Hands-On CPI instructions

Run Simple

TABLE OF CONTENTS

PREREQUISITES

i. Cloud Platform Integration tenant on SAP Cloud Platform

You require a Cloud Platform Integration tenant on SAP Cloud Platform, if you already have an SAP Cloud Platform Integration tenant you can skip this step.

Follow the instructions at Set Up Your Trial SAP Cloud Platform Integration Tenant to create your CPI trial tenant.

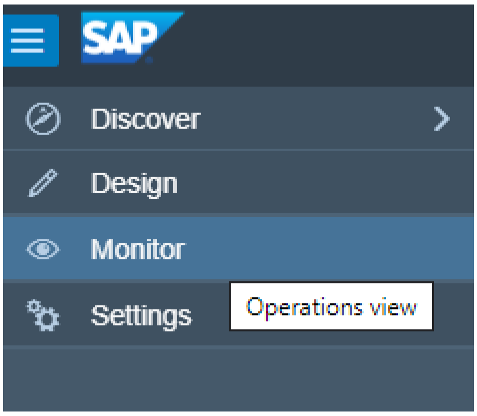
ii. Import oData services into your SAP Business ByDesign tenant (not required for B1)

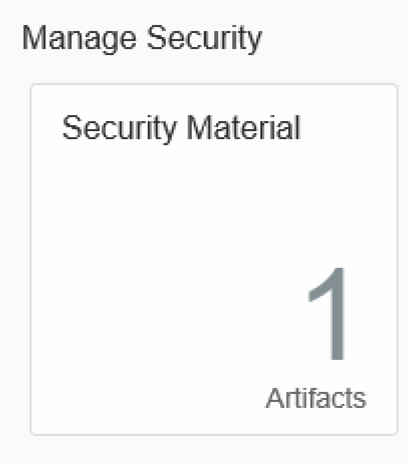
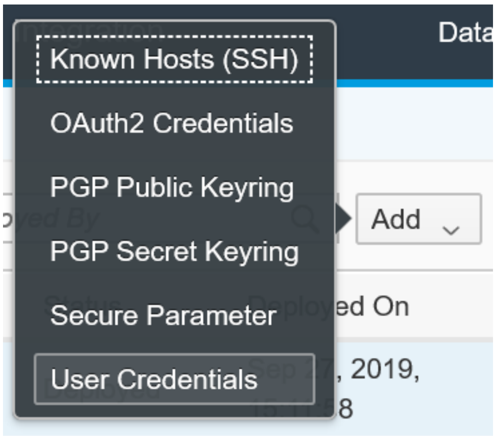
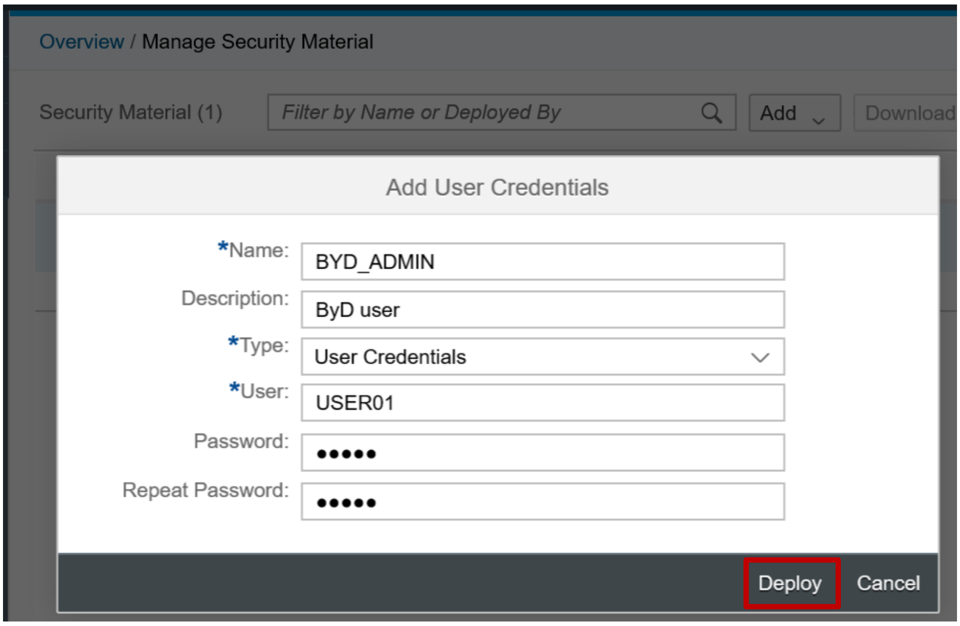
Read the explanations in the following blog to learn more about OData API Samples:
https://blogs.sap.com/2019/02/27/sap-business-bydesign-api-usage-samples/

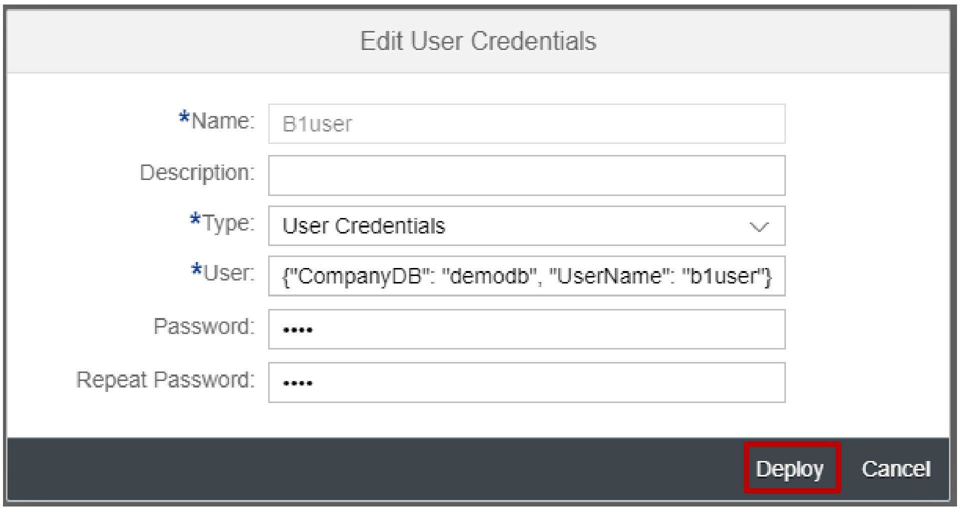
Go to the following repository to understand how to download and import the provided OData API samples provided:
https://github.com/SAP/sapbydesign-api-samples
Go to the **Configuration** section in the Github repository README.txt file to understand the steps to follow in order to import the OData API samples provided into your SAP Business ByDesign tenant.

We are using the khsalesorder OData service in these exercises, only that service requires then to be imported.

iii. Store B1 and/or ByD User Credentials at SAP Cloud Platform Integration tenant

Explanation	Screenshot
Chose the Operations/Monitor view.	 SAP Discover Design Monitor Settings Operations view

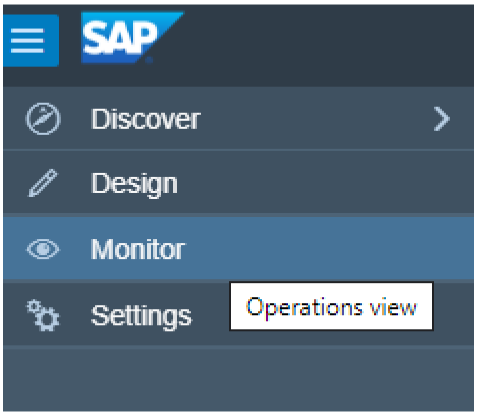
Explanation	Screenshot
Go to the **Manage Security** section, **Security Material** tile.	
Select **Add** and choose **User Credentials**.	 Known Hosts (SSH) OAuth2 Credentials PGP Public Keyring PGP Secret Keyring Secure Parameter User Credentials Add Deployed On Sep 27, 2019,
<span>**For ByD:**</span> Enter a **name** for the User Credentials, a **ByD user** and **password**. Make sure the Type is **User Credentials**. Press **Deploy** to save the credentials.	 Overview / Manage Security Material Security Material (1) Filter by Name or Deployed By Add Download Add User Credentials Deploy Cancel *Name: BYD_ADMIN Description: ByD user *Type: User Credentials *User: USER01 Password: ●●●●● Repeat Password: ●●●●●

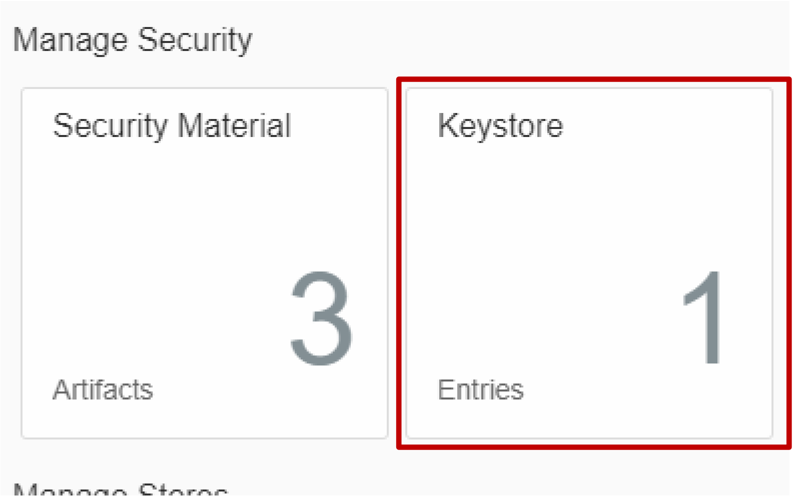
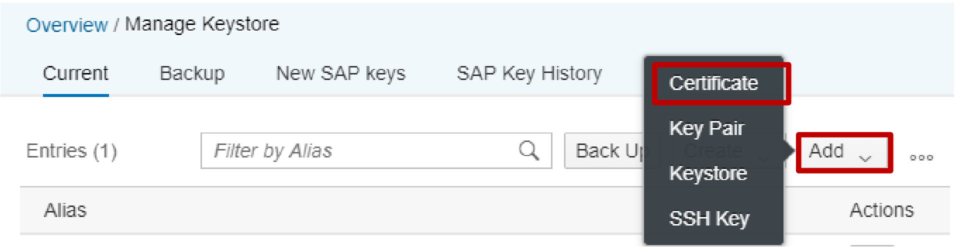
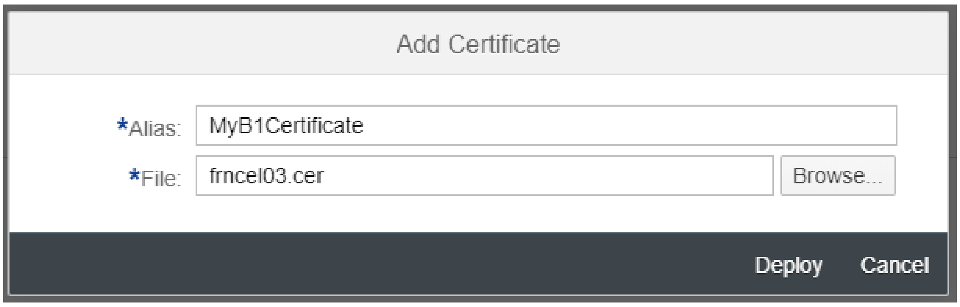
Explanation	Screenshot
For B1: Enter a **name** for the User Credentials, a **B1 user** (json format containing the CompanyDB and the UserName) and **password**. Make sure the Type is **User Credentials**. Press **Deploy** to save the credentials.	 Edit User Credentials *Name: B1user Description: *Type: User Credentials *User: {"CompanyDB": "demodb", "UserName": "b1user"} Password: •••• Repeat Password: ••••

iv. For B1: create a CPI Keystore (not required for ByD)

In order to be able to connect to SAP Business One Service Layer from SAP Cloud Platform Integration via OData v4, B1 certificate needs to be imported into CPI as a Keystore.
You can find more details about this topic in the blog Cloud Integration – How to Setup Secure Outbound HTTP Connection using Keystore Monitor.

