2: 

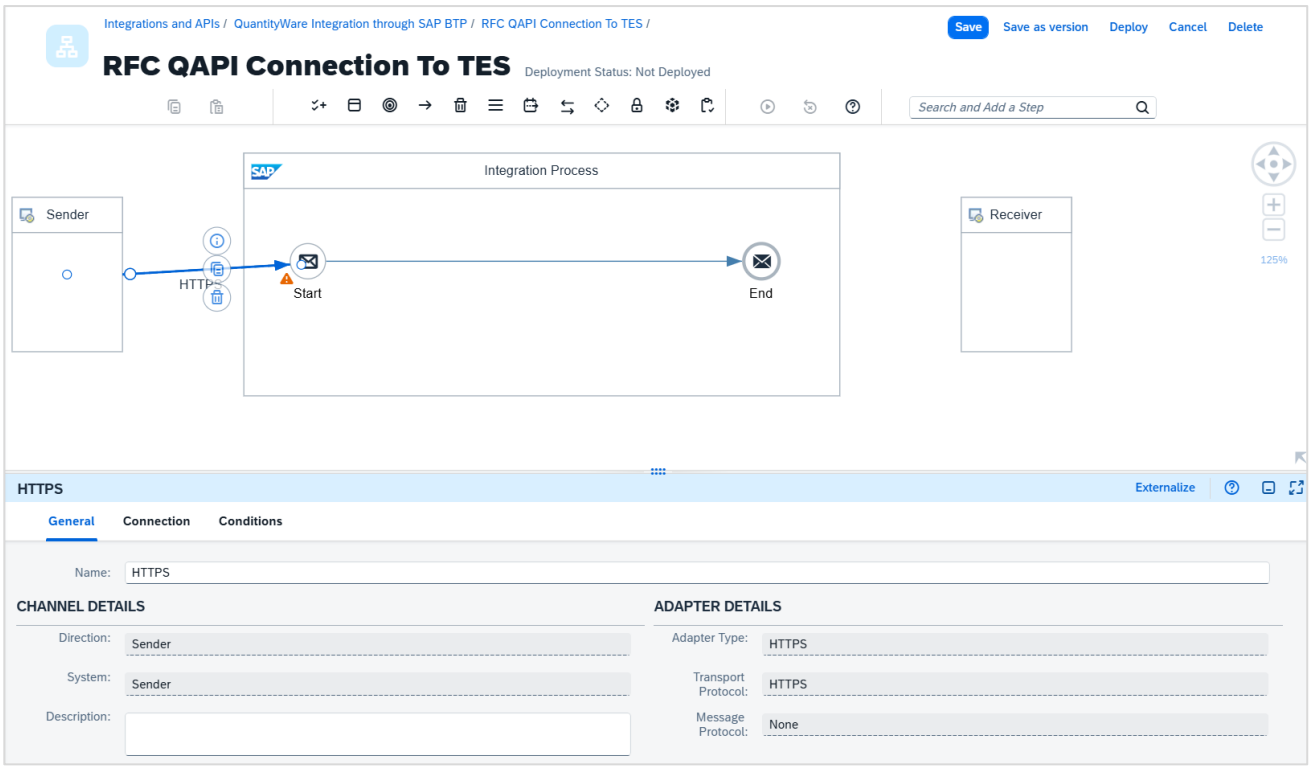
3: 

The Adapter Type pop up menu will appear.

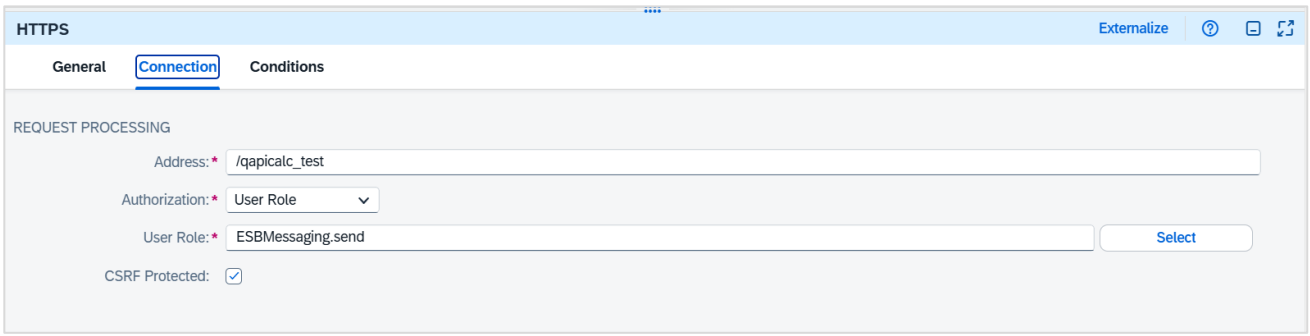
Select "**HTTPS**":



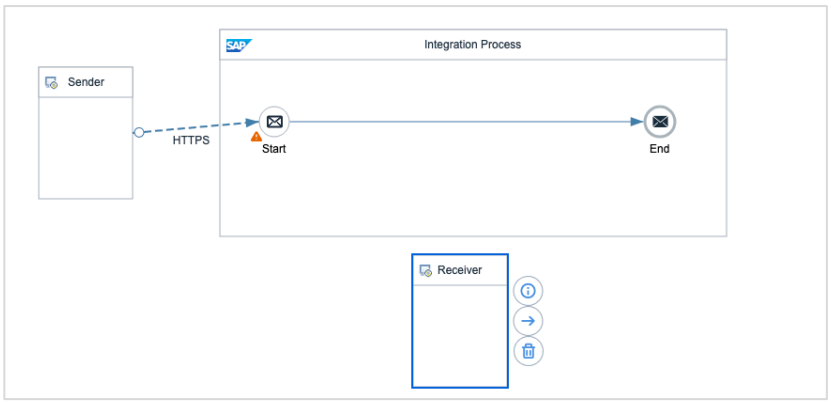
NOTE: If you have not already, drag the bottom bar on the page up to view the details of the new HTTPS component:



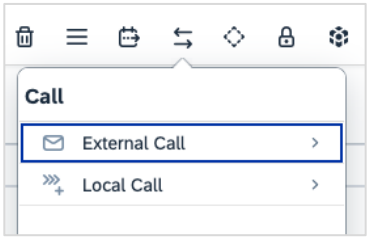
In the HTTPS details section, click the "**Connection**" tab and enter the address for connecting to the iFlow (this will come after the domain name):

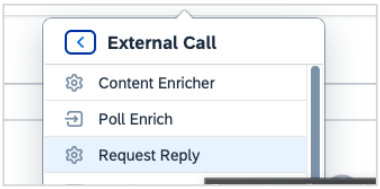


In the diagram, drag the "**Receiver**" block under the "Integration Flow" block:

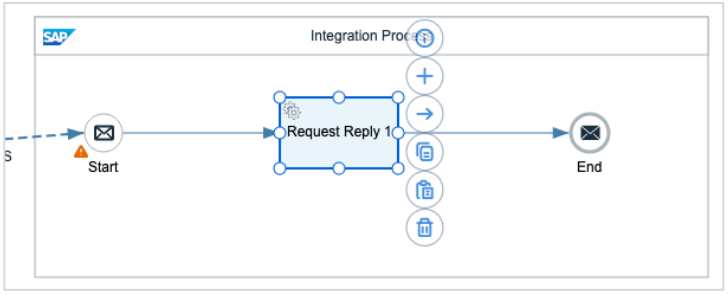


Click the "**Double arrow**" button above the diagram, and in the pop up menu select "**External Call**", then "**Request Reply**":





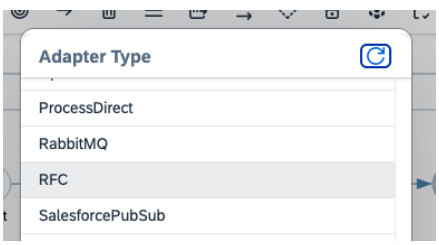
Move your mouse to the middle of the connector within the "Integration Process" block and click to add the "Request Reply 1" item. The item will be added as below:



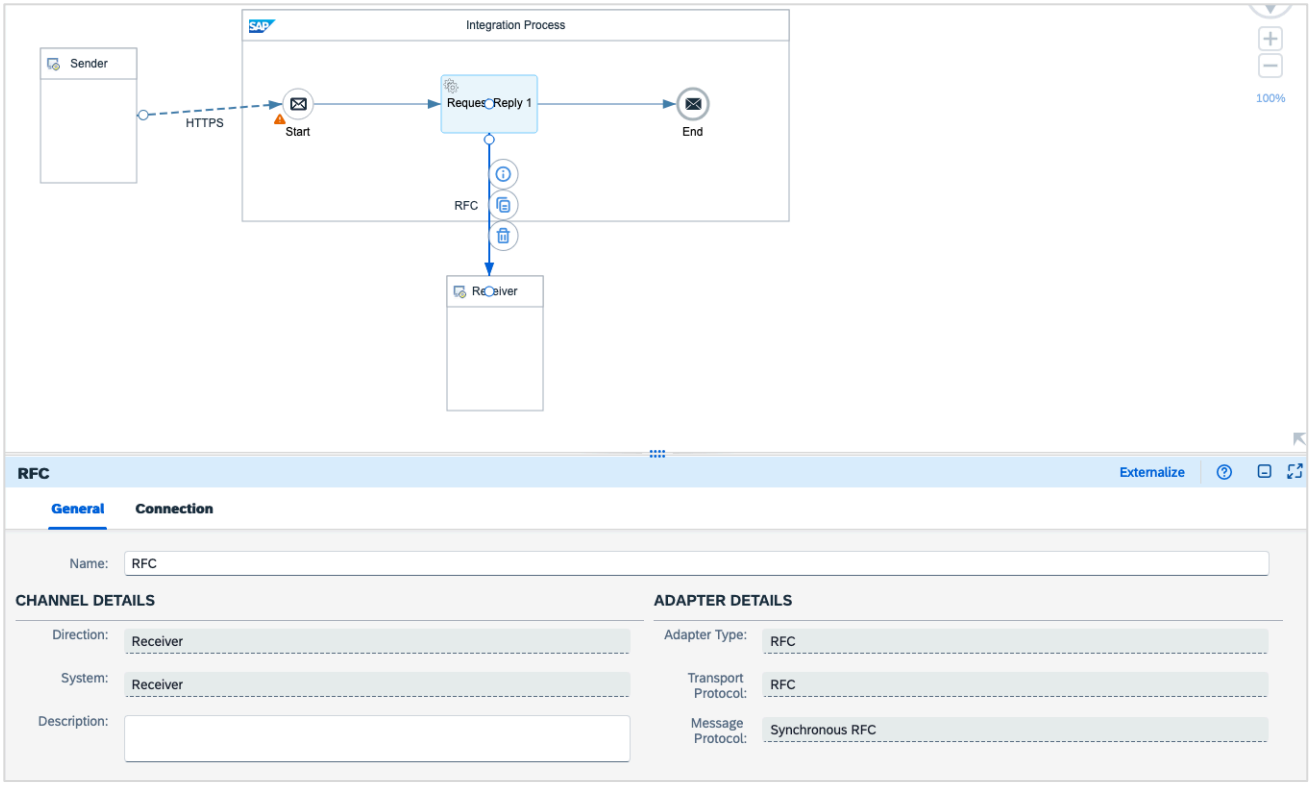
Click the arrow to the left of the "Request Reply 1" block and drag it down to the "Receiver" block below.

The Adapter Type pop up menu will appear.

Select "**RFC**":



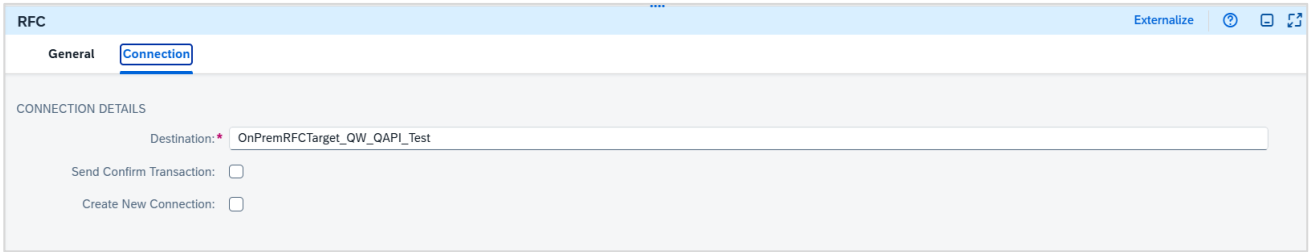
This will add the RFC connection:



In the RFC details section, click the "**Connection**" tab.

In the "**Destination**" field, enter the name of the destination.

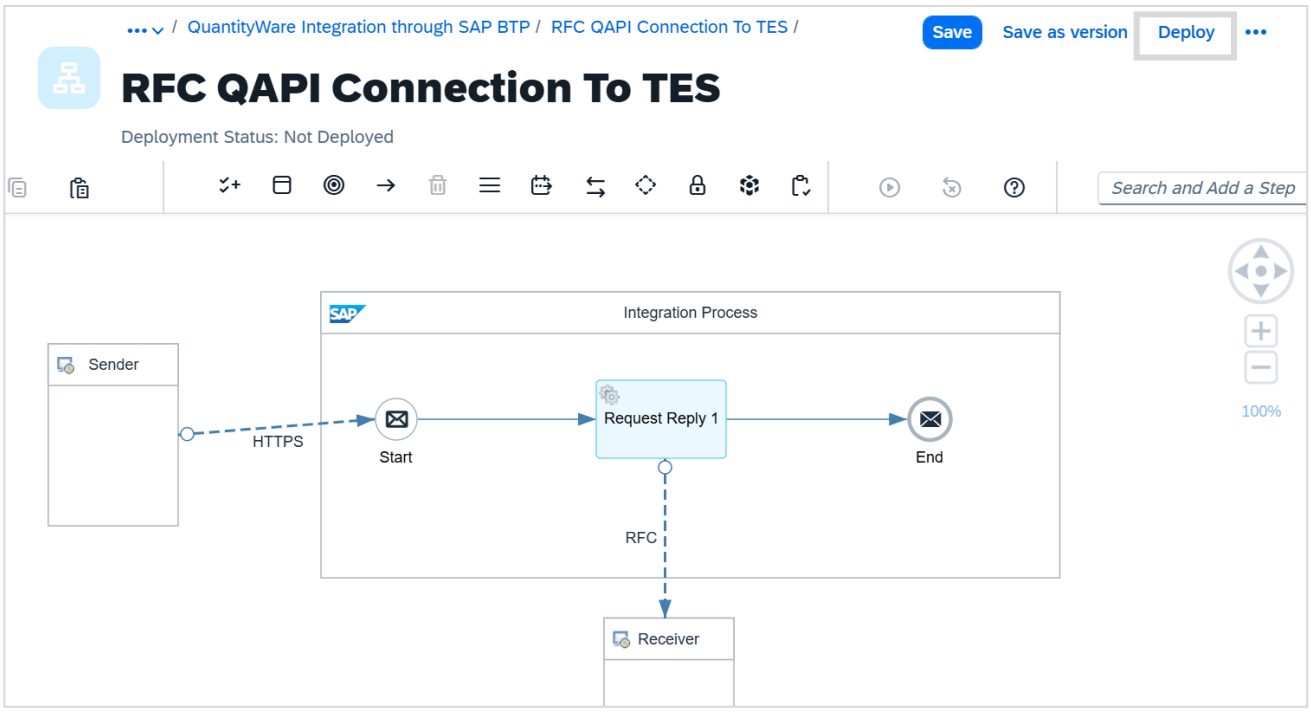
This must match the name of the destination you configured in Create Destination in BTP:



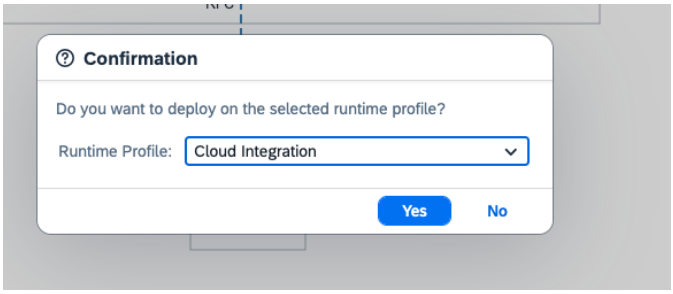
Click the "**Save**" button in the top bar (note: this will remove the warning icon on the "Start" item in the diagram).