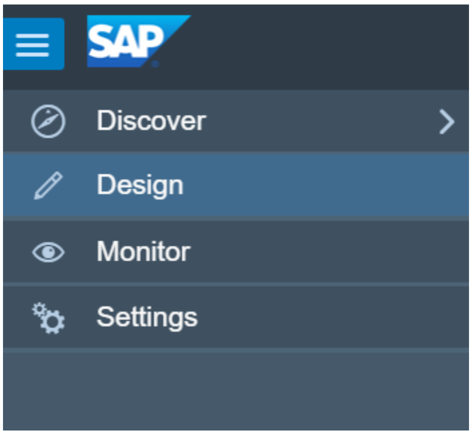
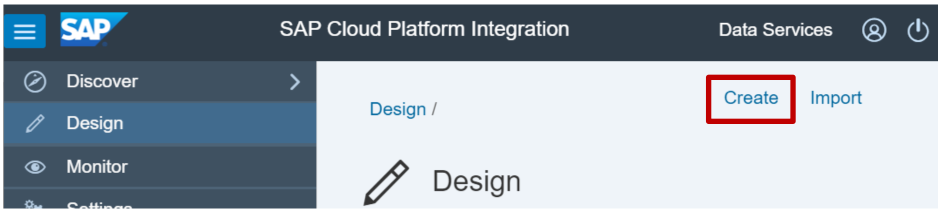
Before following the next steps please make sure you have your B1 certificate saved as a file in your disk.

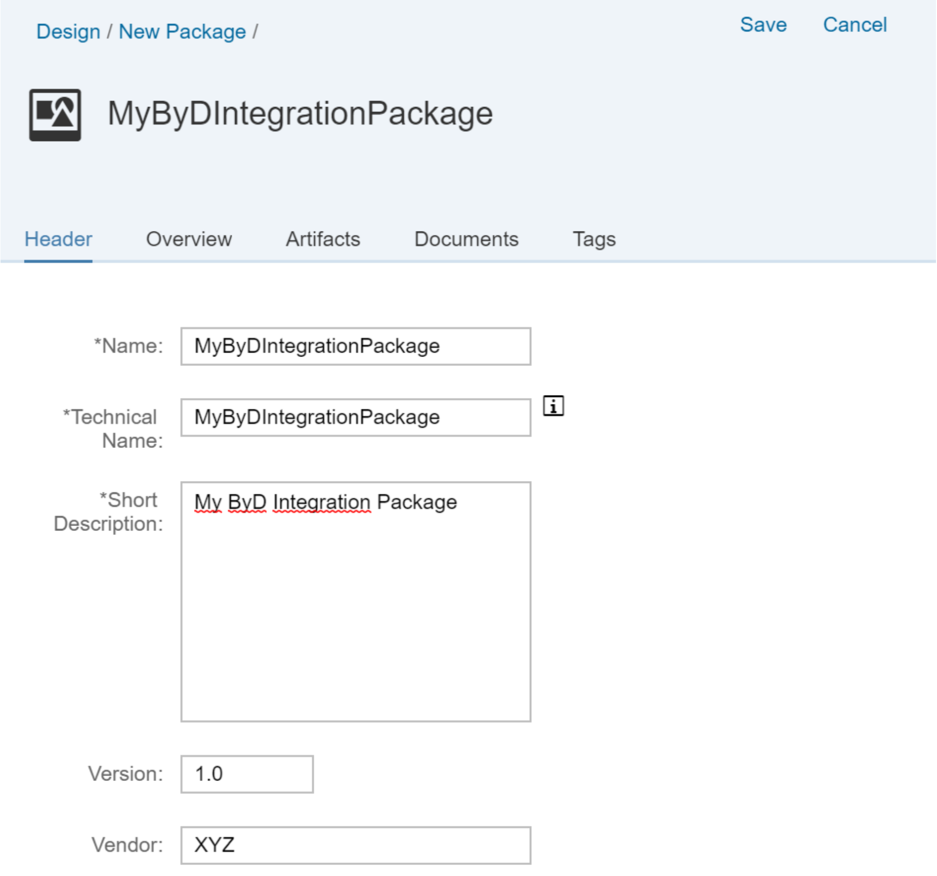
Explanation	Screenshot
Chose the Operations/Monitor view.	 SAP Discover Design Monitor Settings Operations view

Explanation	Screenshot
Go to the **Manage Security** section, **Keystore** tile.	 Manage Security Security Material 3 Artifacts Keystore 1 Entries Manage Stores
Choose **Add** -> **Certificate** to create a new Keystore containing B1 certificate.	 Overview / Manage Keystore Current Backup New SAP keys SAP Key History Certificate Key Pair Keystore SSH Key Entries (1) Filter by Alias Back Up Add Alias Actions
Enter an **Alias** for your **certificate** and select your B1 certificate file. Press **Deploy** to add the certificate to the keystore.	 Add Certificate *Alias: MyB1Certificate *File: frncel03.cer

STEP 1: CREATE AN INTEGRATION PACKAGE

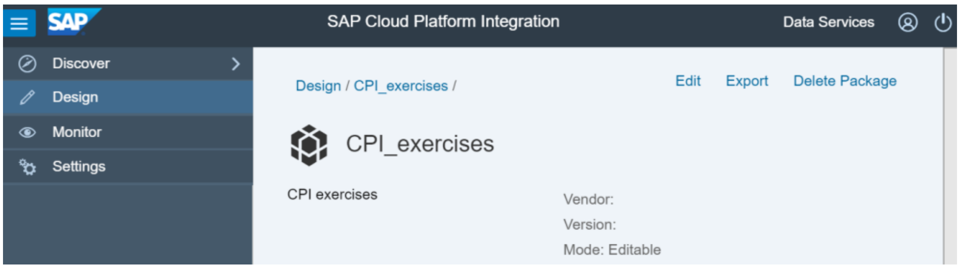
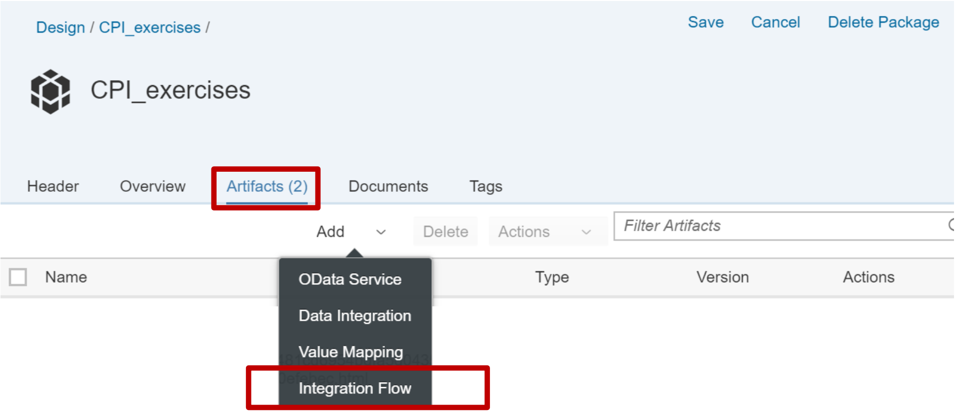
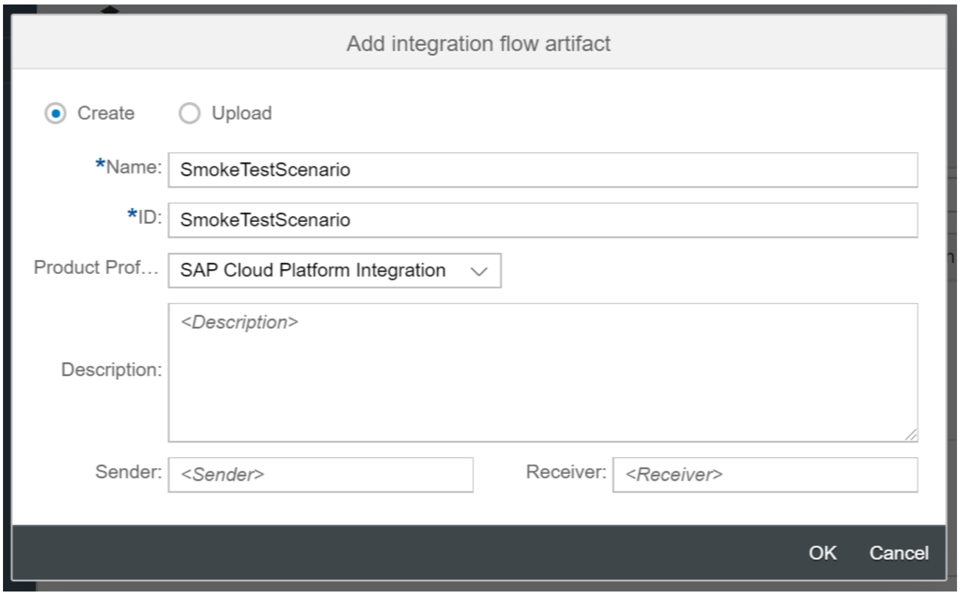
Create an integration package that contains your integration flows.

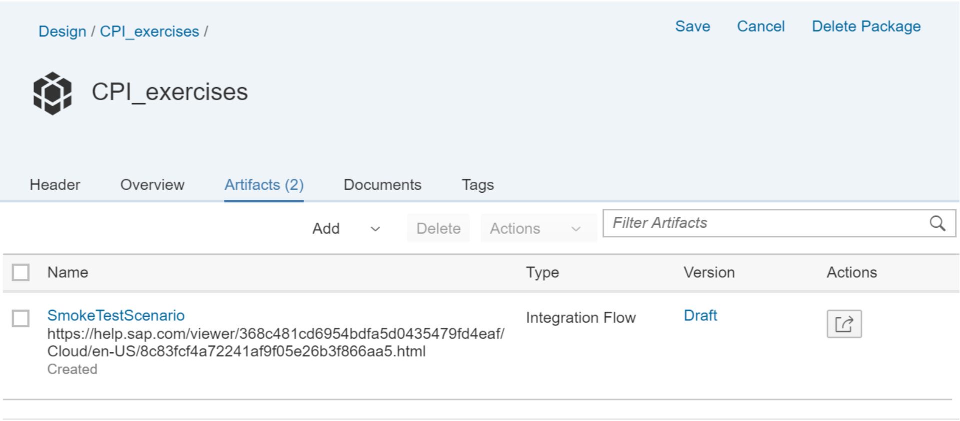
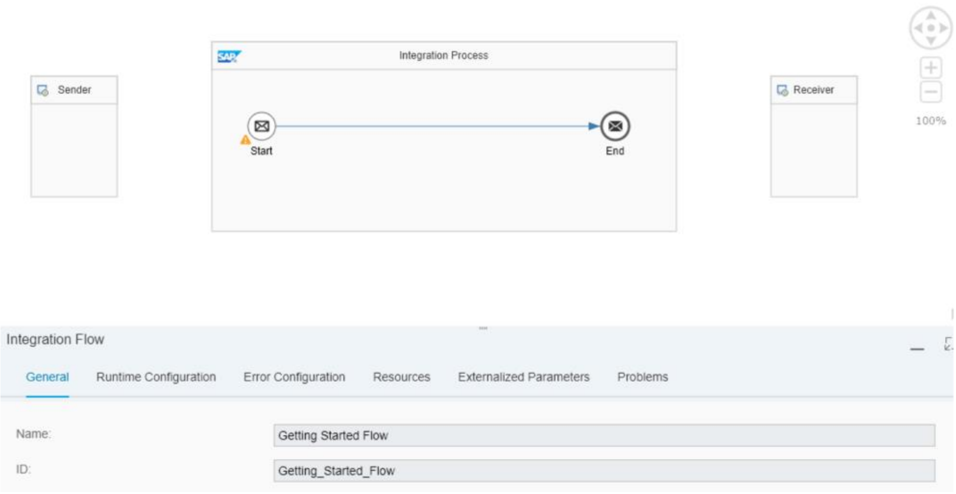
Explanation	Screenshot
Open the Web UI using the hyperlink provided to you.	
Go to the **Design** section of the Web UI.	
Choose **Create**.	 SAP SAP Cloud Platform Integration Data Services Discover Design Monitor Settings Design / Create Import Design

Explanation	Screenshot
Enter a name and description for your integration package and choose **Save**.	 Design / New Package / Save Cancel Header Overview Artifacts Documents Tags MyByDIntegrationPackage *Name: MyByDIntegrationPackage *Technical Name: MyByDIntegrationPackage *Short Description: My ByD Integration Package Version: 1.0 Vendor: XYZ

STEP 2: CREATE AN INTEGRATION FLOW

Create the integration flow as part of your integration package.

Explanation	Screenshot
Select your integration package (in my case named CPI_exercises) and choose **Edit**.	 SAP Cloud Platform Integration Design / CPI_exercises / CPI_exercises CPI exercises Vendor: Version: Mode: Editable
Go to the **Artifacts** tab and choose **Add -> Integration Flow.**	 Design / CPI_exercises / CPI_exercises
Provide a name and description for the integration flow and choose **OK**.	 Add integration flow artifact ● Create ○ Upload *Name: SmokeTestScenario *ID: SmokeTestScenario Product Prof…: SAP Cloud Platform Integration Description: <Description> Sender: <Sender> Receiver: <Receiver>

Explanation	Screenshot
Select the integration flow from the list of Artifacts of your package.	 Design / CPI_exercises / Save Cancel Delete Package CPI_exercises Header Overview Artifacts (2) Documents Tags Add Delete Actions Filter Artifacts □ Name | Type | Version | Actions □ SmokeTestScenario https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US/8c83fcf4a72241af9f05e26b3f866aa5.html Created | Integration Flow | Draft
An integration flow template opens that contains the following shapes: - Sender (represents your sender system) - Receiver (represents a receiver system) - Integration Process (contains all the processing steps that define how a message is processed on the tenant). The Integration Process shape contains a Start and an End event.	 Integration Flow General Runtime Configuration Error Configuration Resources Externalized Parameters Problems Name: Getting Started Flow ID: Getting_Started_Flow If you select a shape in the integration flow modeling area, the properties of the selected shape are displayed in the section below the modeling area. If you click the area outside of the shapes, the properties are displayed that are related to the integration flow.

STEP 3: SMOKE TEST SCENARIO

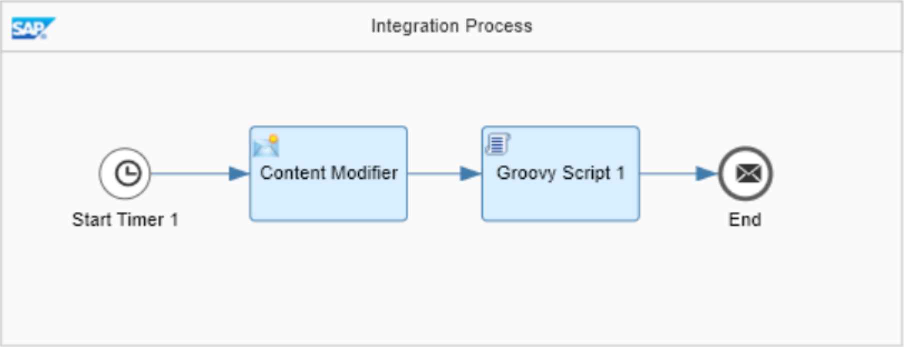
This is a very simple test to verify that your SAP Cloud Platform Integration is working as expected. You do not need any receiver system to perform this test.

In this scenario, you create a *Hello World* text and write it into the message body (scheduled on deployment of the integration flow). The result is written into the message processing log which you can directly inspect with the message monitoring application.

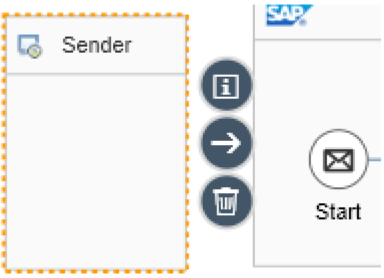
Note: This is not according to standard best practice. When designing productive scenarios, don't store the message payload in the message processing log. This can cause severe issues with memory consumption. The reason is that tasks such as message processing and message monitoring share the same memory and CPU which are available on your tenant.

For more detail son this scenario please check this page:
https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US/8c83fcf4a72241af9f05e26b3f866aa5.html

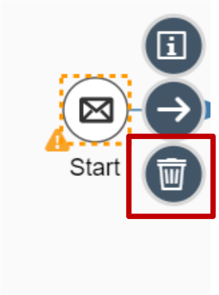
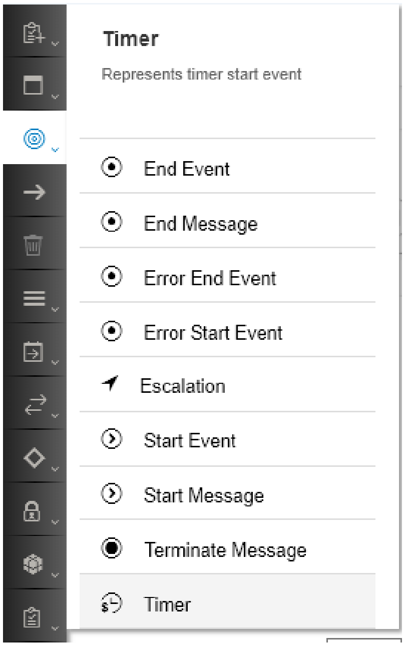
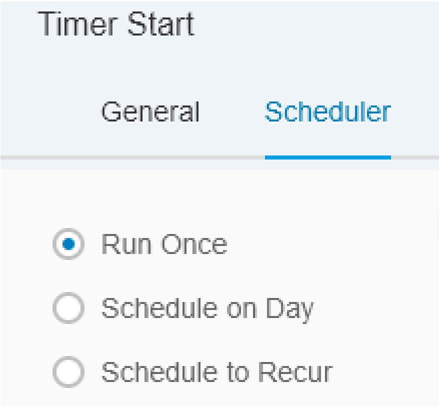
In this step we will develop the following scenario:



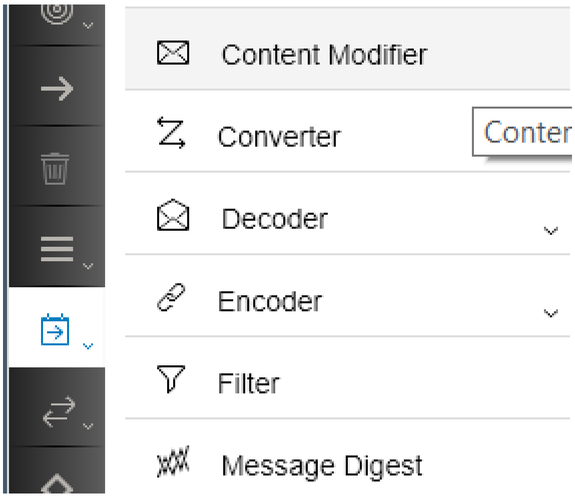
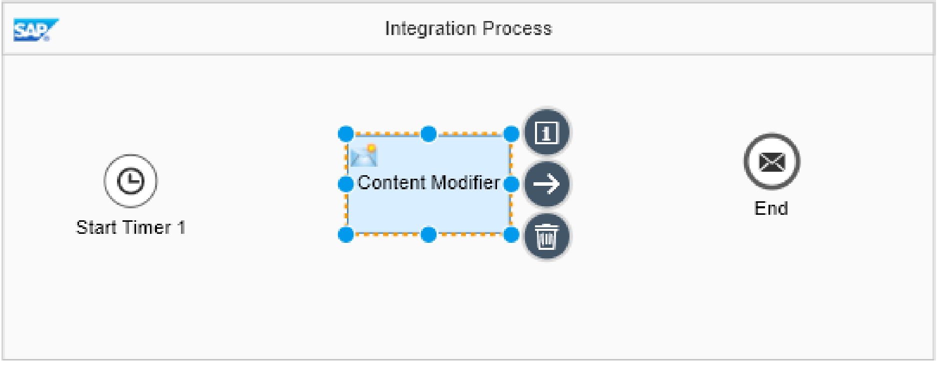
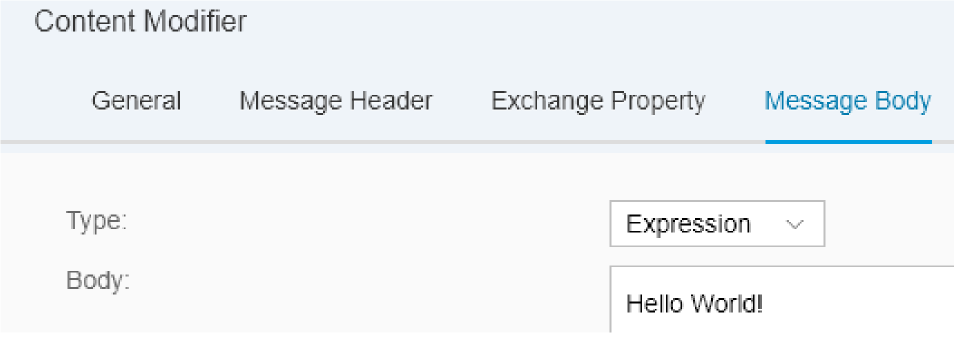
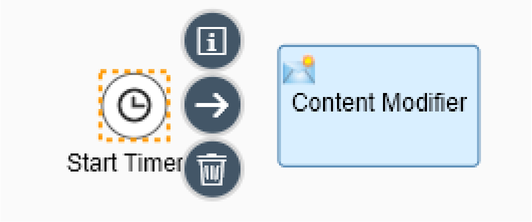
i. Remove the Sender and Receiver shapes