6.2. Deploy iFlow

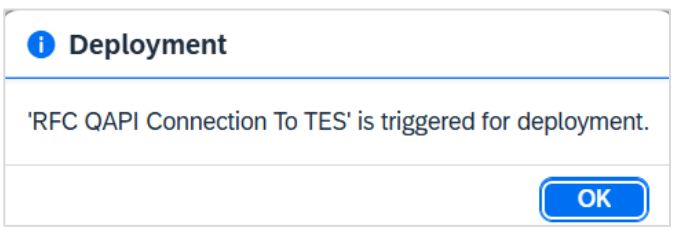
From within the Integration Flow Editor for your newly imported / created iFlow, click the "**Deploy**" button in the top right of the bar.



A confirmation dialog will appear:

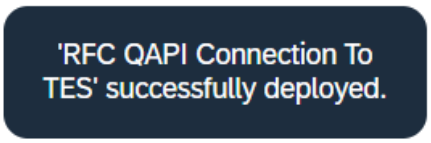


Click "**Yes**":



Click "**OK**":

After a few seconds, a message will appear at the bottom to state the iFlow has been deployed:

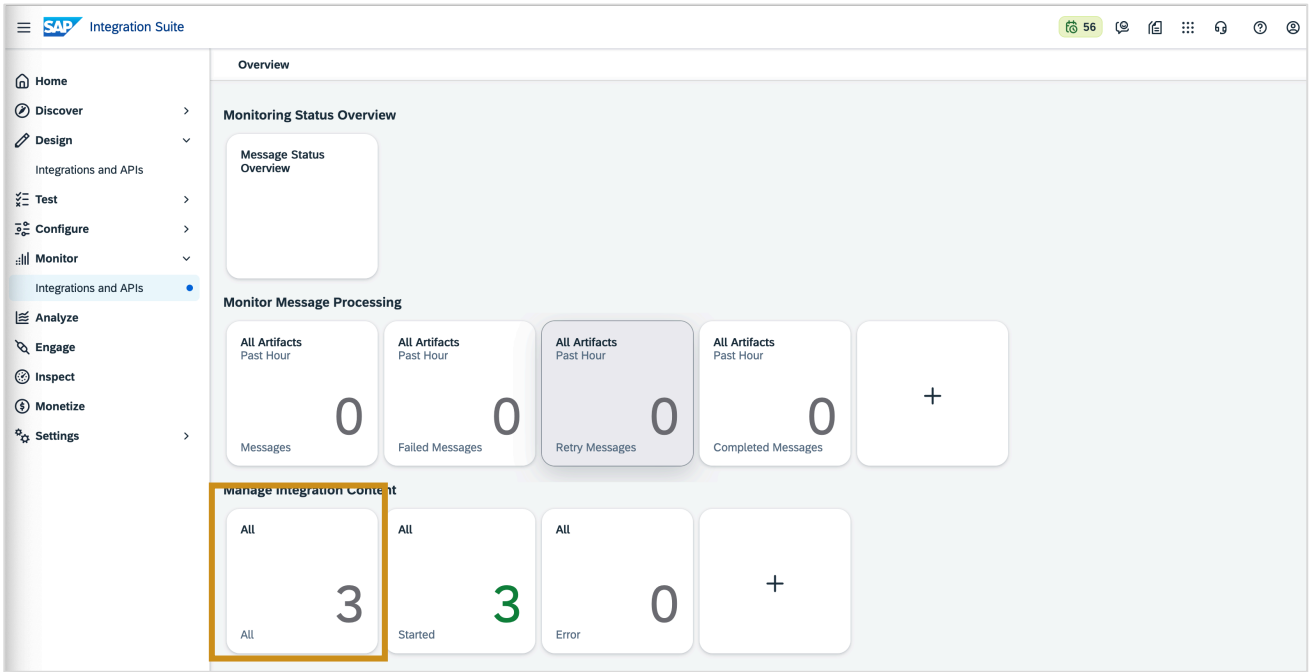


6.2.1. Obtain iFlow Target Endpoint

To obtain the iFlow's target end point, perform the following steps:

In the navigation menu click "**Monitor**", then "**Integrations and APIs**".

The Overview page will load.

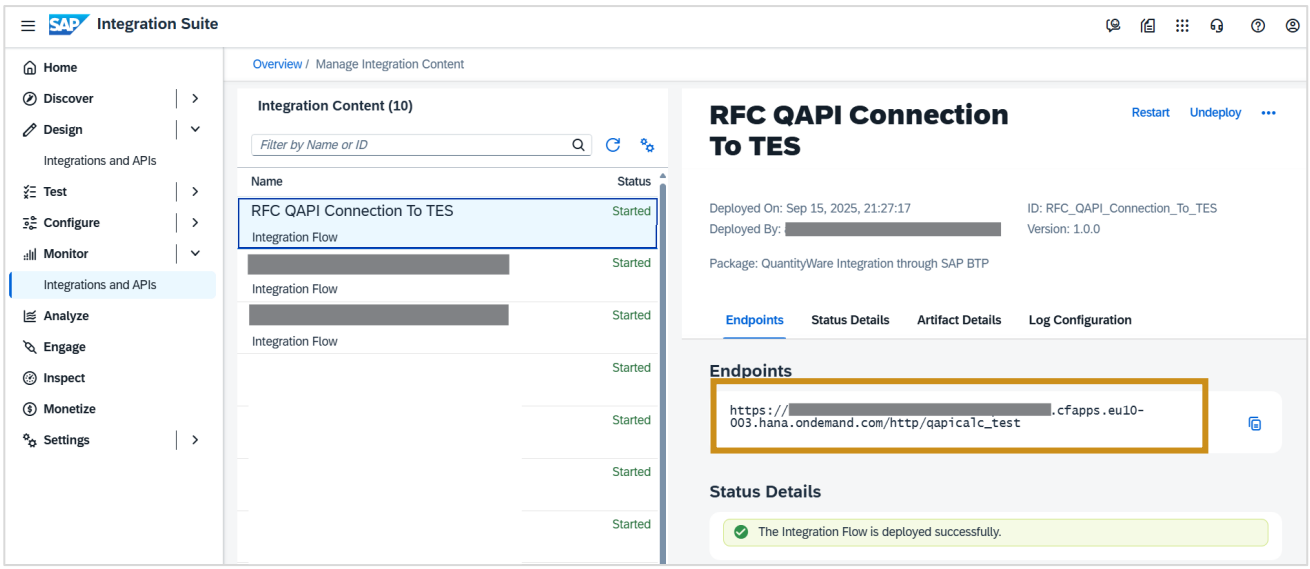


Under the "**Manage Integration Content**", click the first "**All**" tile on the left (as highlighted above).

The list of iFlows will be shown.

NOTE: Sometimes an integration flow's endpoint may take several minutes to appear. If you do not see an endpoint, please wait a few minutes and refresh the page.

Select your iFlow from the list (if not already selected) to view the Endpoints:



The endpoint URL for the iFlow is shown (as highlighted above).

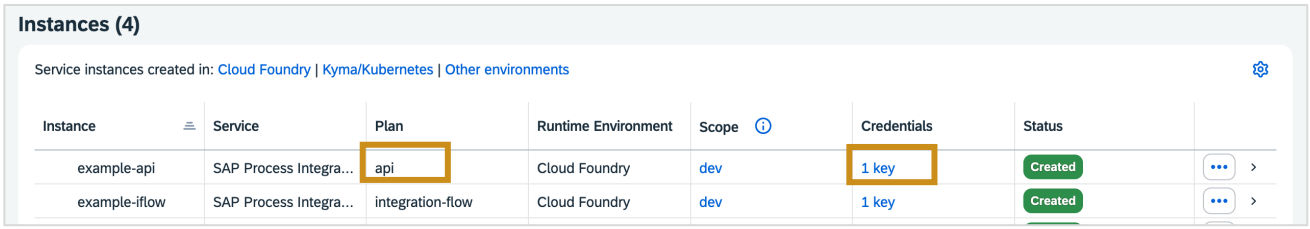
6.3. Create API Provider

6.3.1. Obtain Credentials

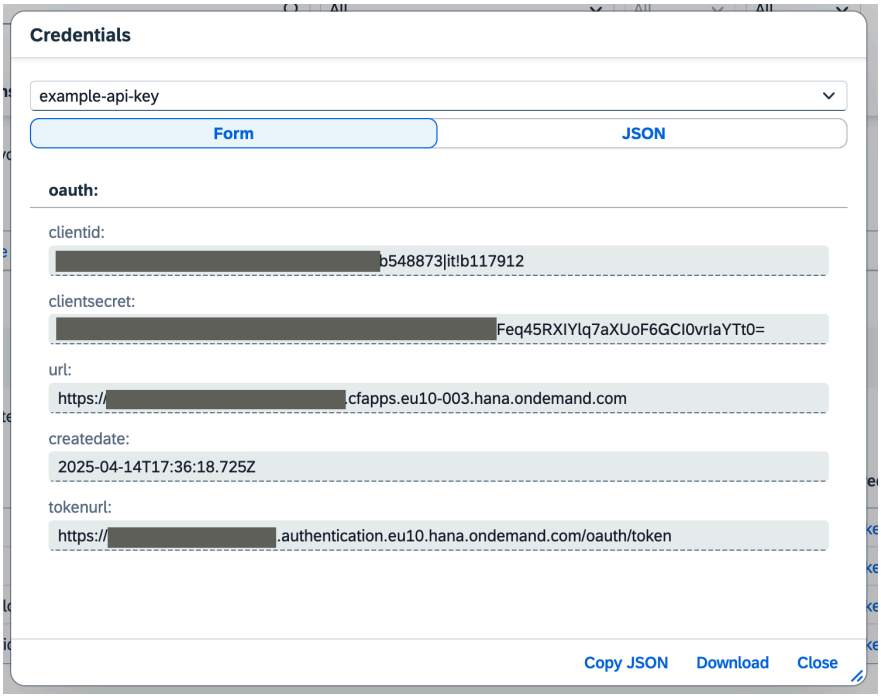
From SAP BTP Cockpit, navigate to the relevant subaccount.

From the navigation menu, under the "**Services**" section, click "**Instances and Subscriptions**" to view the Instances and Subscriptions page.

Click on the "**1 key**" link on the **api** Instance previously created:



A dialog will be shown containing the authentication details to use for the API:



From here, **copy and save** the information from the following fields - these will be used in the next step:

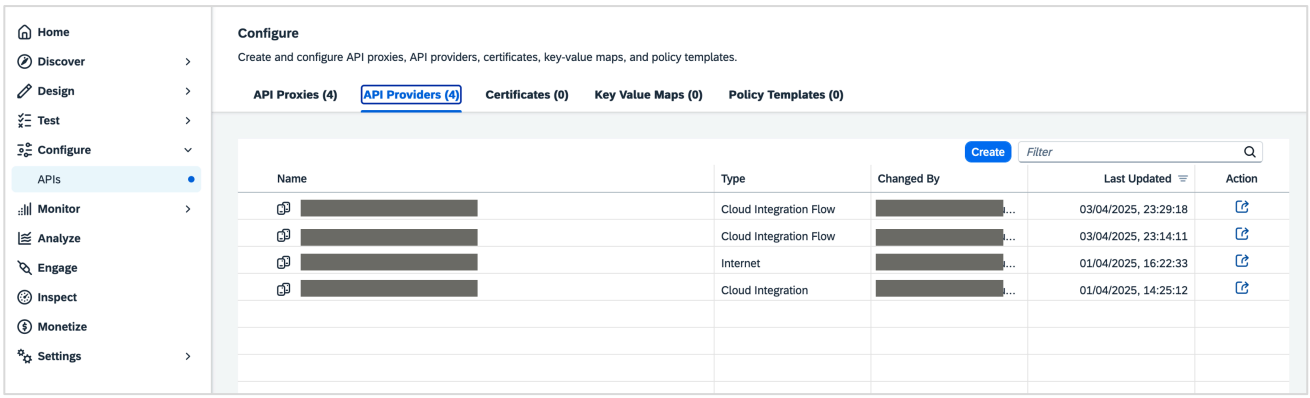
- **clientid**
- **clientsecret**
- **url**
- **tokenurl**

6.3.2. Create API Provider

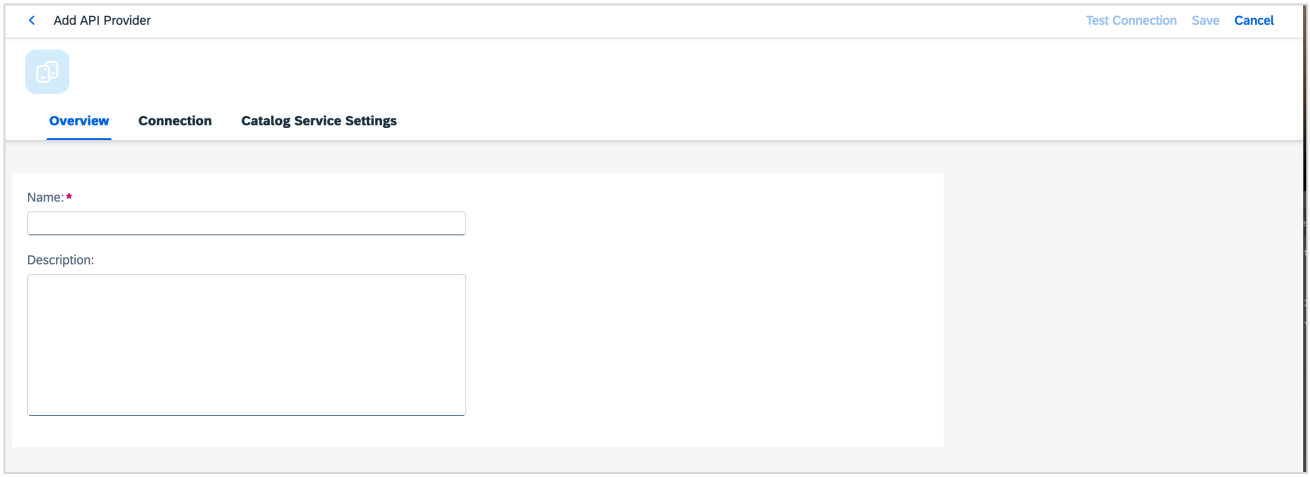
Return to the Integration Suite.

From the navigation menu, click "**Configure",** then "**APIs**". The Configure page will display.

Click the "**API Providers**" tab to load the list of existing API providers:

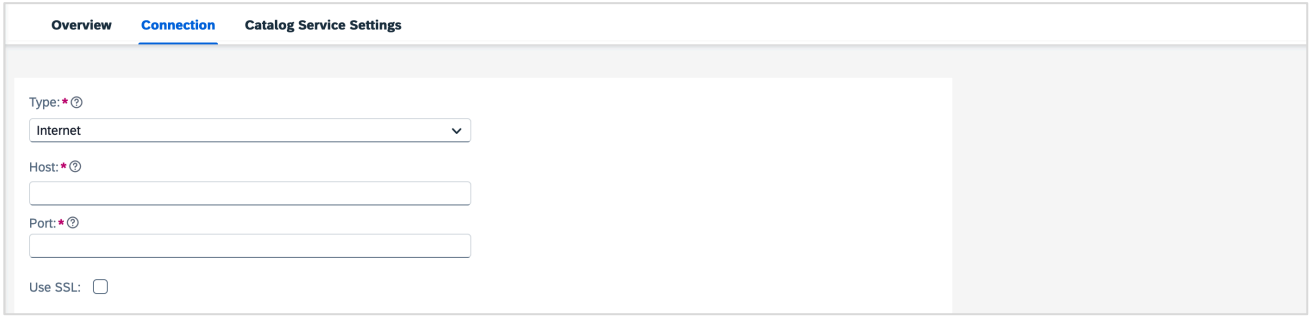


Click the "**Create**" button. The Add API Provider page will display:



In the "**Name**" field enter a name for the API Provider. This must only feature alpha-numeric characters, underscores and dashes (e.g. "QAPI_Provider_Example").