Explanation	Screenshot
Open the integration flow model (Edit mode), select the Sender shape, and choose the recycle bin icon (to **remove the Sender shape**). In the same way, **remove the Receiver shape**.	

ii. Add a Start Timer

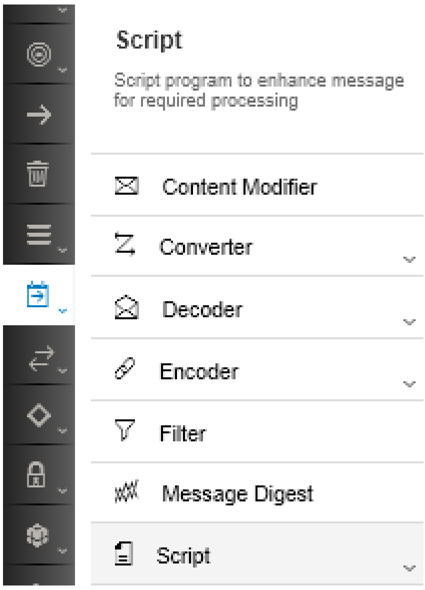
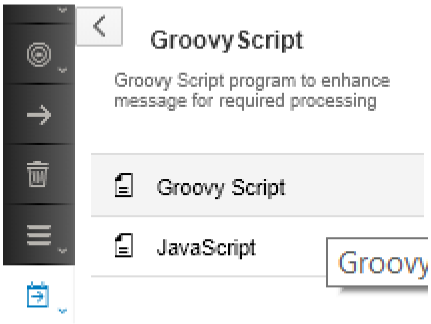
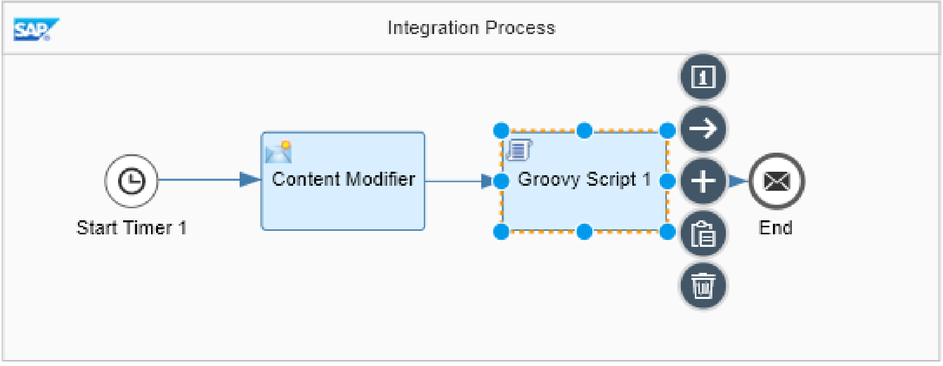
Explanation	Screenshot
Remove the Start message shape.	
In the palette, select the **Events** entry, and then select the **Timer** shape. Place the Timer shape in the Integration Process shape (at the point where the Start event was previously located).	 Timer Represents timer start event End Event End Message Error End Event Error Start Event Escalation Start Event Start Message Terminate Message Timer
In the properties section of the Timer event (displayed below the integration flow model when the Timer event is selected in the model), go to the **Scheduler** tab. Make sure that the option **Run Once** is selected. It will be run only once every time we deploy the flow.	 Timer Start ● Run Once ○ Schedule on Day ○ Schedule to Recur

iii. Create a Content Modifier to Define the Message Body

Explanation	Screenshot
To add a Content Modifier, go to the palette, choose the **Message Transformers** icon, and select the **Content Modifier** icon.	 Content Modifier Converter Conter Decoder Encoder Filter Message Digest
Place the Content Modifier in the model after the Timer Start event.	 SAP Integration Process Start Timer 1 Content Modifier End
In the Content Modifier properties section, go to the **Message Body** tab and enter the following string sequence in the entry field: `Hello World!` This simulates the inbound XML message.	 Content Modifier General · Message Header · Exchange Property · Message Body Type: Expression Body: Hello World!
Connect the Timer event with the Content Modifier. To do this, select the Timer event. Click the arrow icon and drag and drop	

Explanation	Screenshot
the cursor to the Content Modifier.	

iv. Create a Script Step to Log the Payload

Explanation	Screenshot
To add a Script step (containing a Groovy script), go to the palette and choose the **Message Transformers** icon and select the **Script** icon.	 Script Script program to enhance message for required processing Content Modifier Converter Decoder Encoder Filter Message Digest Script
In the **Script** submenu, select **Groovy Script**. Place the Script step shape after the Content Modifier step and connect both shapes.	 Groovy Script Groovy Script program to enhance message for required processing
Select the Script step. The context icons are displayed. Choose the **+** icon.	 Integration Process Start Timer 1 Content Modifier Groovy Script 1 End

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>

The default script coding of the step is displayed.

Replace the default content by the script provided in the coding example below.

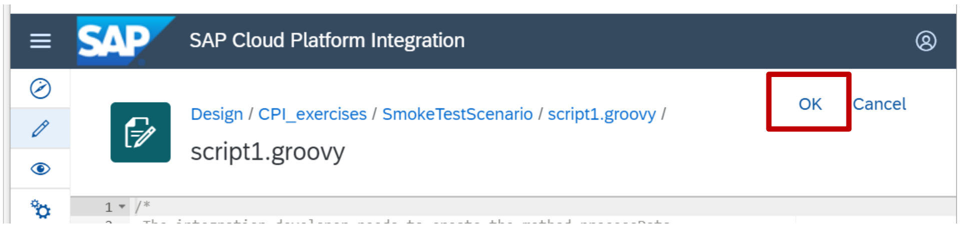
</td><td>

```
import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;

def Message processData(Message message)
{
  def body = message.getBody(java.lang.String) as String;
  def messageLog = messageLogFactory.getMessageLog(message);
  if(messageLog != null)
  {
    messageLog.addAttachmentAsString("Log current Payload:", body, "text/plain");
  }
  return message;
}
```

</td></tr>
<tr><td>

Choose **OK** to close the script editor and go back to the model.

</td><td>


SAP Cloud Platform Integration
Design / CPI_exercises / SmokeTestScenario / script1.groovy /
script1.groovy
OK
Cancel
1 /*

</td></tr>
</table>

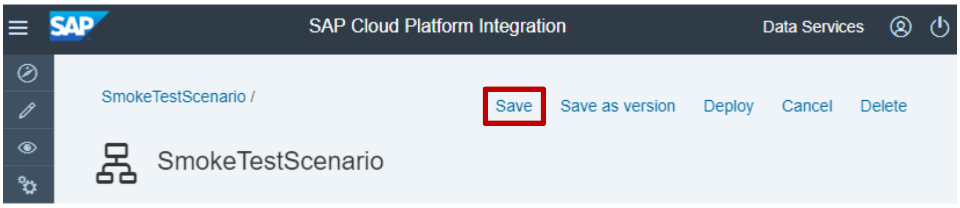
v. Deploy the integration flow

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>

When you have finished modeling, click **Save**.

On successful save, a corresponding status message is displayed.

</td><td>

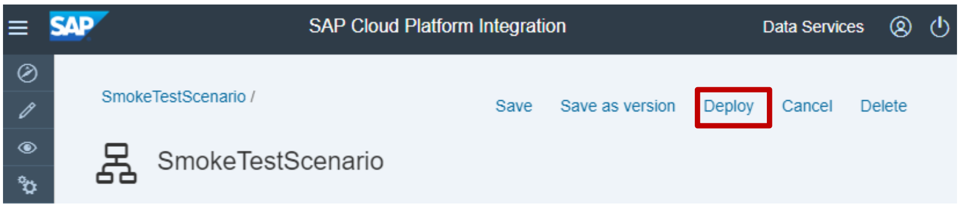


</td></tr>
<tr><td>

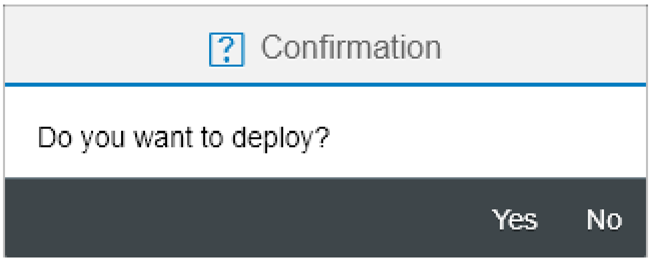
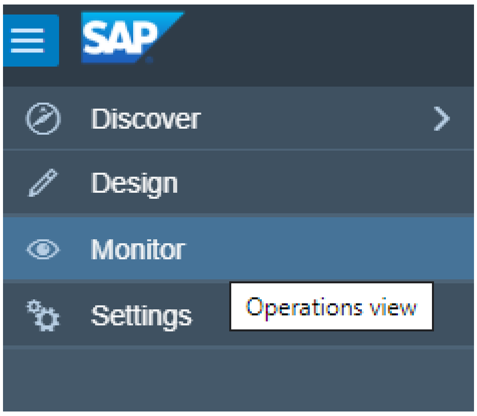
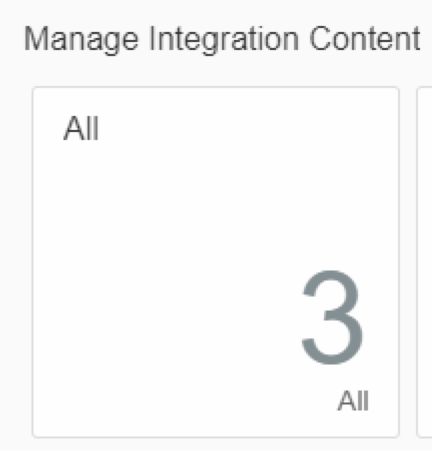
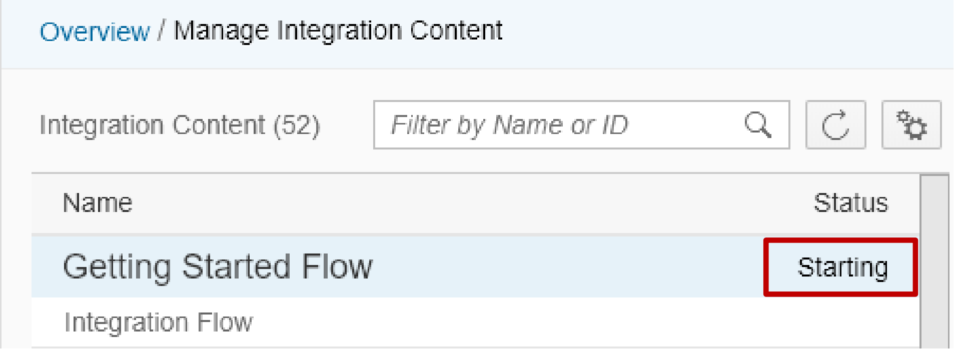
Click **Deploy**.

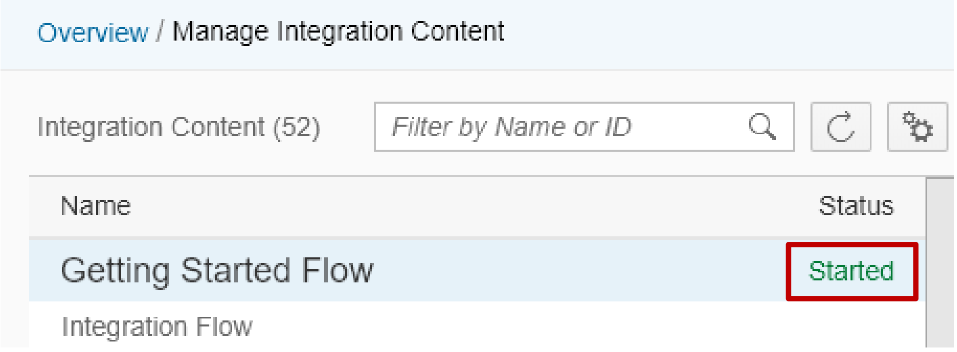
A message is displayed that asks you to confirm this action.

</td><td>

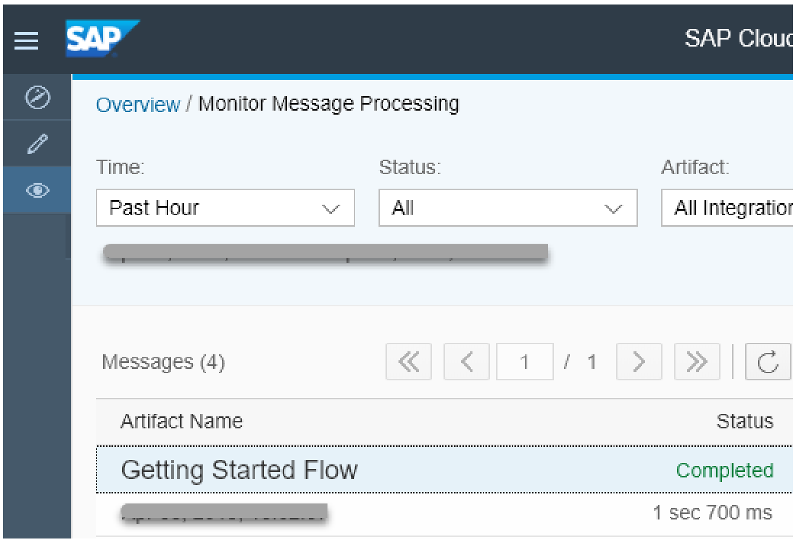


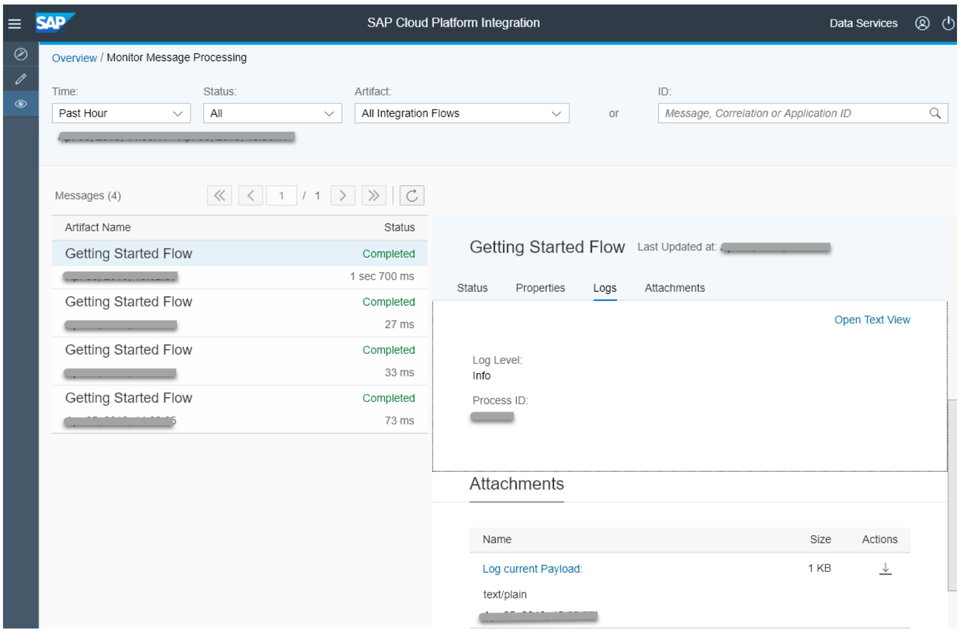
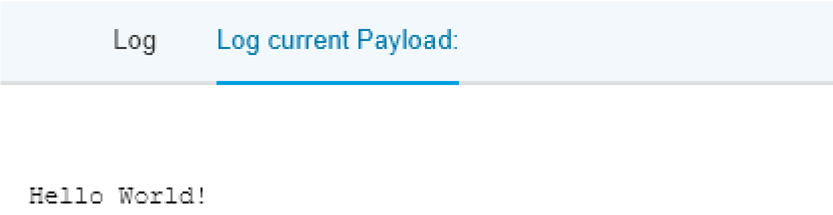
</td></tr>
</table>

Explanation	Screenshot
	 Confirmation Do you want to deploy?
Chose the **Operations/Monitor view** to check the status of the deployment.	 SAP Discover Design Monitor Settings Operations view
Click the first tile in the **Manage Integration Content** section.	 Manage Integration Content
You can check the deployment status of your integration flow. At the beginning the status of your flow will be **Starting** and afterwards move to **Started**.	 Overview / Manage Integration Content Integration Content (52) Name Status Getting Started Flow Starting Integration Flow

Explanation	Screenshot
	 Overview / Manage Integration Content Integration Content (52) Filter by Name or ID Name Status Getting Started Flow Started Integration Flow

vi. Monitor Message Processing

Explanation	Screenshot
Go to the Operations view and select a tile under **Monitor Message Processing**. If your integration flow has been processed successfully, the status **Completed** should be shown.	 SAP Clouc Overview / Monitor Message Processing Time: Past Hour Status: All Artifact: All Integratior [redacted] Messages (4) 1 / 1 Artifact Name Status Getting Started Flow Completed [redacted] 1 sec 700 ms

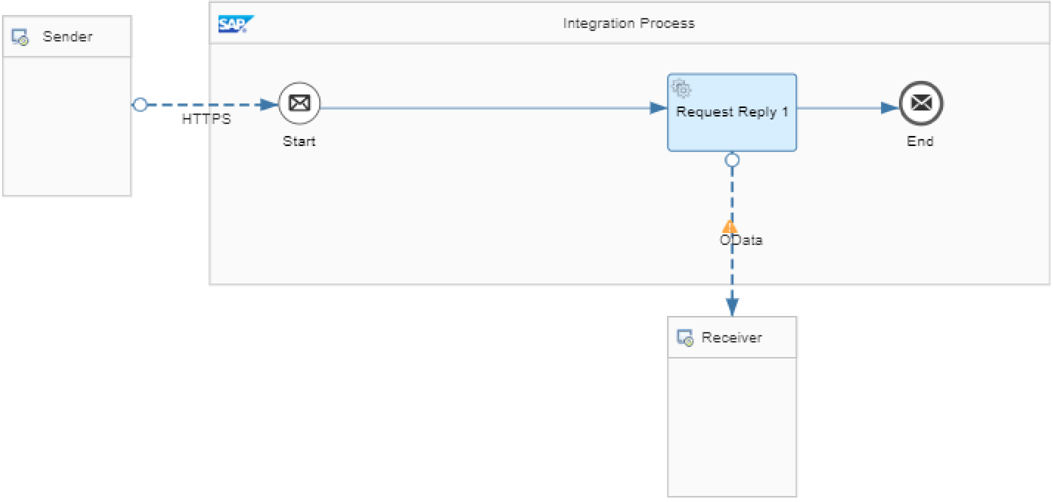
Explanation	Screenshot
Select the integration flow and analyze the details area to the right of the integration flow list.	 SAP Cloud Platform Integration Data Services Overview / Monitor Message Processing Time: Past Hour Status: All Artifact: All Integration Flows or ID: Message, Correlation or Application ID [redacted] Messages (4) 1 / 1 Artifact Name — Status Getting Started Flow — Completed [redacted] — 1 sec 700 ms Getting Started Flow — Completed [redacted] — 27 ms Getting Started Flow — Completed [redacted] — 33 ms Getting Started Flow — Completed [redacted] — 73 ms Getting Started Flow Last Updated at: [redacted] Status Properties Logs Attachments Open Text View Log Level: Info Process ID: [redacted] Attachments Name — Size — Actions Log current Payload: — 1 KB text/plain [redacted]
Under **Attachments**, click **Log current Payload**. You should see the message content, which consists of the following text: Hello World!	 Log Log current Payload: `Hello World!` This shows you that the message has been processed correctly.

STEP 4: HTTPS SENDER INITIATED SCENARIO, GET REQUEST

Creates a simple integration scenario that is initiated by a sender (using the HTTPS sender adapter) and retrieving SalesOrders details from SAP Business ByDesign or SAP Business One.