Click the "**Connection**" tab:



Fill in the form:

- In the "**Type**" field, choose "**Cloud Integration**"

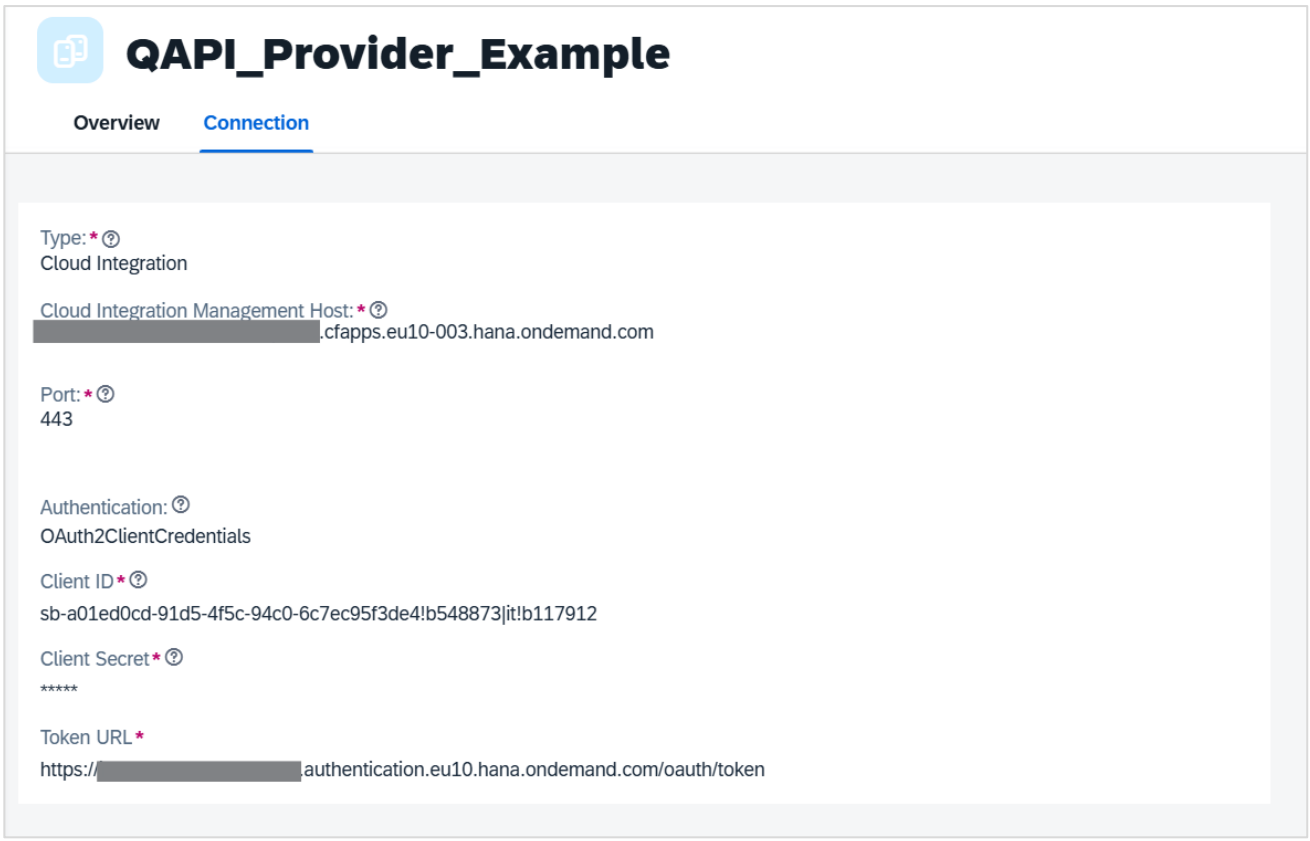
The fields displayed will be modified

- In the "**Cloud Integration Management Host**" field, enter the saved **url** value without the "https://" prefix
- In the "**Authentication**" field, choose "**OAuth2ClientCredentials**"
- In the "**Client ID**" field, enter the saved **clientid** value
- In the "**Client Secret**" field, enter the saved **clientsecret** value
- The "**Token URL**" field should be automatically filled in. If not, enter the saved **tokenurl** value

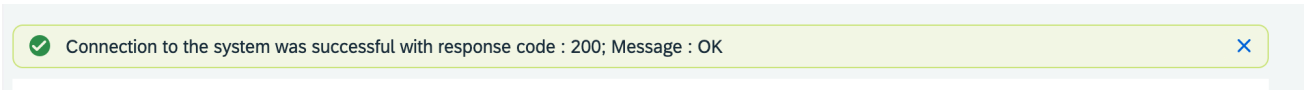
Example configuration:

Click "**Save**" in the top right.

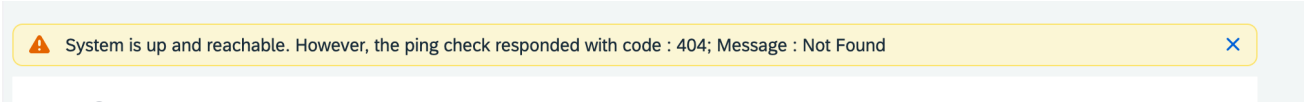
A summary will be shown:



Click "**Test Connection**" in the top bar to ensure successful response code 200 is received:



NOTE: You **might** get the following message:



If so, click "**Edit**" in the top bar, re-enter the client secret, click "**Save**" and try again.

6.4. Create API Proxy and Generate OAuth

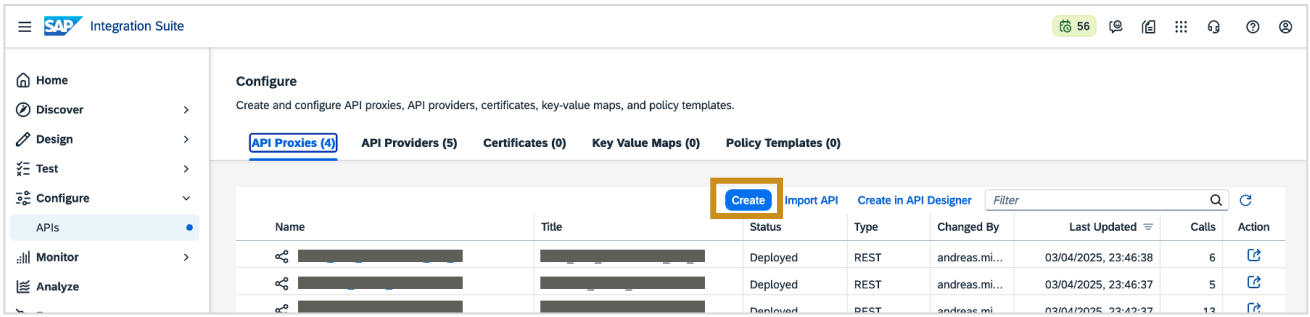
The first Proxy will be created for the purpose of generating an OAuth Token.

6.4.1. Create API Proxy

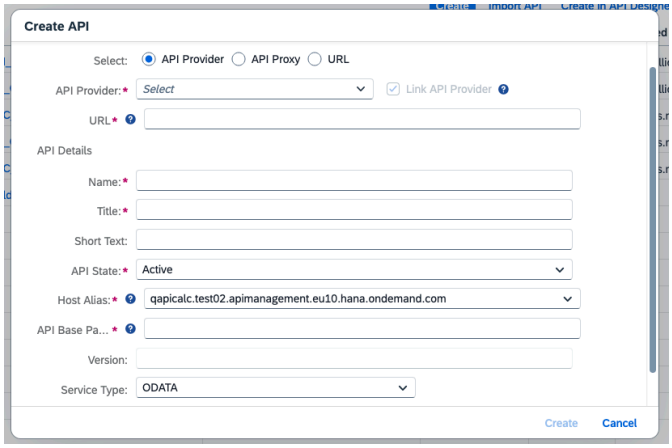
Load the Integration Suite.

From the navigation menu, select "**Configure",** then **"APIs"**. The Configure page will display.

From the "**API Proxies**" tab, click the "**Create**" button at the top of the list:



The Create API dialog will display:



Fill in the form:

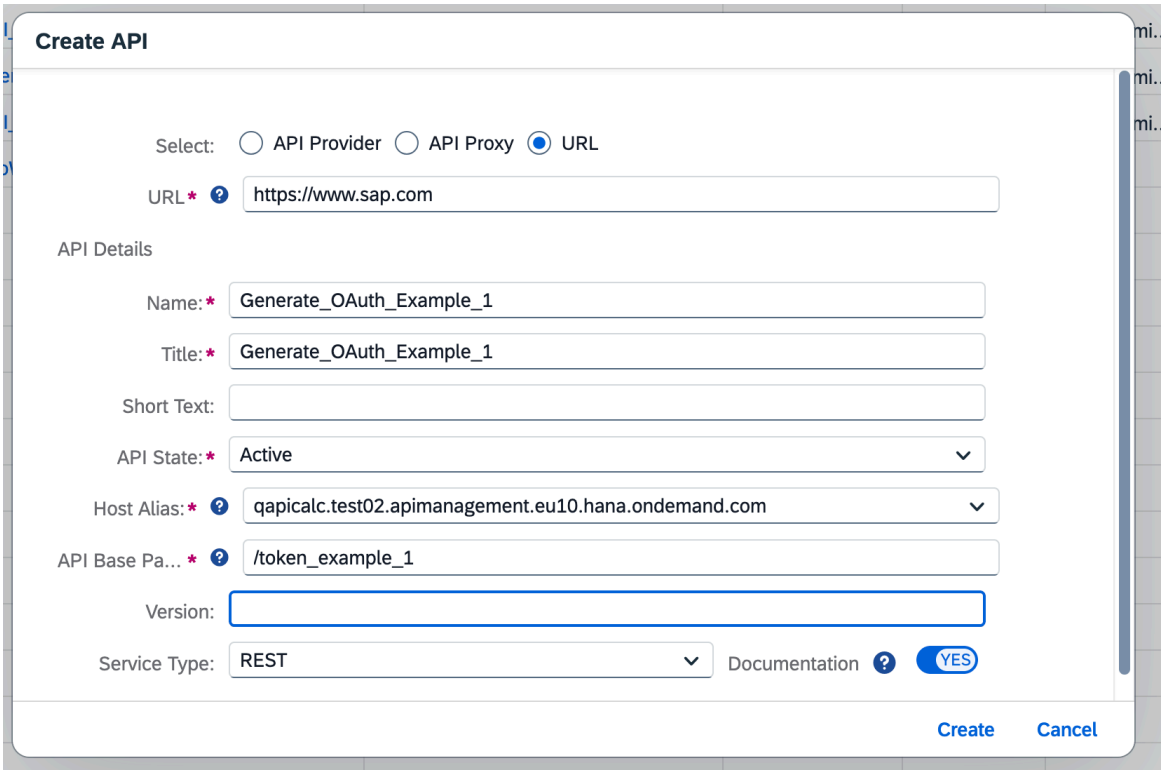
- In the "**Select**" field, choose "**URL**"

  The fields shown will be modified

- In the "**URL**" field, enter "**https://www.sap.com**"

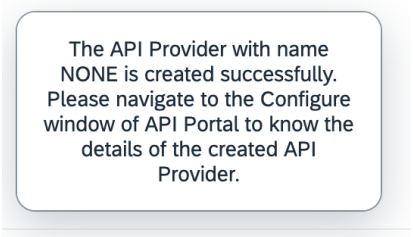
- In the "**Name**" field, enter the internal name you wish to use for the Oauth generator API (e.g. "Generate_OAuth_Example_1")
- In the "**Title**" field, enter the display name you wish to use (this can be the same as for "Name")
- Optionally, change the "**Host Alias**" from the default value selected
- In the "**API Base Path**" field, enter the path for the token URL (e.g. "/token_example_1")

Example configuration:

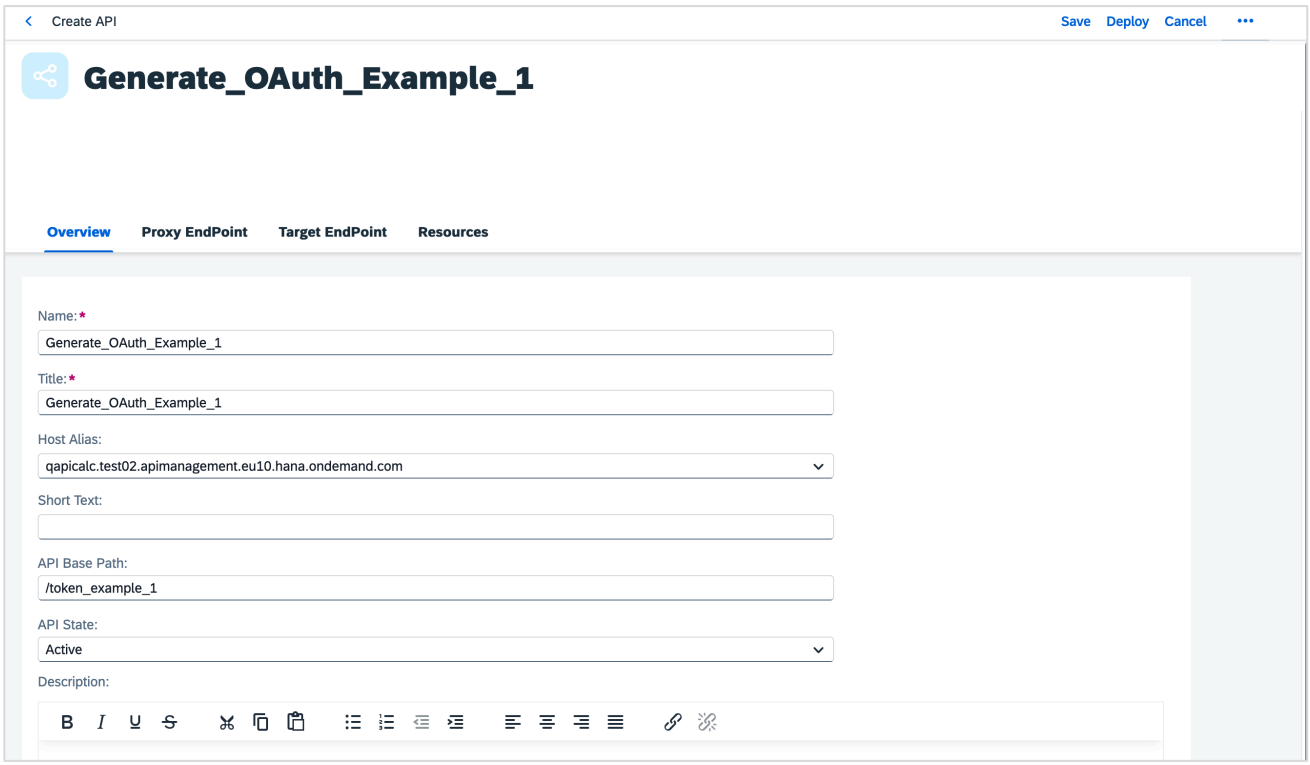


Click "**Create**".