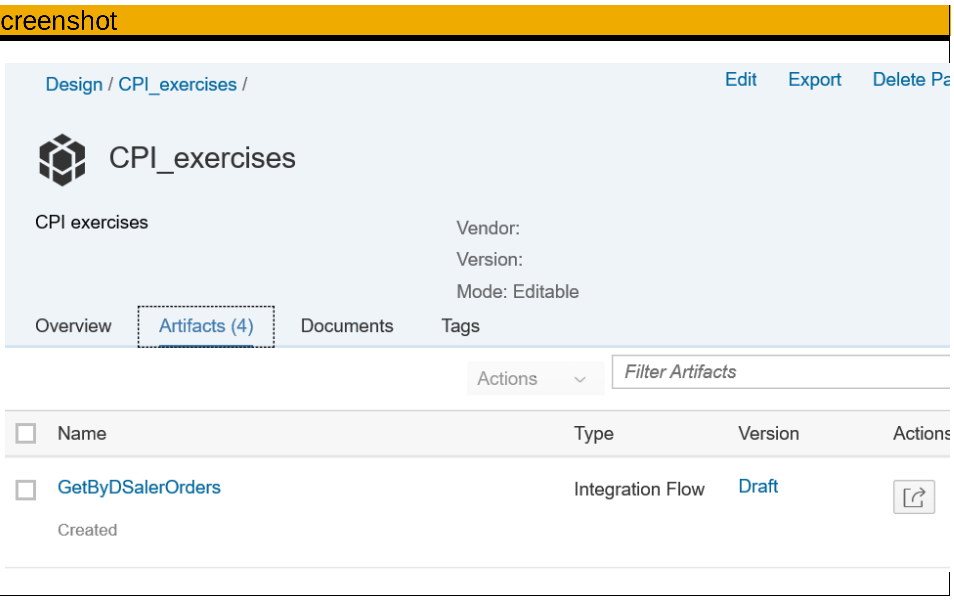
When you have finished the integration flow design, you can call the flow through an HTTP client.

The flow will look as follows at the end:



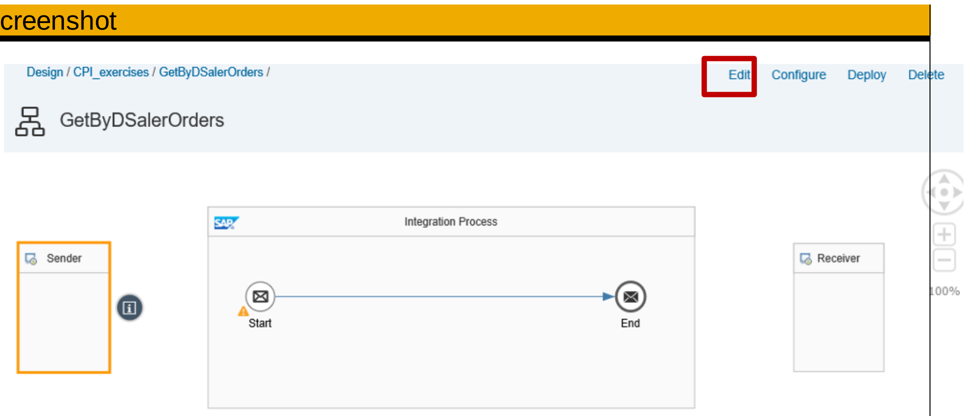
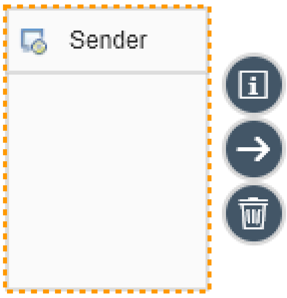
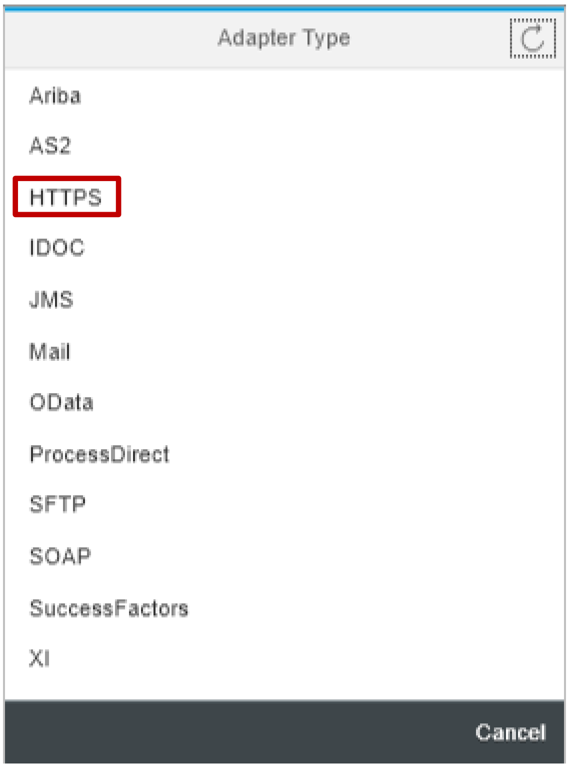
A similar exercise (not connecting to B1/ByD though) is detailed here: https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US/ccdb189ad99a4db0908150c799cbe9e3.html

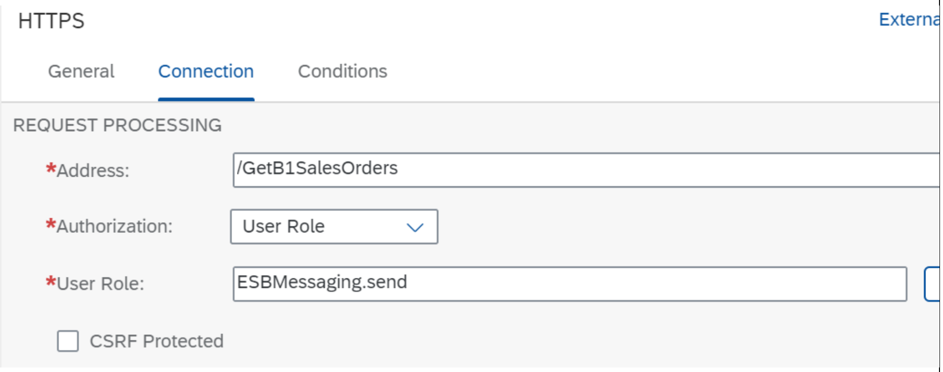
i. Create an Integration Flow

Explanation	Screenshot
Create a new Integration Flow by following STEP 2: CREATE AN INTEGRATION FLOW. Call the new Integration Flow something like: “GetByDSalesOrders” or “GetB1SalesOrders”.	 Design / CPI_exercises / Edit Export Delete Pa CPI_exercises CPI exercises Vendor: Version: Mode: Editable Overview Artifacts (4) Documents Tags Actions Filter Artifacts □ Name Type Version Actions □ GetByDSalerOrders Created Integration Flow Draft

ii. Create the HTTPS Sender Channel

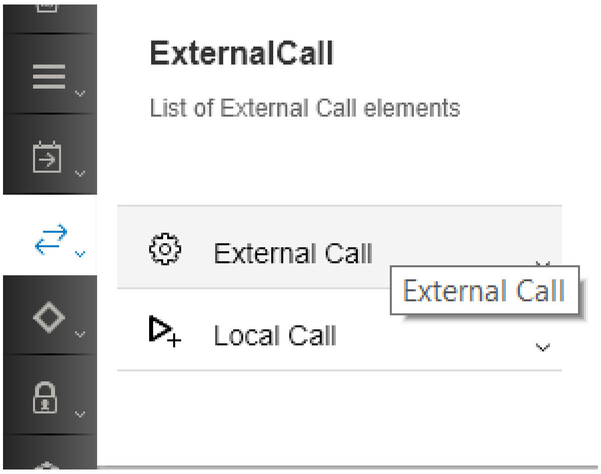
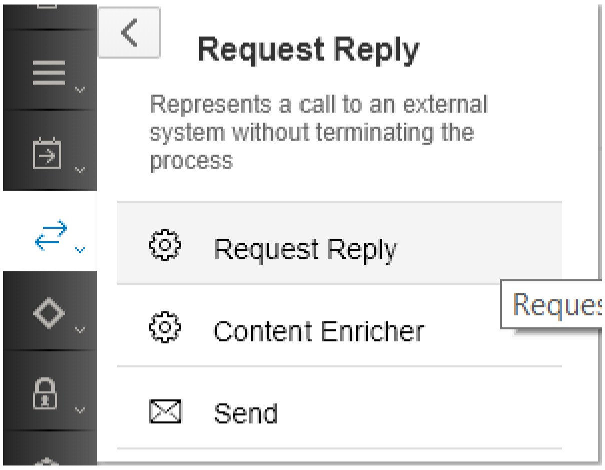
Add an HTTPS sender channel to enable the integration flow to receive HTTP requests.

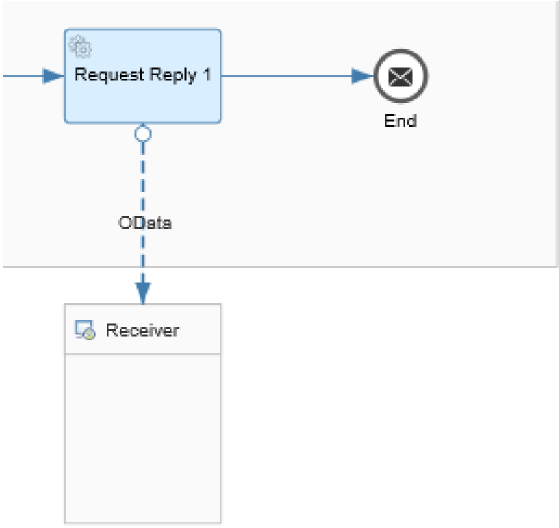
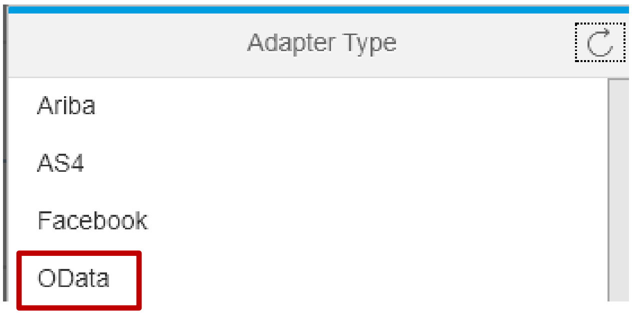
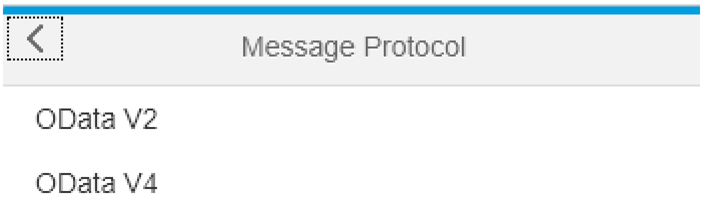
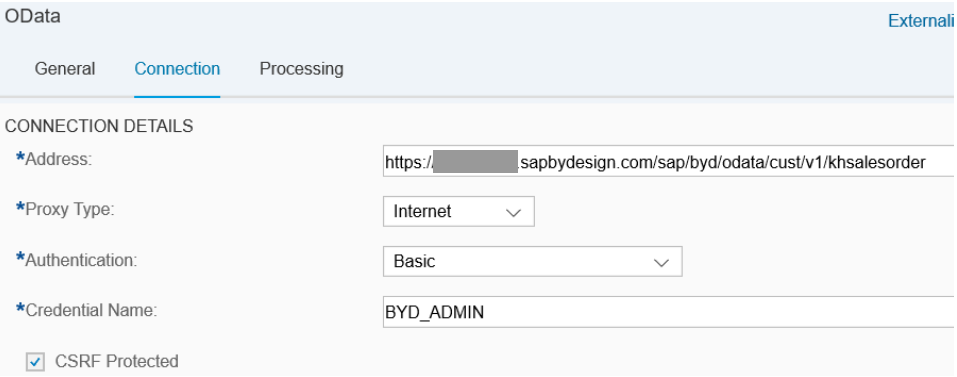
Explanation	Screenshot
Select the integration flow and choose **Edit**.	 Design / CPI_exercises / GetByDSalerOrders / Edit Configure Deploy Delete GetByDSalerOrders Sender Integration Process Start End Receiver 100%
Select the Sender shape. The context icons for the Sender appear.	
Click the arrow icon and drag and drop the cursor on the Start event. The list of available adapter types is displayed in a dialog. **Choose adapter type HTTPS**.	 Adapter Type Ariba AS2 HTTPS IDOC JMS Mail OData ProcessDirect SFTP SOAP SuccessFactors XI Cancel