Once done, a confirmation dialog will appear (ignore the "NONE", it has saved correctly):



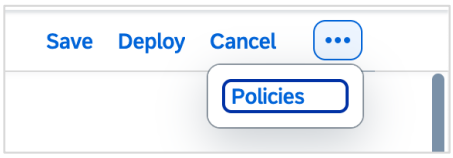
The API will be loaded in the API Editor:



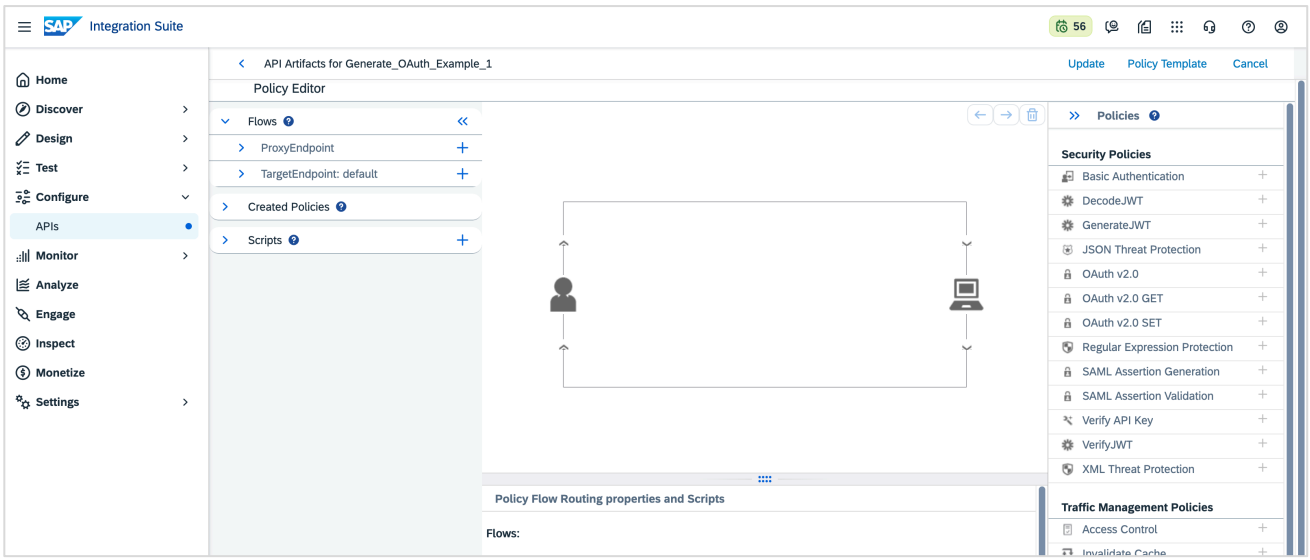
6.4.2. Define Access Policy

NOTE: The set up described in this section is using a "lightweight" security configuration– you may wish to create a more detailed security policy as required.

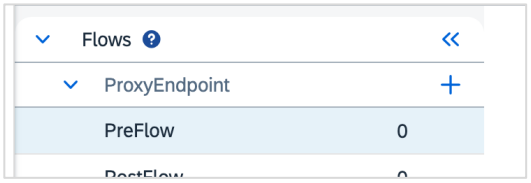
In the top bar to the right, click the "three dots" button and click "**Policies**":



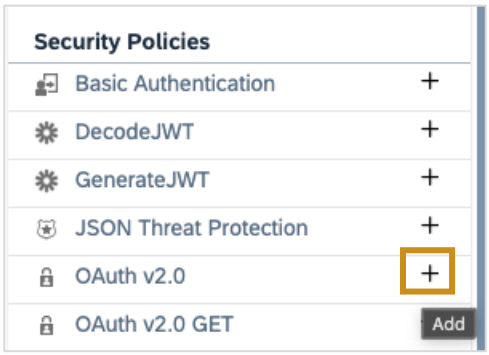
The "Policy Editor" will display, containing a default set up:



In the "**Flows**" menu to the left, select "**ProxyEndpoint**", then "**PreFlow**":

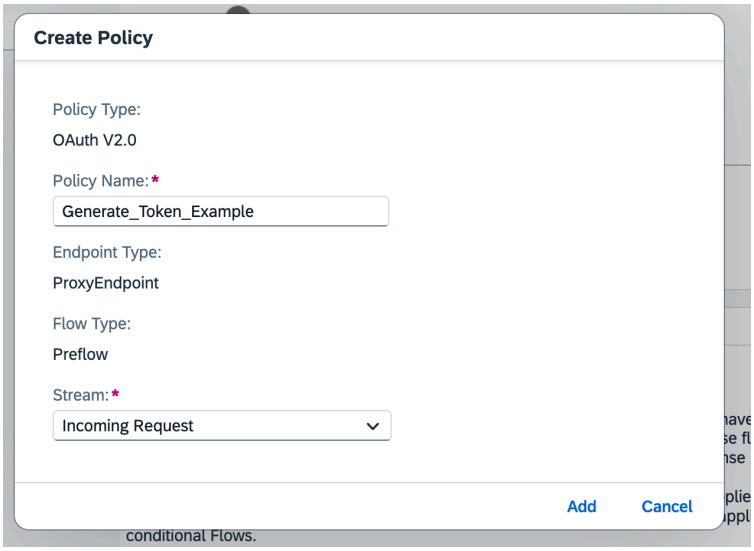


In the "**Security Policies**" menu to the right, select the plus to the right of "**OAuth v2.0**":

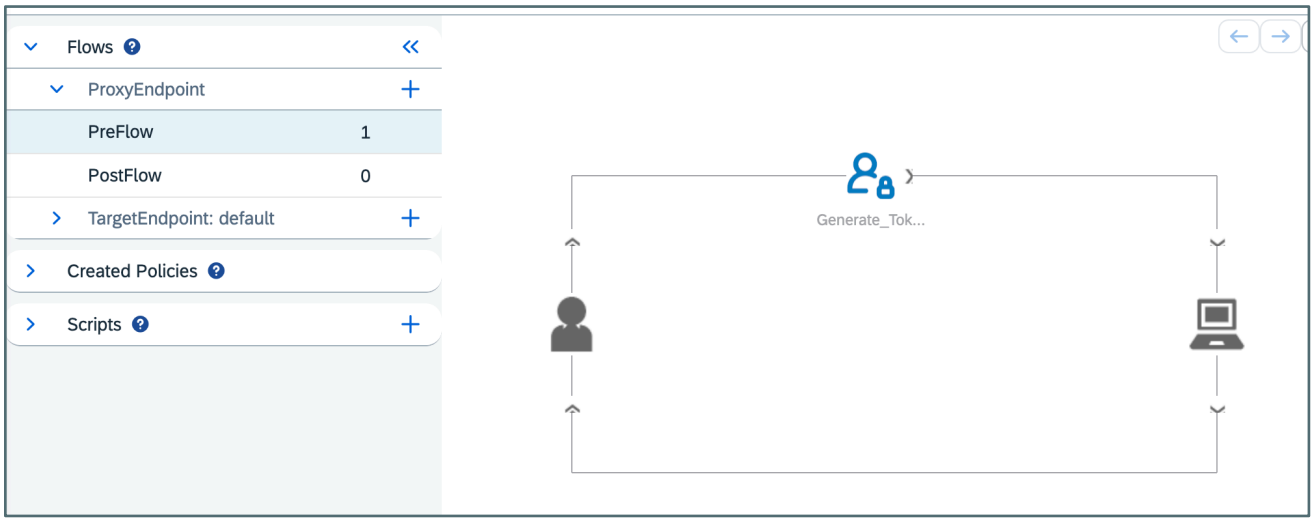


The Create Policy dialog will be displayed.

In the "**Policy Name**" field, enter a name for the policy that describes its purpose (e.g. "Generate_Token_Example"):



Click "**Add**". The new policy will be added to the diagram:



Below the diagram, XML code will be shown, which configures the policy:

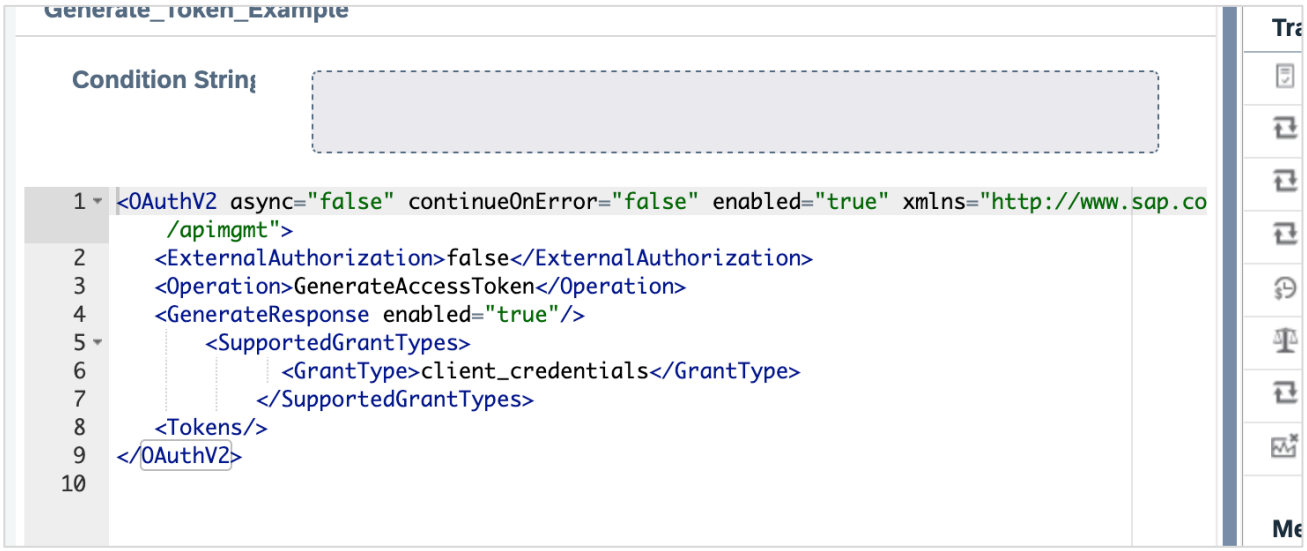
```
<OAuthV2 async="false" continueOnError="false" enabled="true" xmlns="http://www.sap.co
    /apimgmt">
    <!-- By default, VerifyAccessToken expects the access token to be sent in an
        Authorization header. You can change that default using this element<AccessToke
        > -->
    <!-- If you want to pass access token in an customer header "access_token": -->
    <!-- <AccessToken>request.header.access_token</AccessToken> -->
    <!-- If you want to pass access token in query param "access_token": -->
    <!-- <AccessToken>request.queryparam.access_token</AccessToken> -->
    <!-- this flag has to be set when you want to work with third-party access tokens
        -->
    <ExternalAuthorization>false</ExternalAuthorization>
    <!-- valid values are GenerateAccessToken, GenerateAccessTokenImplicitGrant,
        GenerateAuthorizationCode ,
     RefreshAccessToken , VerifyAccessToken , InvalidateToken , ValidateToken  -->
    <Operation>VerifyAccessToken</Operation>
    <GenerateResponse enabled="true"/><SupportedGrantTypes/>
    <Tokens/>
</OAuthV2>
```

For the pre-flow, **this needs to be edited** to use “RefreshAccessToken” and include supported grant types.

To do this, **replace** the XML with the following (the updated/new XML is shown in bold):

```
<OAuthV2 async="false" continueOnError="false" enabled="true" xmlns="http://www.sap.com/apimgmt">
   <ExternalAuthorization>false</ExternalAuthorization>
   <Operation>GenerateAccessToken</Operation>
   <GenerateResponse enabled="true"/>
      <SupportedGrantTypes>
         <GrantType>client_credentials</GrantType>
      </SupportedGrantTypes>
   <Tokens/>
</OAuthV2>
```

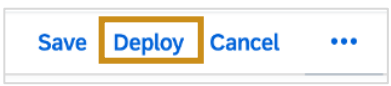
Once done it should look like this:



Click “**Update**” in the top right.

You will be returned to the API Editor.

Click “**Deploy**” in the top right:



The API will be deployed.

6.5. Create API Proxy for iFlow

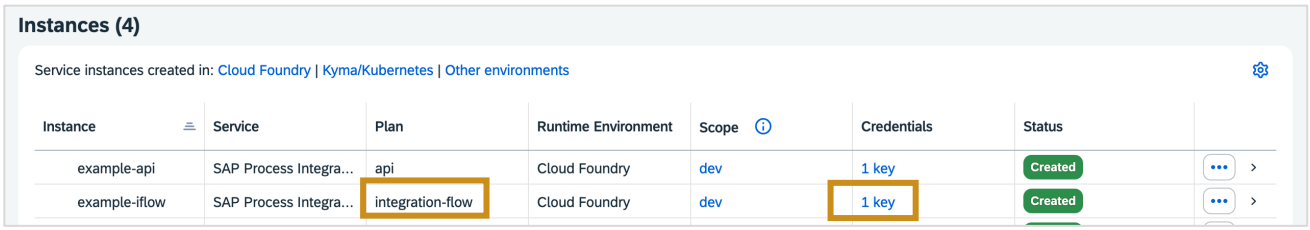
The second API proxy will be based on the iFlow created in Import / Create iFlow and the API Provider created in Create API Provider. It requires that the service key has been created for the iFlow in Create Integration Flow Instance + Keys.

6.5.1. Obtain Credentials

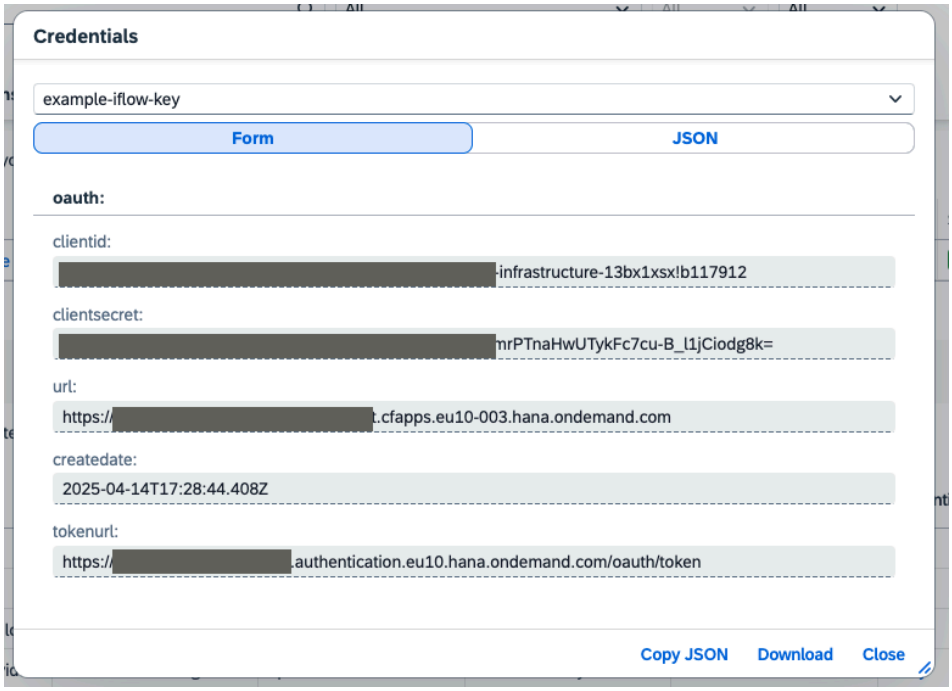
From SAP BTP Cockpit, navigate to the relevant subaccount.

From the navigation menu, under the "**Services**" section, click "**Instances and Subscriptions**" to view the Instances and Subscriptions page.

Click on the "**1 key**" link on the **integration-flow** Instance previously created:



A dialog will be shown containing the authentication details to use for the integration flow:



From here, **copy and save** the information from the following fields - these will be used in the next step:

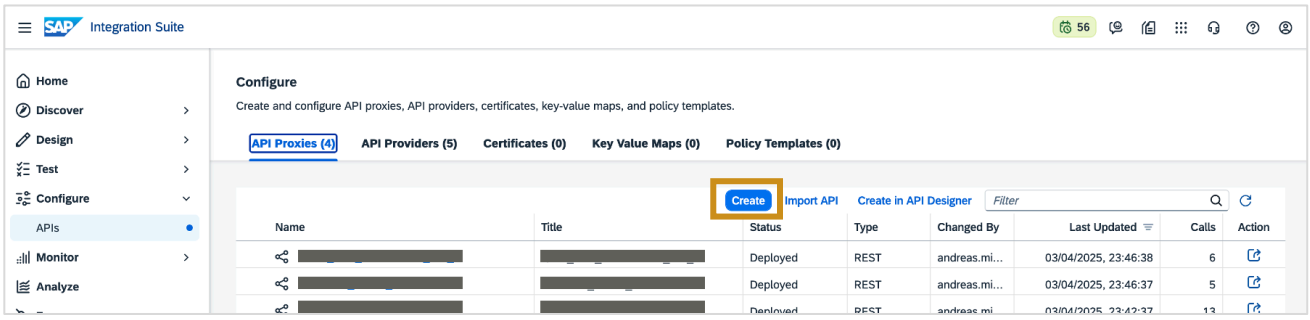
- **clientid**
- **clientsecret**
- **url**
- **tokenurl**

6.5.2. Create API Proxy for iFlow

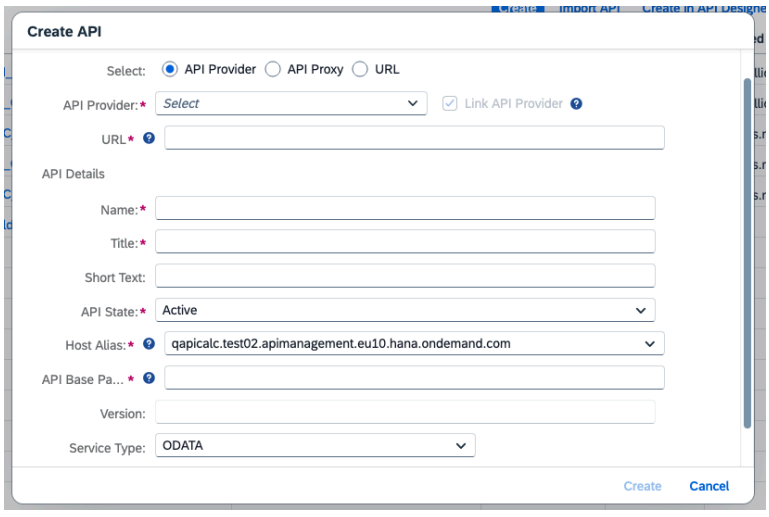
Return to the Integration Suite.

From the navigation menu, click "**Configure",** then "**APIs**". The Configure page will display.

From the "**API Proxies**" tab, click the "**Create**" button at the top of the list:

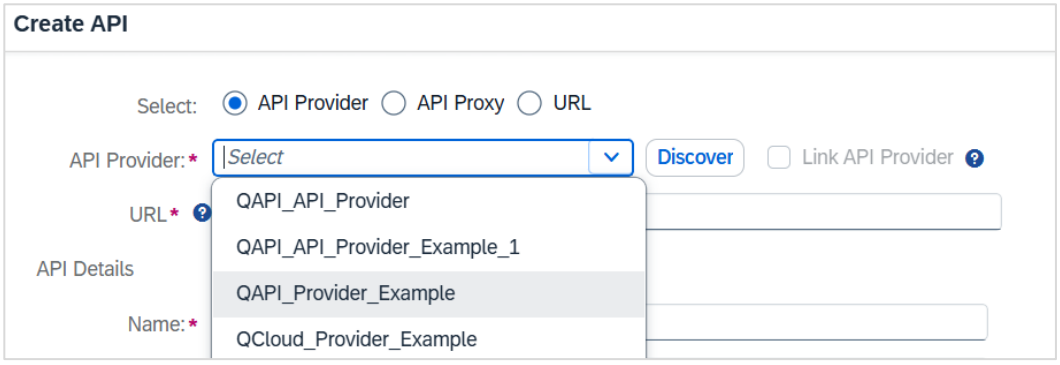


The Create API dialog will display:



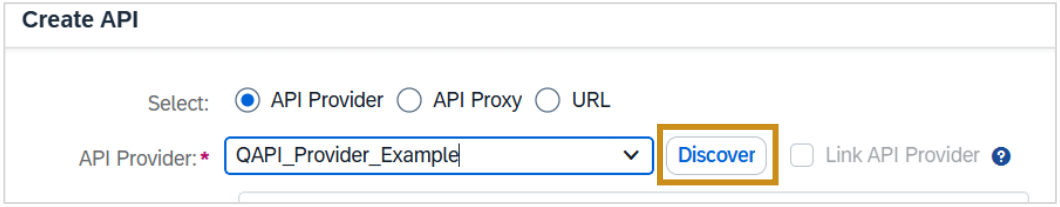
Leave the "**Select**" field as "**API Provider**".

Click "**API Provider**" field to view a list of deployed API providers:

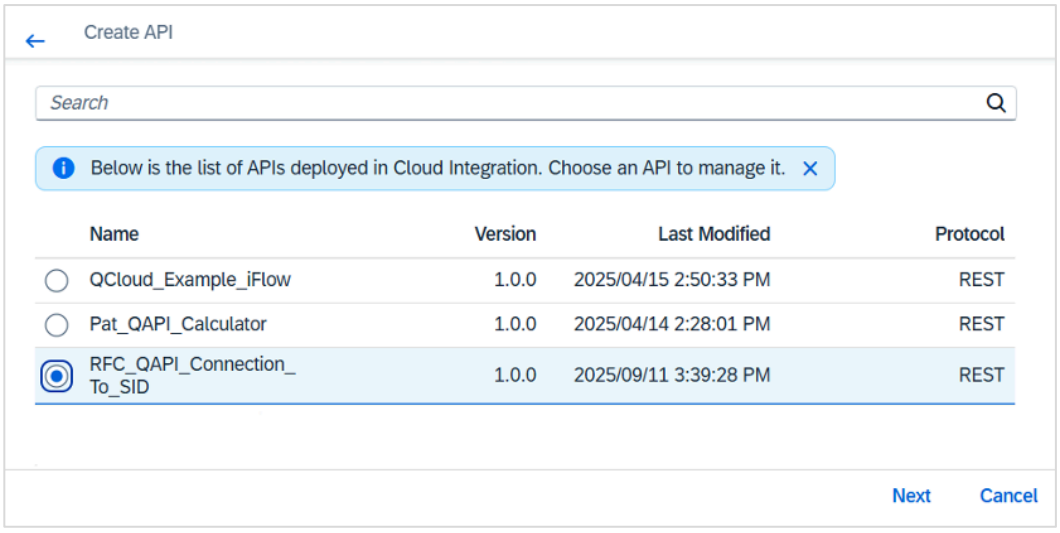


Select the API provider you created in Create API Provider from the list.

Click the "**Discover**" button:



A new panel will be shown listing all the created iFlows:



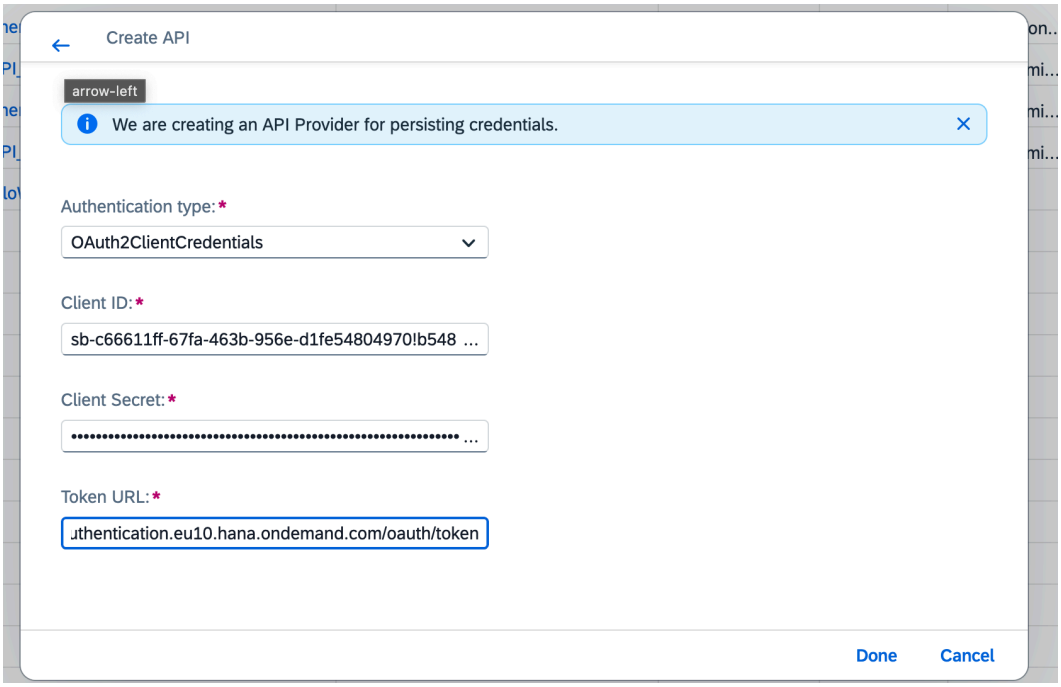
Select the iFlow you created in Import / Create iFlow and click "**Next**".

The authentication panel will display.

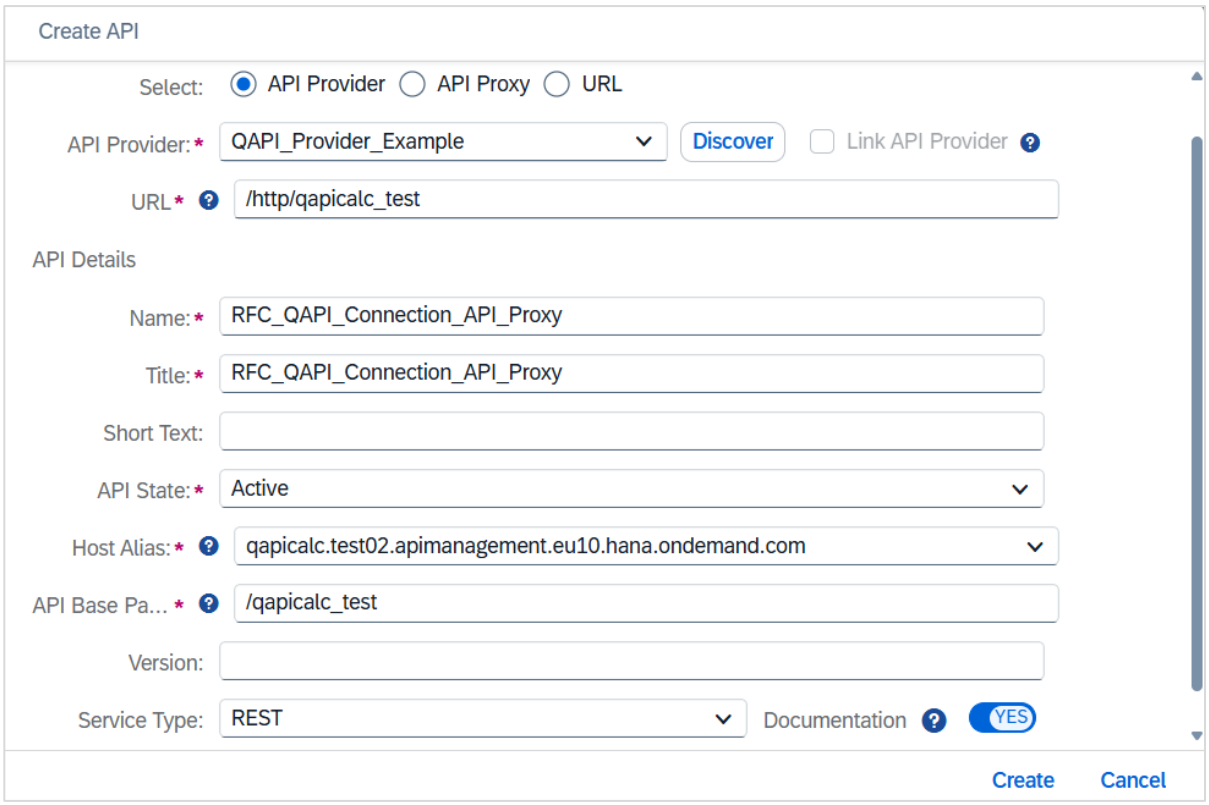
Fill in the form:

- In the "**Authentication Type**" field, select "**OAuth2ClientCredentials**"
- In the "**Client ID**" field, enter the saved **clientid** value
- In the "**Client Secret**" field, enter the saved **clientsecret** value
- The "**Token URL**" field should be automatically filled in. If not, enter the saved **tokenurl** value

Example configuration:



Click "**Done**" to show the summary panel:



Note: By default, the **Name** and **Title** are set as the iFlow name "**RFC QAPI Connection To SID"**.

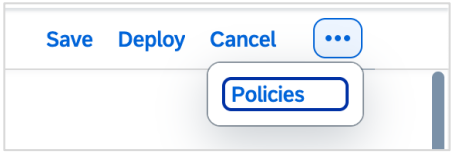
Modify the default "**Name**" and "**Title**" fields if required, as in the example above.

Click "**Create**".

The panel will close and the created iFlow API will be displayed.

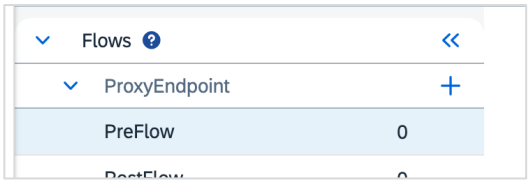
6.5.3. Define Access Policy

In the top bar to the right, click the "three dots" button and click "**Policies**":



The Policy Editor will display:

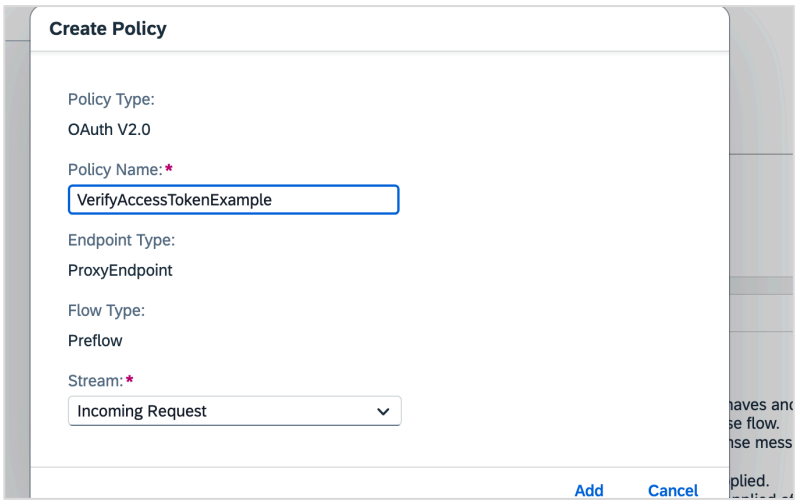
In the "**Flows**" menu to the left, select "**ProxyEndpoint**", then "**PreFlow**":



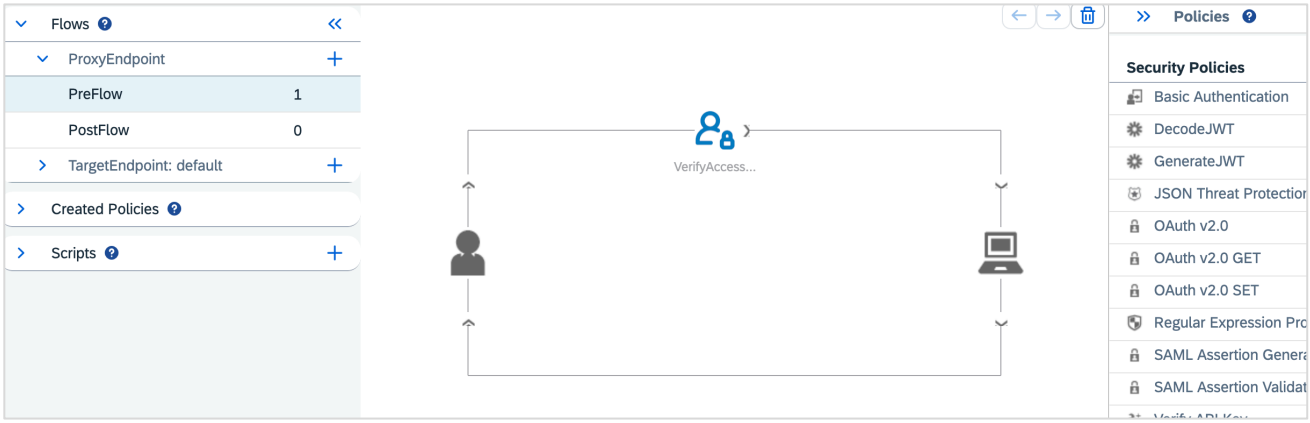
In the "**Security Policies**" menu to the right, select the plus to the right of "**OAuth v2.0**":

The Create Policy dialog will be displayed.