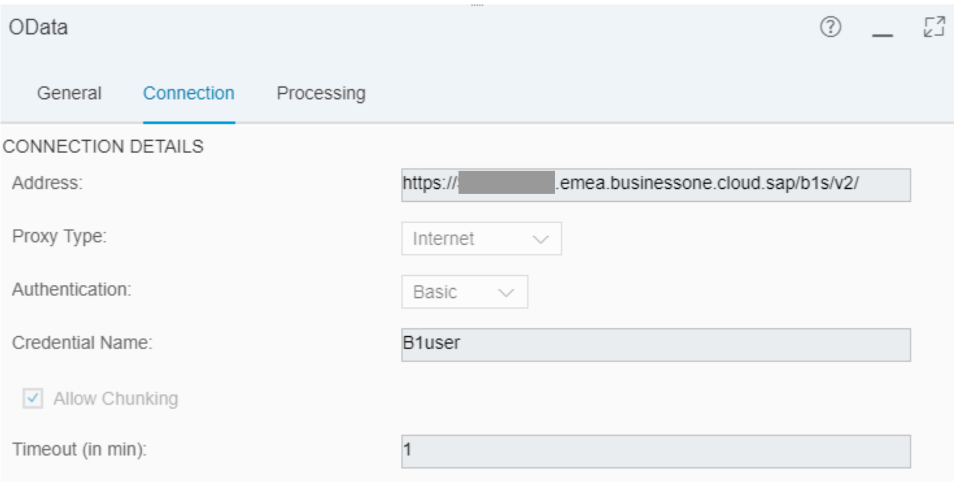
Explanation	Screenshot
Go to the **Connection** tab. Specify the required parameters: • Address **/GetByDSalesOrders** or **/GetB1SalesOrders** • Authorization • User Role • CSRF Protected	 HTTPS General | Connection | Conditions REQUEST PROCESSING *Address: /GetB1SalesOrders *Authorization: User Role *User Role: ESBMessaging.send □ CSRF Protected **Address**: endpoint under which the integration flow can be called from the sender. **Authorization** and **User Role**: Selecting **User Role** does not mean that you are determining the usage of basic authentication. **User Role** authorization only means that the permissions of the sender of the message are checked based on roles (which are assigned to the user that is associated with the sender). For productive scenarios, we recommend using client certificate authentication with certificate-to-user mapping. However, to simplify the setup of we propose that you choose basic authentication - simply because it is much easier to configure the sender in this case. **CSRF Protected**: Keep this option selected (default setting). It ensures that your integration flow is protected against Cross-Site-Request-Forgery, a kind of attack where a malicious party can perform harmful actions by masquerading as the logged in user (the user specified for the

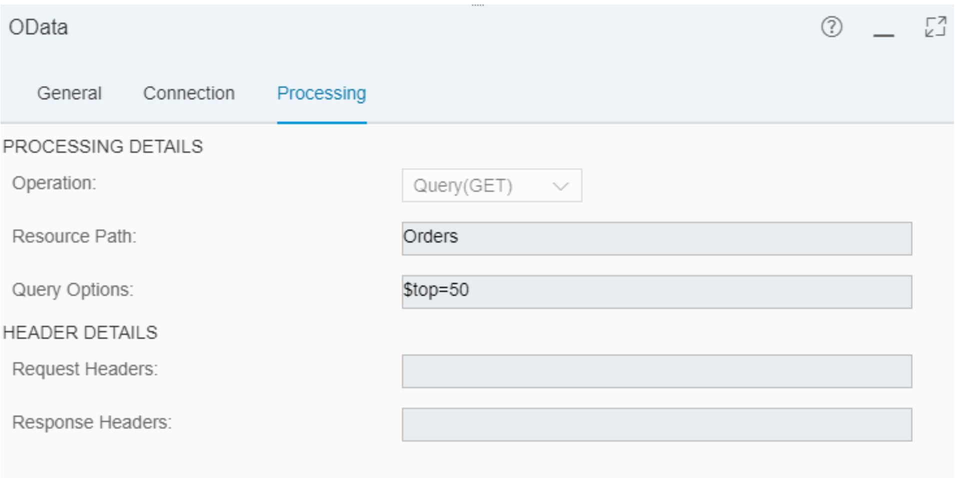
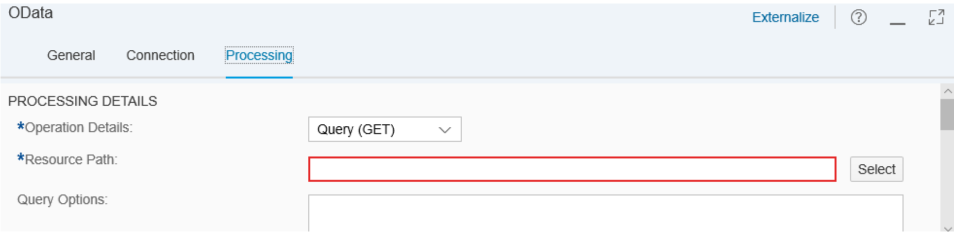
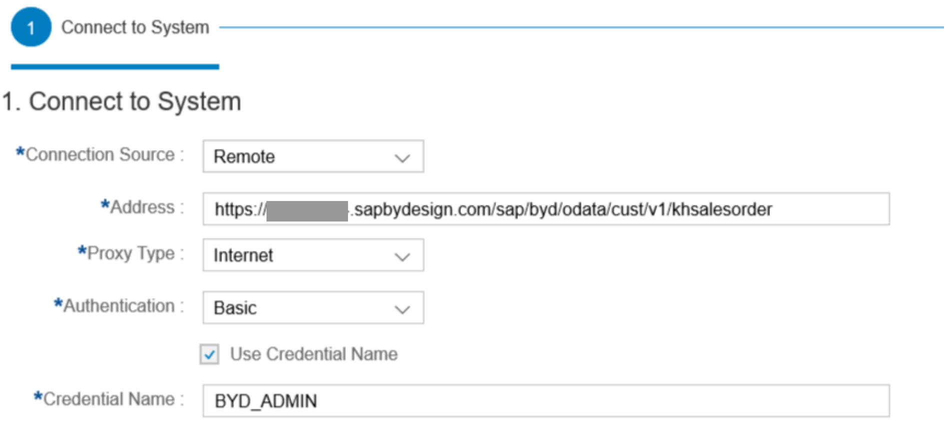
iii. Create the Outbound OData Channel

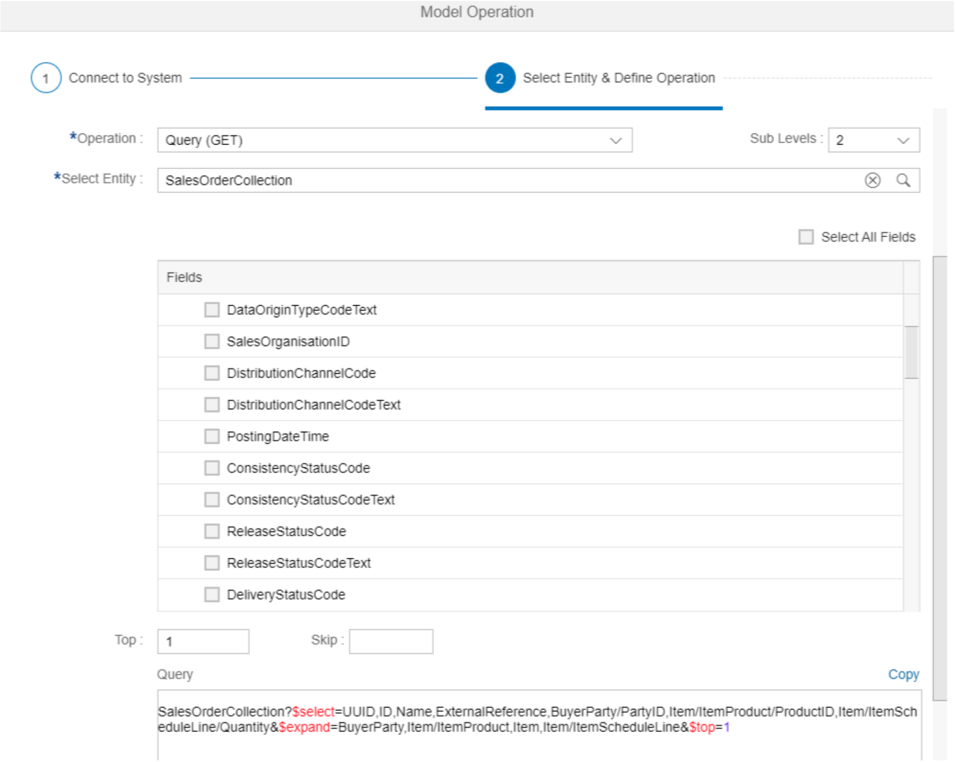
To call the B1/ByD external data source, add a Request Reply step to the integration flow model and connect this step with the external system using an OData channel.