In the "**Policy Name**" field, enter a name for the policy that describes its purpose (e.g. "VerifyAccessTokenPolicy"):



Click "**Add**". The new policy will be added to the diagram:



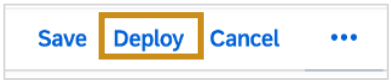
Below the diagram, XML code will be shown, which configures the policy.

"VerifyAccessToken" is the default Operation in the "OAuth v2" security Policy. As such, **you do not need to make any changes to the XML**.

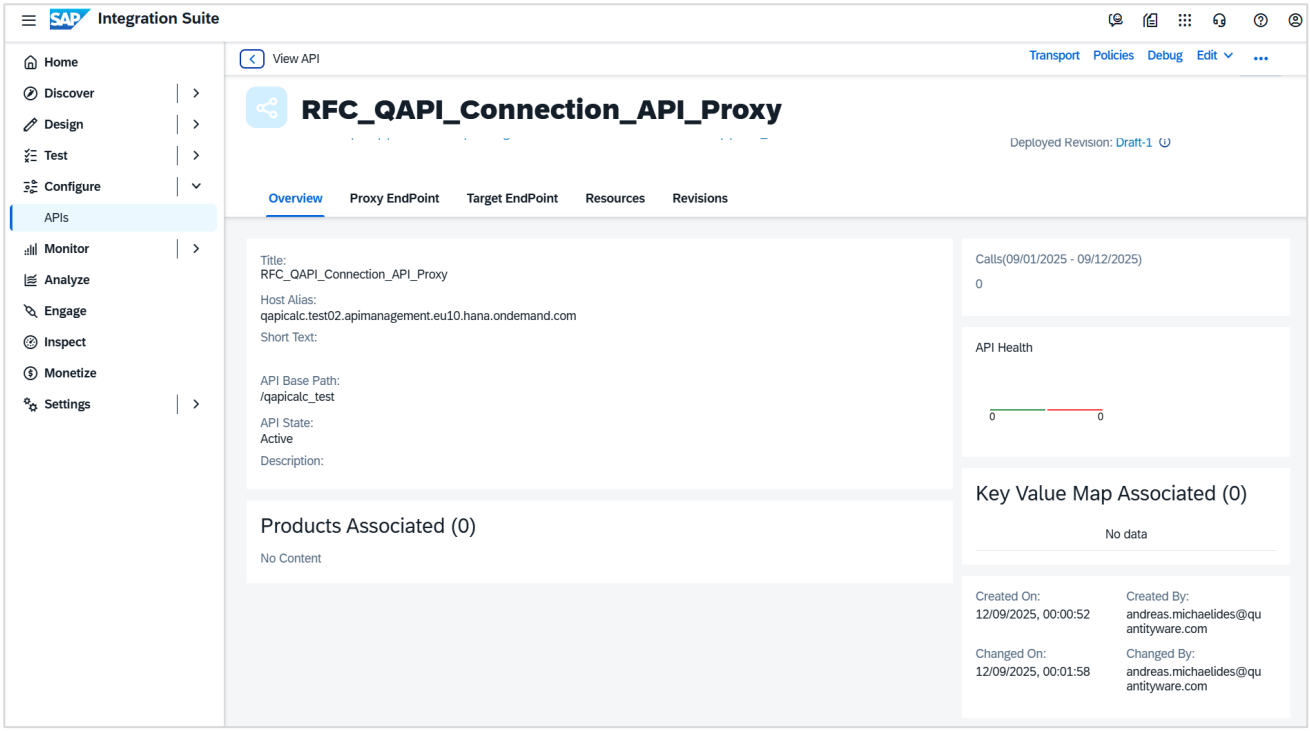
Click "**Update**" in the top right.

You will be returned to the API Editor.

Click "**Deploy**" in the top right:



The API will be deployed:

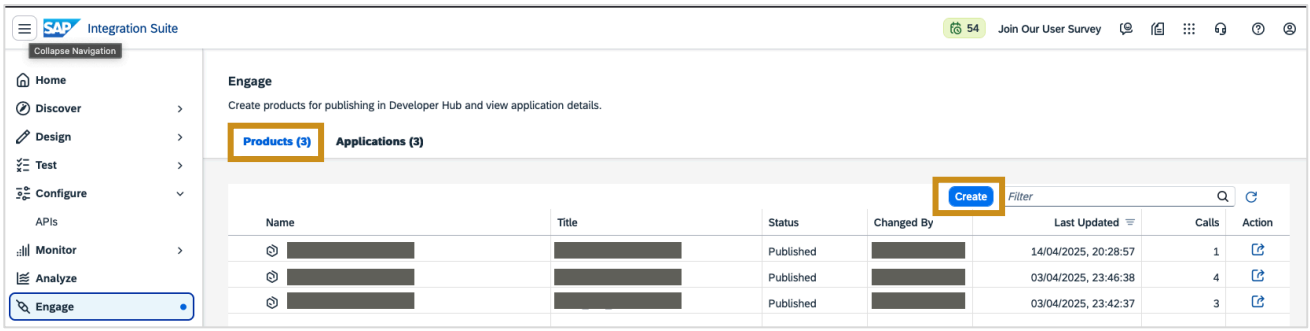

Integration Suite
Home
Discover
Design
Test
Configure
APIs
Monitor
Analyze
Engage
Inspect
Monetize
Settings
View API
Transport
Policies
Debug
Edit
RFC_QAPI_Connection_API_Proxy
Deployed Revision: Draft-1
Overview
Proxy EndPoint
Target EndPoint
Resources
Revisions
Title:
RFC_QAPI_Connection_API_Proxy
Host Alias:
qapicalc.test02.apimanagement.eu10.hana.ondemand.com
Short Text:
API Base Path:
/qapicalc_test
API State:
Active
Description:
Products Associated (0)
No Content
Calls(09/01/2025 - 09/12/2025)
0
API Health
0
0
Key Value Map Associated (0)
No data
Created On:
12/09/2025, 00:00:52
Created By:
andreas.michaelides@quantityware.com
Changed On:
12/09/2025, 00:01:58
Changed By:
andreas.michaelides@quantityware.com

7. Create a Product

We now need to create a Product that includes our two APIs.

From within the Integration Suite, in the navigation menu, click "**Engage**". The Engage page will display.

From the "**Products**" tab, click "**Create**":

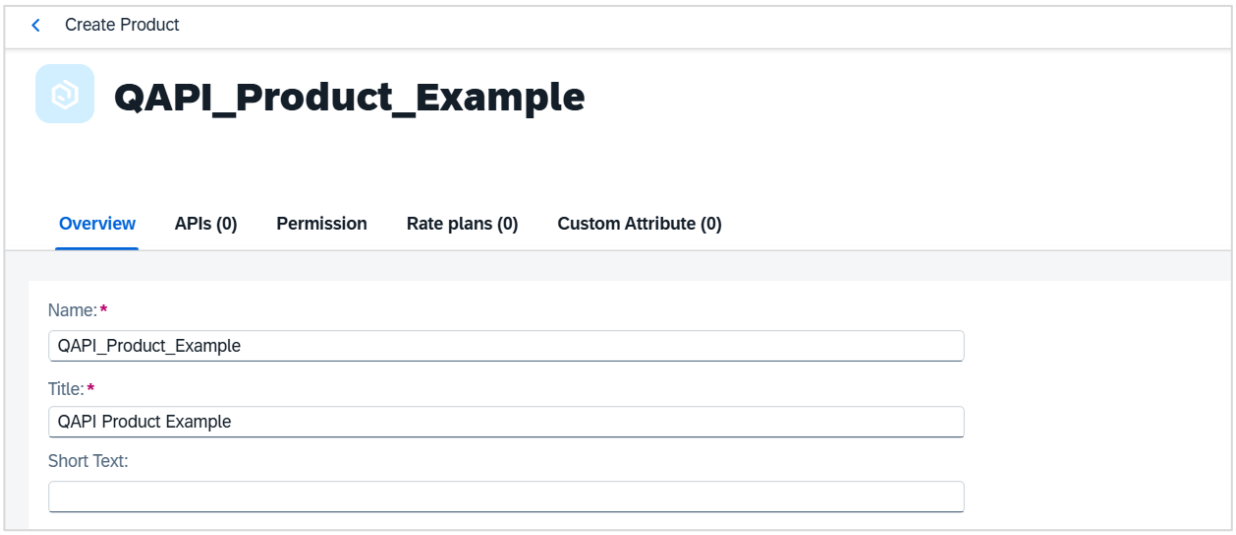


The Create Product page will display.

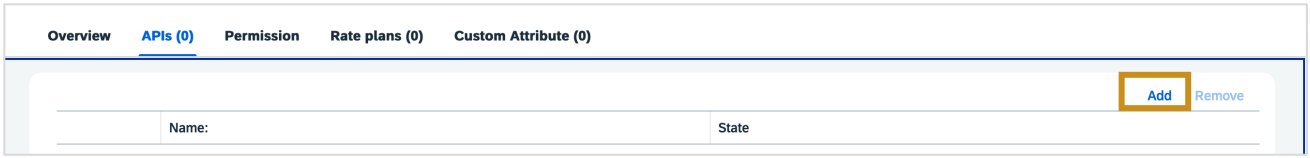
Fill in the form:

- In the "**Name**" field, enter the internal name of the product [not about chars here pat]
- In the "**Title**" field, enter the display name for the product
- All other fields can be left blank

Example configuration:

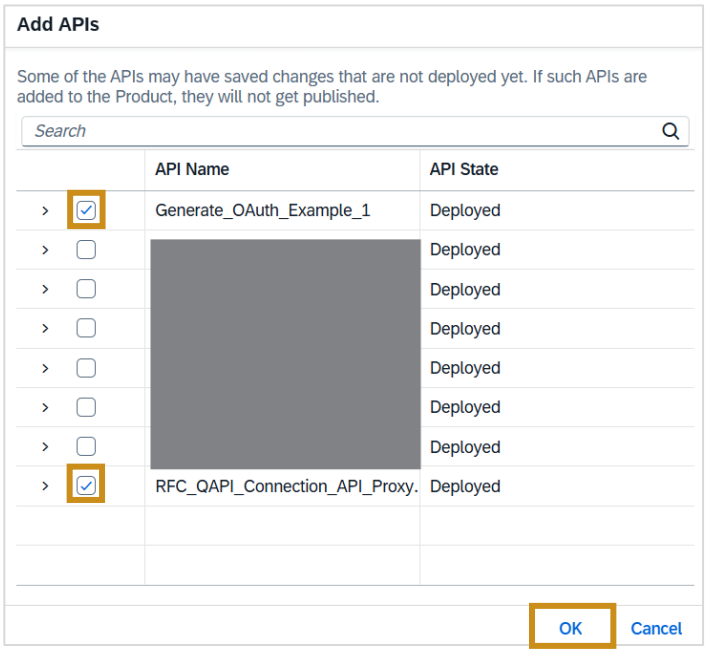


Click on the "**APIs**" tab, then click "**Add**":



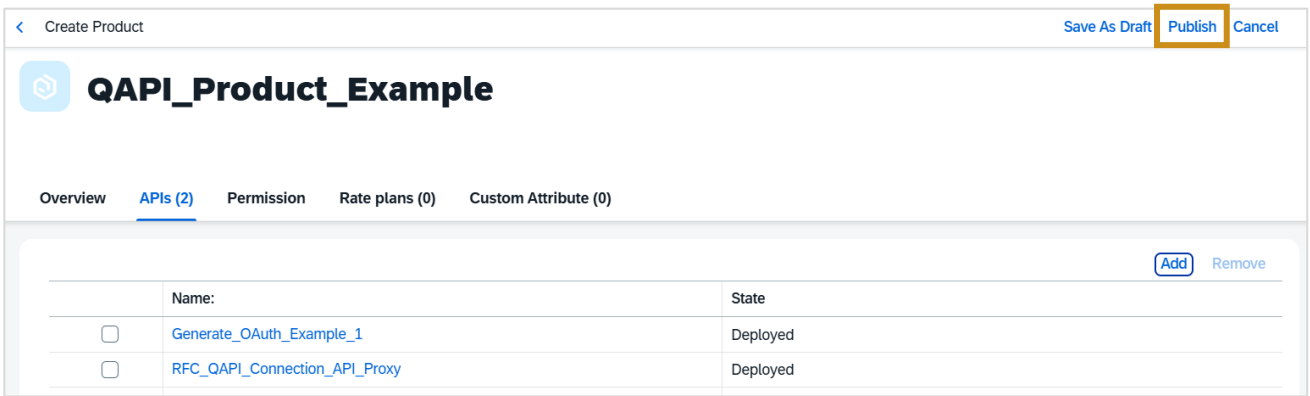
The Add APIs dialog will display.

Select the two API Proxies previously created, and click "**OK**" to save:

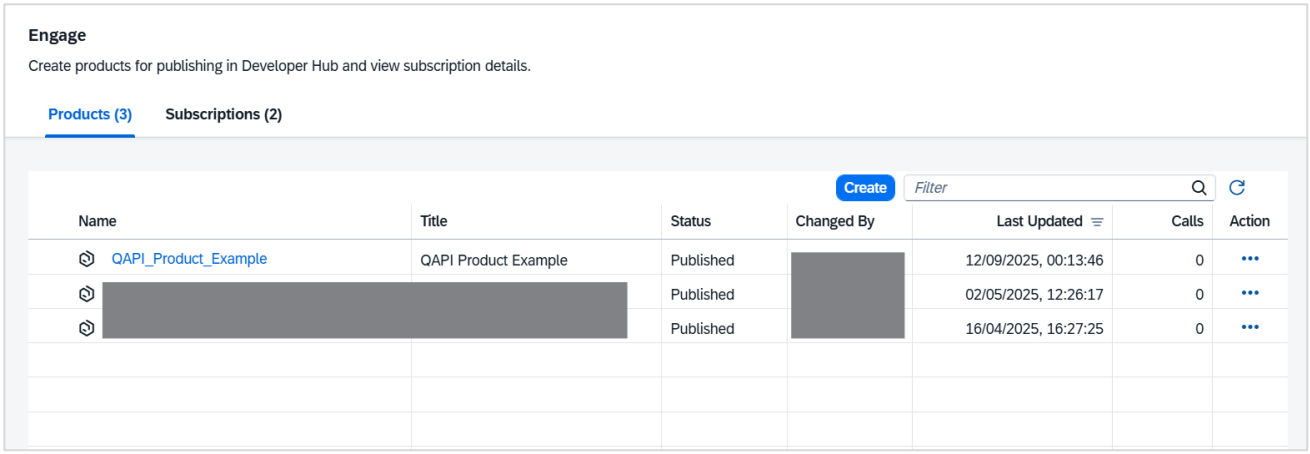


You will see the two APIs in the APIs list.

Click "**Publish**" in the top right:



The product will be published and shown in the Products list:

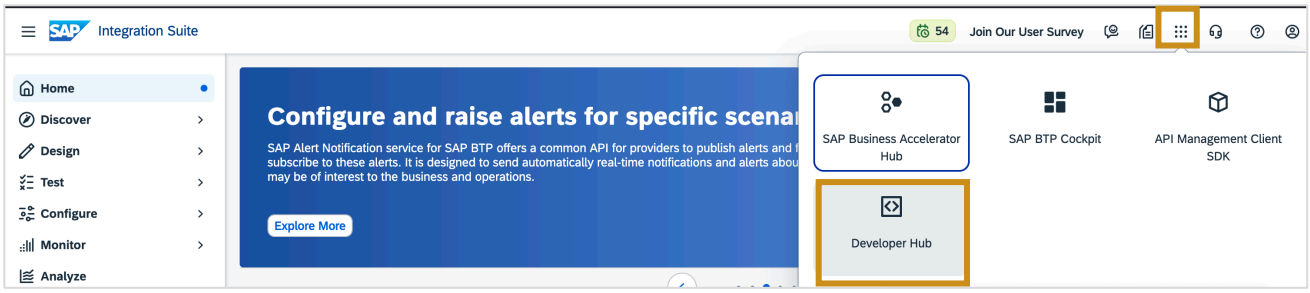


8. Create the Application

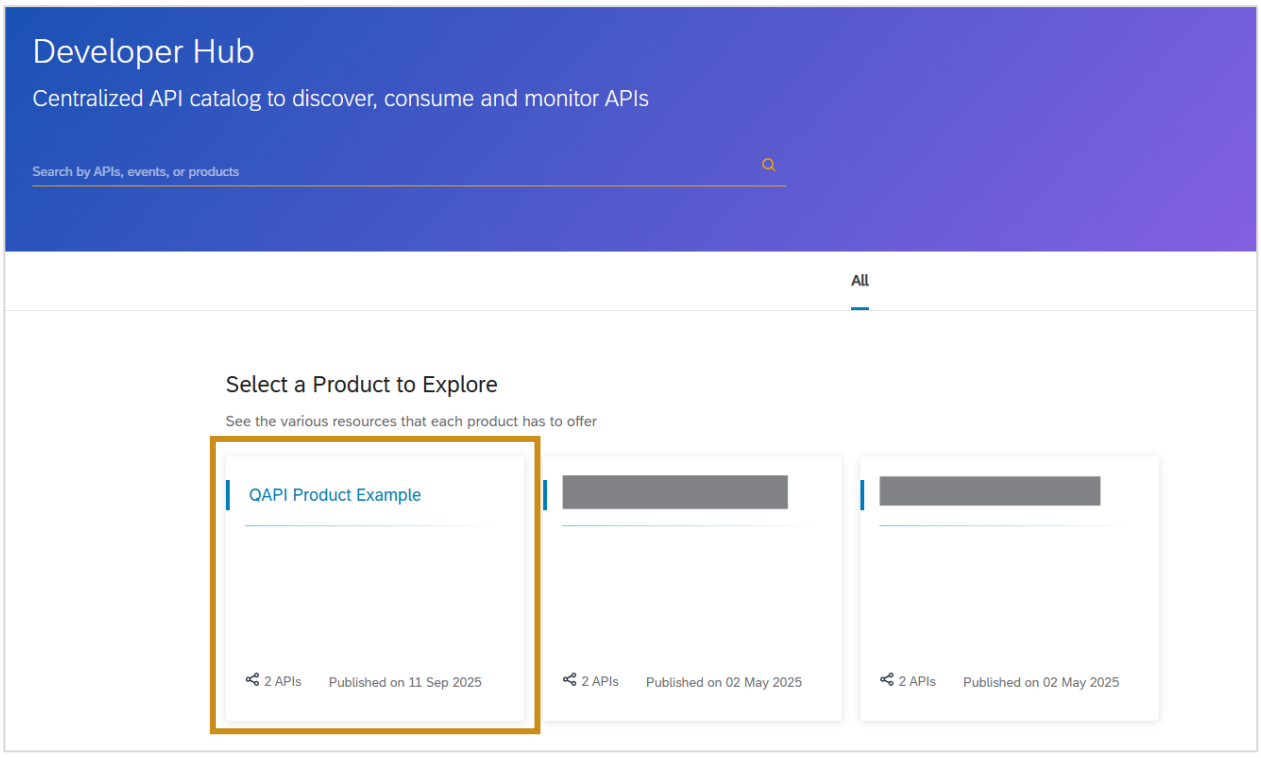
Finally, we create a new Application for the product you created in the previous section and generate API keys, which can be used to generate an OAuth token to securely access the API.

Load the Integration Suite.

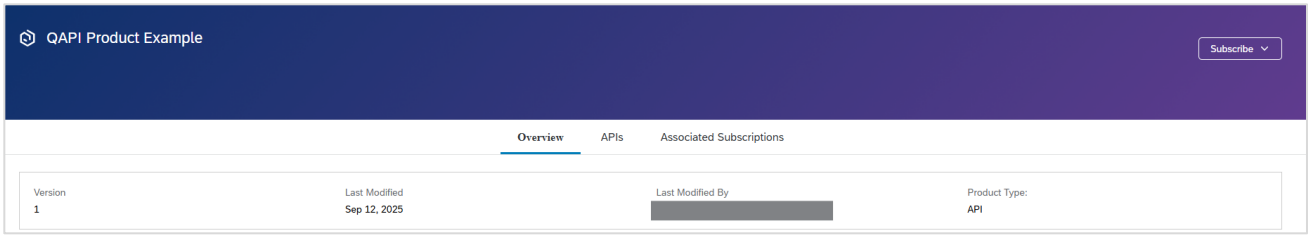
Click the grid icon in the top right of the page, and select "**Developer Hub**":



The Developer Hub will load in a new browser tab, displaying a list of products, including the one you have created:



Select the Product you created in the previous section to load its Overview page:



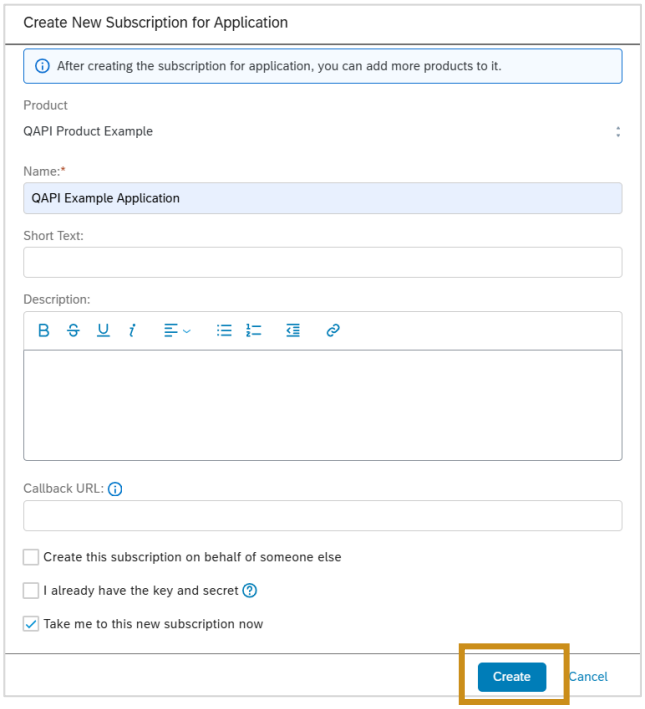
Click on the "**Subscribe**" button in the top right, then select "**Create New Subscription for Application**":

The Create New Application dialog will be displayed.

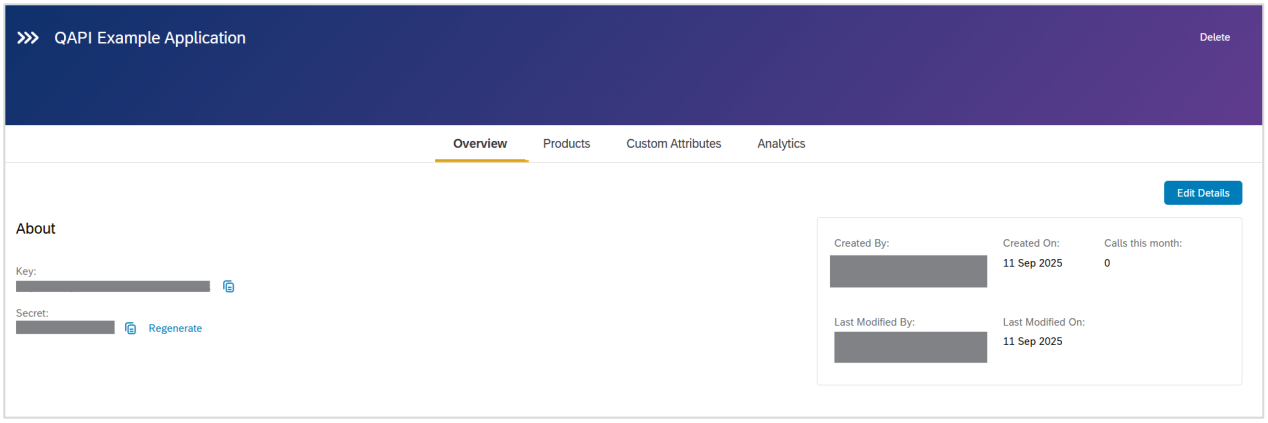
Fill in the Form:

- In the "**Name**" field, enter your application's title (e.g. "**QAPI Example Application**")
- Optionally fill in any other fields you require

Click "**Create**":



The application will be shown:



8.1. Obtain Key Info for Access

We need to obtain the following information to access QAPI:

1. **Access Token URL** (for obtaining the access token)
2. **Client Secret** (for obtaining the access token)
3. **Client Key** (for obtaining the access token)
4. **API Proxy URL** (for interacting with QAPI using the access token)

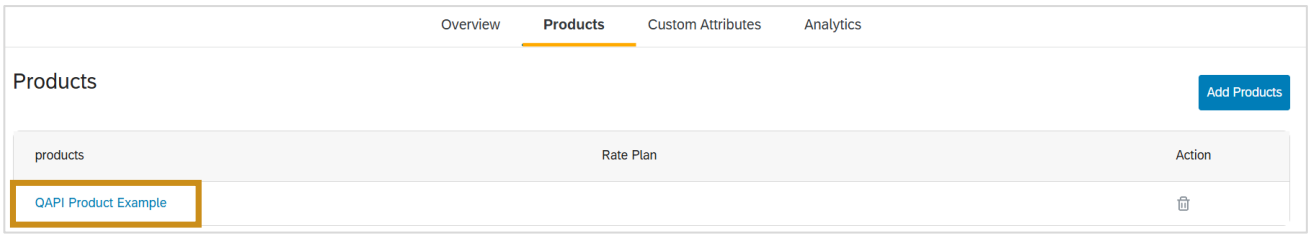
8.1.1. Application Secret + Key

From the application page (shown above), within the About section:

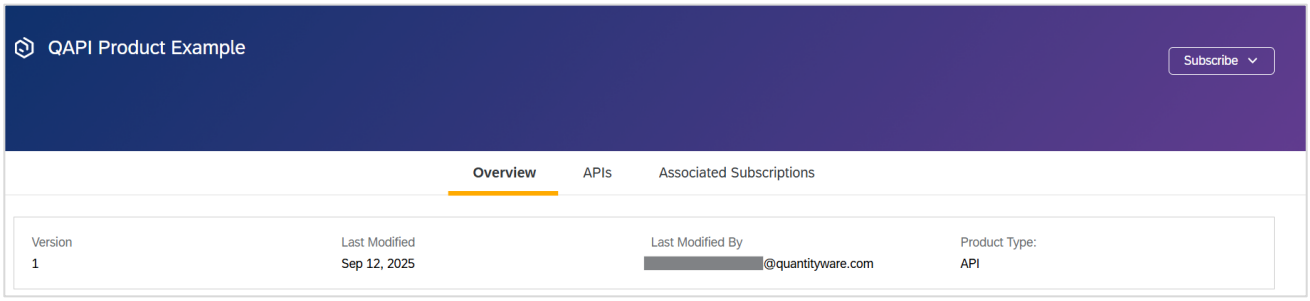
- Copy the "**Application Secret**" as your **Client Secret (2)**
- Copy the "**Application Key**" as your **Client Key (3)**

8.1.2. Access Token URL

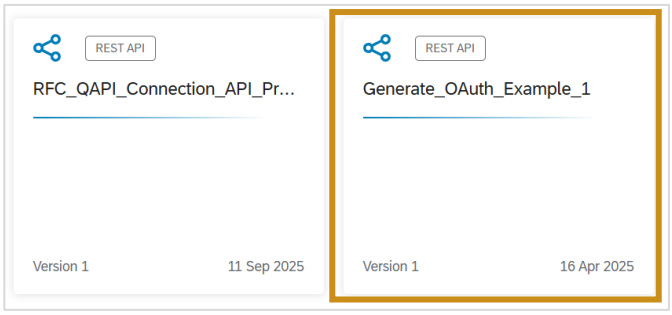
Click the "**Products**" tab:



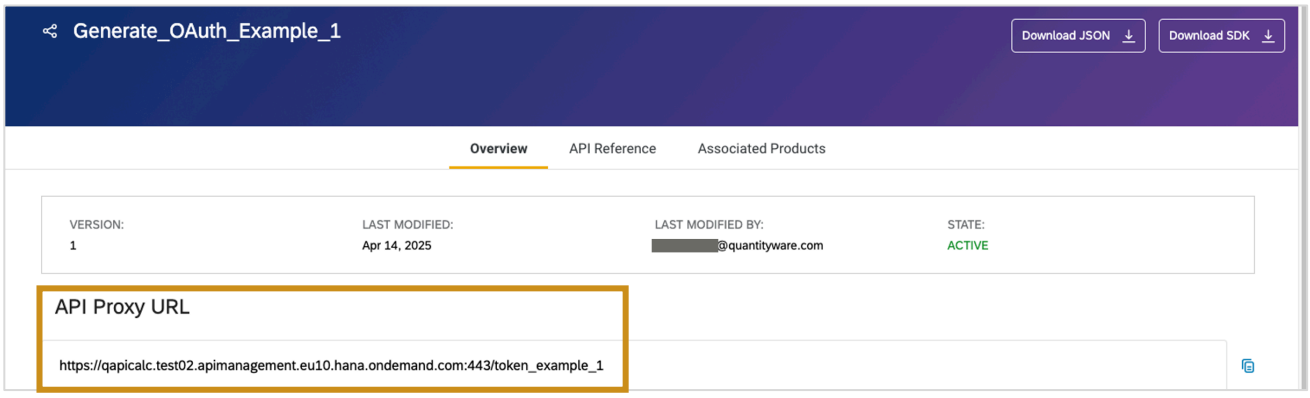
Select the product you created. The Overview page will be displayed:



Click the "**APIs**" tab. The two APIs you created will be displayed:



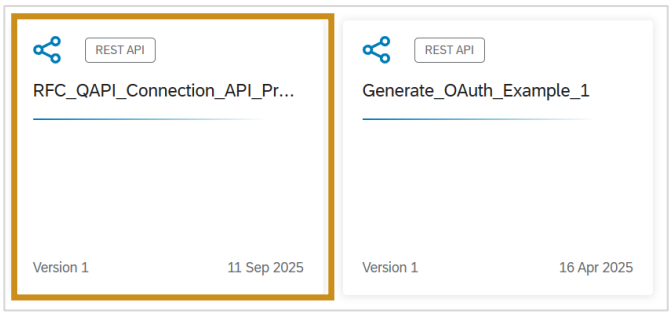
Click the "**Generate_OAuth_Example_1 API**" title. The API's overview page will display:



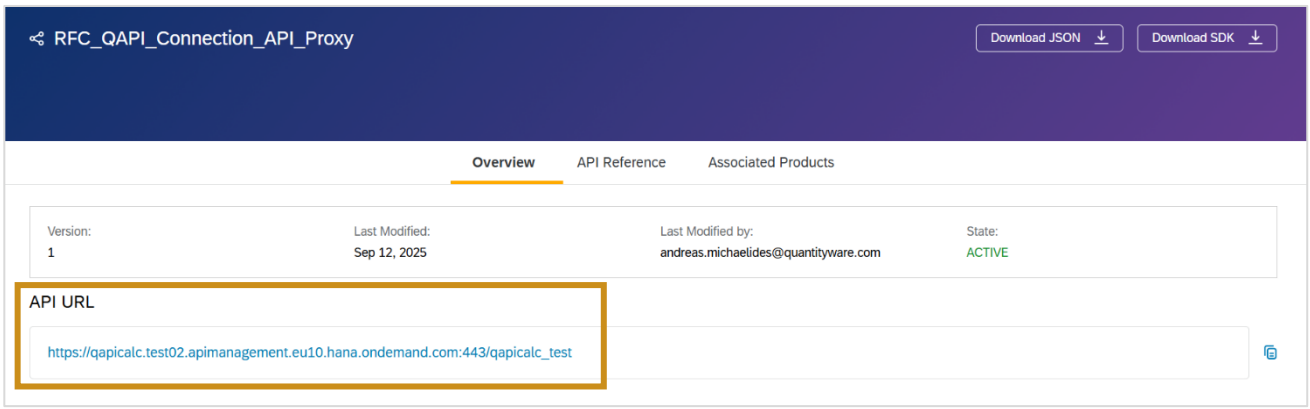
Copy the "**API Proxy URL**" as your **Access Token URL (1)**.

8.1.3. API Proxy URL

Click “back” in your web browser to return to the product page, and click the “**APIs**” tab again to view the two APIs you created:



Click the “**RFC_QAPI_Connection_API_Proxy**” tile. The iFlow API’s overview page will display:



Copy the “**API URL**” as your **API Proxy URL (4)**.

9. Accessing QAPI

This section of the document describes how QAPI can be accessed using Postman, with an additional example in PHP.

9.1. Prerequisites

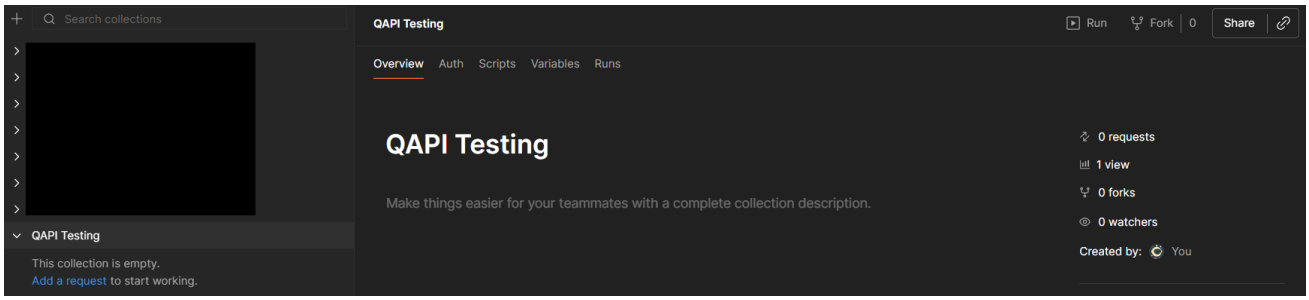
- Postman has been installed, and you are familiar with the software.
- The information for accessing the QAPI product have been obtained as per Obtain Key Info for Access.

9.2. Create Collection

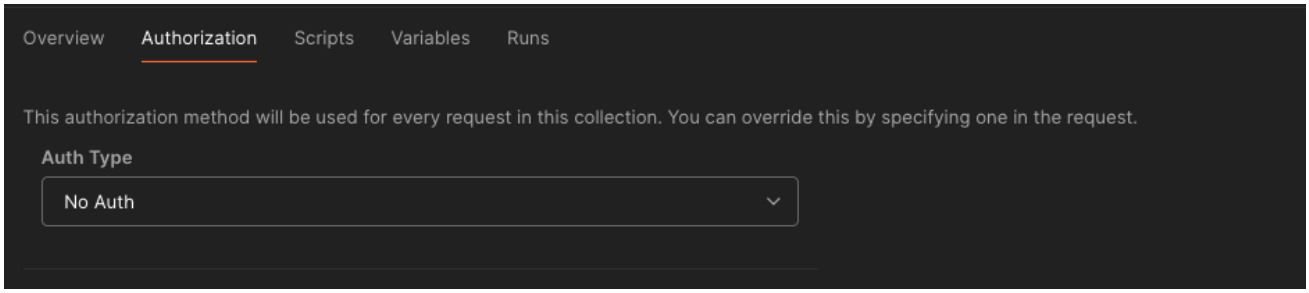
From within Postman, create a new blank collection and give a name, e.g. “QuantityCloud Testing”.

9.3. Create “Get Token” Request

Select the new collection. The “Overview” will be shown:



Click the “**Authorization**” tab:



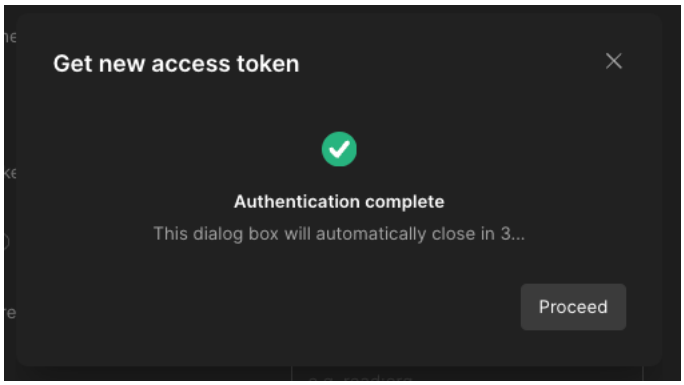
Change the “**Auth Type**” field to “**OAuth 2.0**”. The form will be reloaded.

Fill in the form:

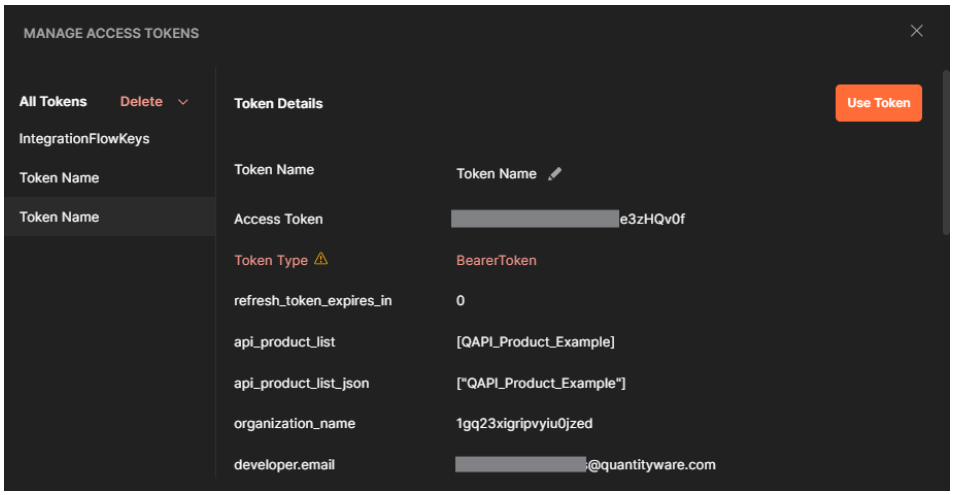
- Change the "**Grant Type**" field to "**Client Credentials**"
- In the "**Access Token URL**" field, enter your **Access Token URL** (Section 8.1.2)
- In the "**Client ID**" field, enter your **Client Key** (Section 8.1.1)
- In the "**Client Secret**" field, enter your **Client Secret** (Section 8.1.1)
- Leave all other fields as default

Scroll to the bottom of the form and click the "**Get New Access Token**" button.

If successful, you will see the following message:



After this is closed, you will see the Manage Access Tokens dialog:



Click "**Use Token**" to make the token available to all requests within this collection.

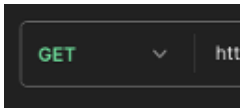
Save the Collection.

9.4. Create QAPI Action Request

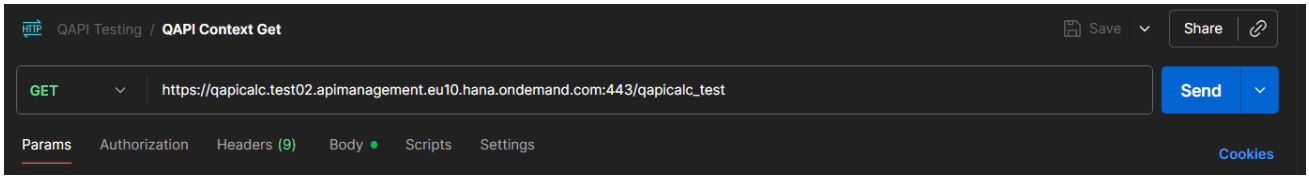
Add a new Request within the collection.

Set the name accordingly (e.g. “QAPI Context Get”)

Ensure that the request type is set to “**GET**”:

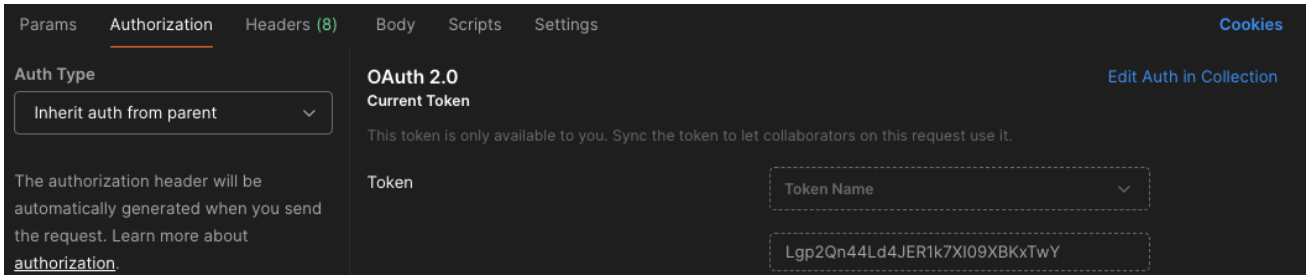


Enter the **API Proxy URL** in the “**URL**” field, e.g: (Section 8.1.3)



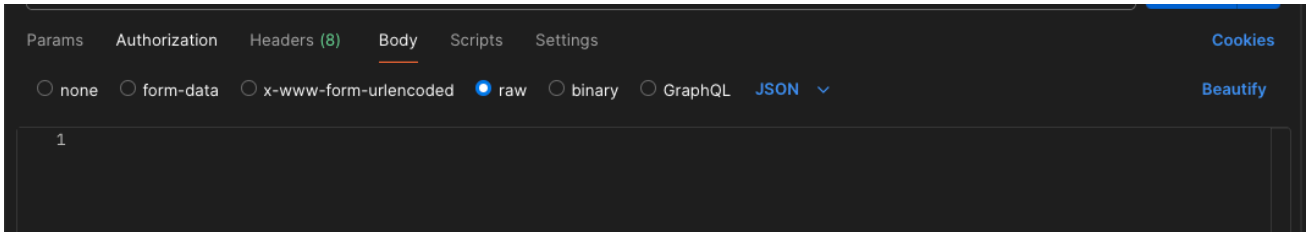
Click the “**Authorization**” tab.

Ensure the “**Auth Type**” field is set to “**Inherit auth from parent**”, for example:



Click the “**Body**” tab.

Select the “**raw**” radio button:



9.5. Test

Copy and paste the XML as described in the *QAPI Developer Guide* into the "**Body**" field and click "**Send**" to send the request.

As a quick test, paste the XML below into the "**Body**" field to attempt to get the context for a QAPI calculation using /QTWY/QAPI_CALC_CONTEXT_GET:

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:_-QTYW_-QAPI_CALC_CONTEXT_GET xmlns:ns0="urn:sap-com:document:sap:rfc:functions">
    <IS_CONTEXT_PARAMETERS>
        <CONVERSION_GROUP>TEST</CONVERSION_GROUP>
    </IS_CONTEXT_PARAMETERS>
</ns0:_-QTYW_-QAPI_CALC_CONTEXT_GET>
```

This should return a set of XML from QAPI, likely stating that the conversion group of "TEST" does not exist. If you see this, your configuration and connection was successful!

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rfc:_-QTYW_-QAPI_CALC_CONTEXT_GET.Response xmlns:rfc="urn:sap-com:document:sap:rfc:functions">
    <ET_INPUT_PARAMETERS/>
    <ET_INP_PARAM_ALLOWED_UOM/>
    <ET_RESULT_PARAMETERS/>
    <ET_RES_PARAM_ALLOWED_UOM/>
    <ET_RETURN>
        <item>
            <TYPE>E</TYPE>
            <ID>/QTYW/QCI</ID>
            <NUMBER>024</NUMBER>
            <MESSAGE>Conversion group TEST is not defined in customizing</MESSAGE>
            <LOG_NO/>
            <LOG_MSG_NO>000000</LOG_MSG_NO>
            <MESSAGE_V1>TEST</MESSAGE_V1>
            <MESSAGE_V2/>
            <MESSAGE_V3/>
            <MESSAGE_V4/>
            <PARAMETER/>
            <ROW>0</ROW>
            <FIELD/>
            <SYSTEM>SOICLNT030</SYSTEM>
        </item>
    </ET_RETURN>
</rfc:_-QTYW_-QAPI_CALC_CONTEXT_GET.Response>
```

9.6. PHP CURL Example

9.6.1. Get Token

Below is an example of a “get token” request, showing how CURL should be configured. This is shown in PHP.

Note that the curl option “CUSTOMREQUEST” must be set to “GET”. This is required by BTP so it knows it’s a "GET" request (in terms of C.R.U.D), otherwise a 403 will occur.

```
// Connection configuration
$url =              "https://YOUR_TOKEN_URL";    // Obtained in 8.1 Key Get Info for Access
$clientID =         "YOUR_CLIENT_ID ";           // Obtained in 8.1 Key Get Info for Access
$clientSecret =     "YOUR_CLIENT_SECRET";        // Obtained in 8.1 Key Get Info for Access

// Prepare for curl
$headers = [
        "Cache-Control: no-cache",
        "Content-Type: application/x-www-form-urlencoded",
];
$authString =       $clientID . ":" . $clientSecret;
$postContent =      "grant_type=client_credentials";

// Configure curl
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_USERPWD, $authString);
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postContent);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "GET" );
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);

// Execute curl after this...
```

9.6.2. QAPI_CALC_CONTEXT_GET

Below is an example of a “QAPI_CALC_CONTEXT_GET” request, showing how CURL should be configured. This is shown in PHP.

Note that the curl option “CUSTOMREQUEST” must be set to “GET”. This is required by BTP so it knows it’s a "GET" request (in terms of C.R.U.D), otherwise a 403 will occur.

```
// Connection configuration
$url =              "https:// API_END_POINT_URL";  // Obtained in 8.1 Key Get Info for Access
$accessToken =      "ACCESS_TOKEN";                // Obtained in 9.5.1 Get Token

// Prepare for curl
$headers = [
        "Cache-Control: no-cache",
        "Content-Type: application/xml",
        "Authorization: Bearer {$accessToken}"
];
$authString =       $clientID . ":" . $clientSecret;

$postContent = <<<EOXML
<?xml version="1.0" encoding="UTF-8"?>
<ns0:_-QTYW_-QAPI_CALC_CONTEXT_GET xmlns:ns0="urn:sap-com:document:sap:rfc:functions">
<IS_CONTEXT_PARAMETERS>
    <CONVERSION_GROUP>TEST</CONVERSION_GROUP>
</IS_CONTEXT_PARAMETERS>
</ns0:_-QTYW_-QAPI_CALC_CONTEXT_GET>
EOXML;

// Configure curl
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postContent);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "GET" );
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);

// Execute curl after this...
```

Legal Notices