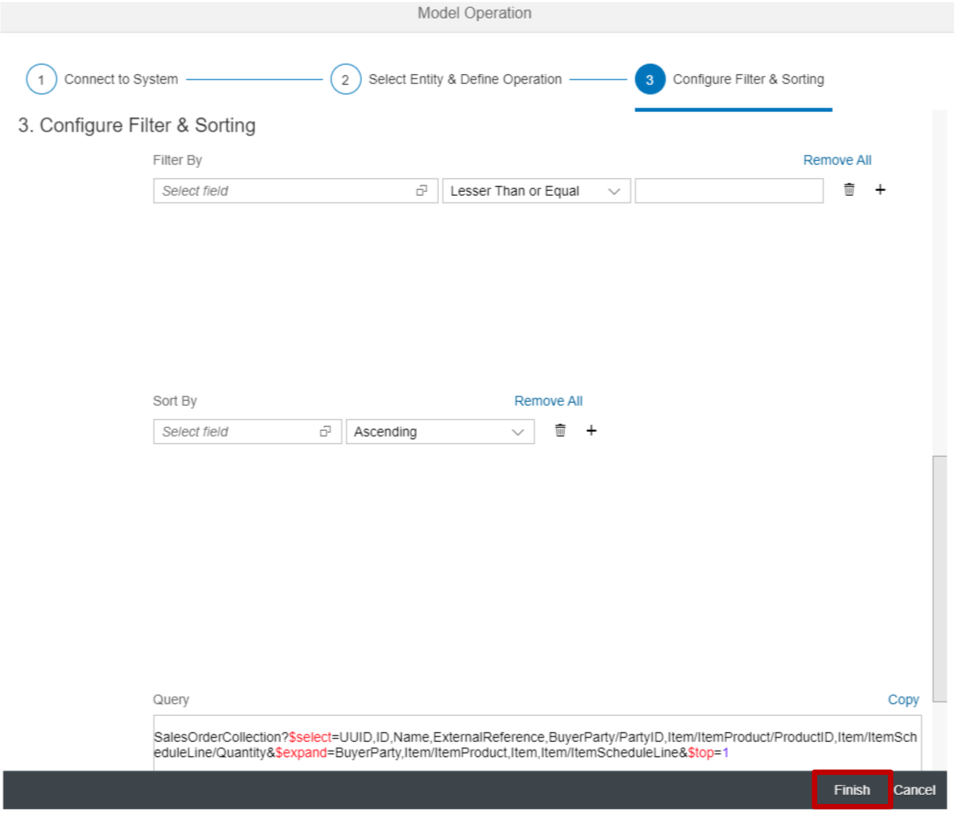
Explanation	Screenshot
Go to the palette and select the **Call -> External Call** option.	 ExternalCall List of External Call elements
In the submenu choose **Request Reply**.	 Request Reply Represents a call to an external system without terminating the process

Explanation	Screenshot
Place the Request Reply before the End event in the model. Move the Receiver shape closer to the Request Reply shape. **Connect the Request Reply shape to the Receiver**.	
In the next dialog, choose adapter type **OData**.	
In the next dialo, as **Message Protocol** **For ByD:** select **OData V2**. **For B1:** select **OData V4**.	
For ByD: Go to the **Connection** tab of the OData adapter and enter your ByD OData **Address**: https://myByD.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder Select :	OData General Connection Processing CONNECTION DETAILS *Address: https:/[redacted]sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder *Proxy Type: Internet *Authentication: Basic *Credential Name: BYD_ADMIN ☑ CSRF Protected

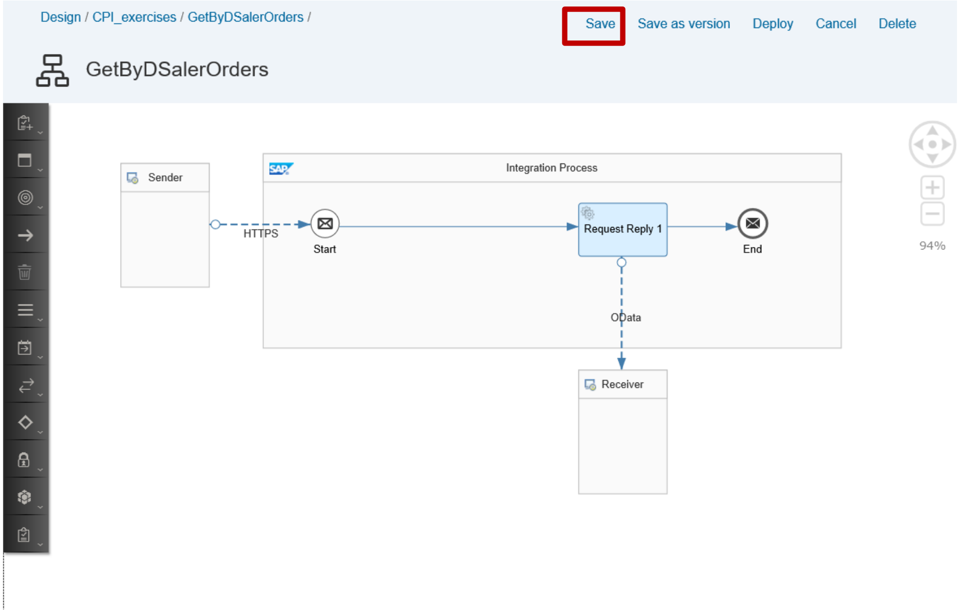
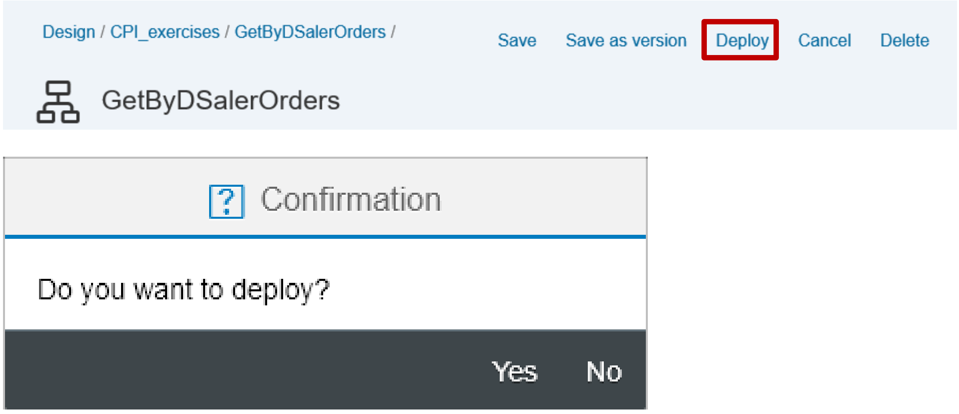
Explanation	Screenshot
- Internet as proxy type. - Basic authentication Enter the User Credentials name created in Store B1 and/or ByD User Credentials at SAP Cloud Platform Integration tenant prerequisite.	PS: If you have no access to a ByD system you can test with the following address not requiring authentication: https://espmrefapps.hana.ondemand.com/espm-cloud-web/espm.svc
For B1: Go to the **Connection** tab of the OData adapter and enter your B1 Service Layer **Address**: https://MyB1.emea.businessone.cloud.sap/b1s/v2/ Select : - **Internet** as proxy type. - **Basic authentication** Enter the **User Credentials name created** in Store B1 and/or ByD User Credentials at SAP Cloud Platform Integration tenant prerequisite.	 CONNECTION DETAILS Address: https://[redacted].emea.businessone.cloud.sap/b1s/v2/ Proxy Type: Internet Authentication: Basic Credential Name: B1user ☑ Allow Chunking Timeout (in min): 1 PS: If you have no access to a B1 system you can test with the following address not requiring authentication: https://espmrefapps.hana.ondemand.com/espm-cloud-web/espm.svc

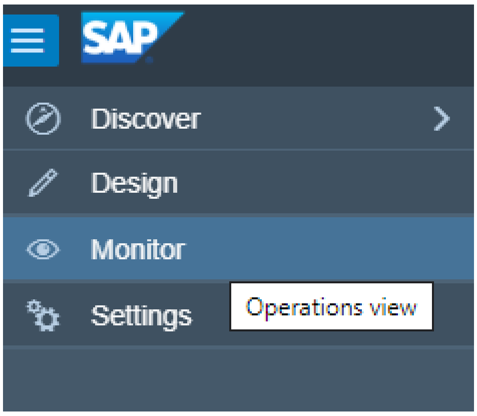
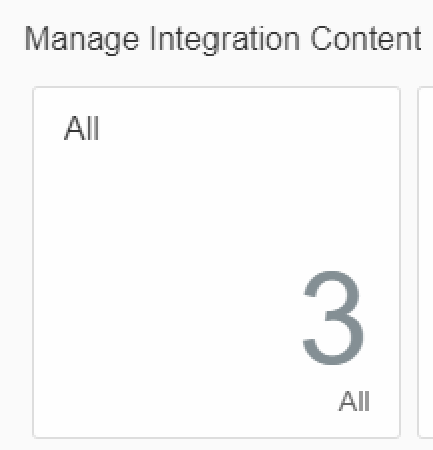
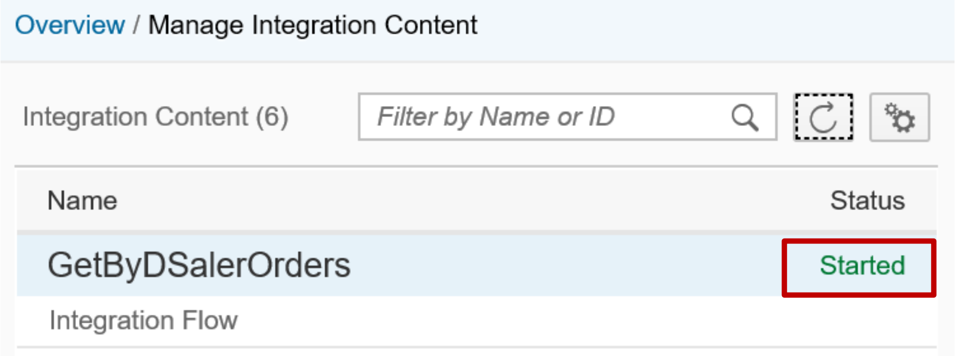
Explanation	Screenshot
For B1: Go to the **Processing** tab. Enter **Orders** as Resources Path. Enter **$top=50** as Query Options. **The $top option is highly recommended.** If not entered and your system contains many Orders CPI will try to retrieve them all (with SL pagination of 20 Orders per request) and you might get a timeout or block the flow due to the huge amount of data to be retrieved.	 PROCESSING DETAILS Operation: Query(GET) Resource Path: Orders Query Options: $top=50 HEADER DETAILS Request Headers: Response Headers:
For ByD: Go to the **Processing** tab. Next to **Resource Path**, choose **Select**.	 OData Externalize General Connection Processing Select PROCESSING DETAILS *Operation Details: Query (GET) *Resource Path: Query Options:
For ByD: The Query Editor opens. The **Address** field is already populated. Make sure that **Remote** is selected as the Connection Source and Authentication is set to **Basic**. Press **Step 2**.	 1. Connect to System *Connection Source : Remote *Address : https://[redacted].sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder *Proxy Type : Internet *Authentication : Basic ☑ Use Credential Name *Credential Name : BYD_ADMIN

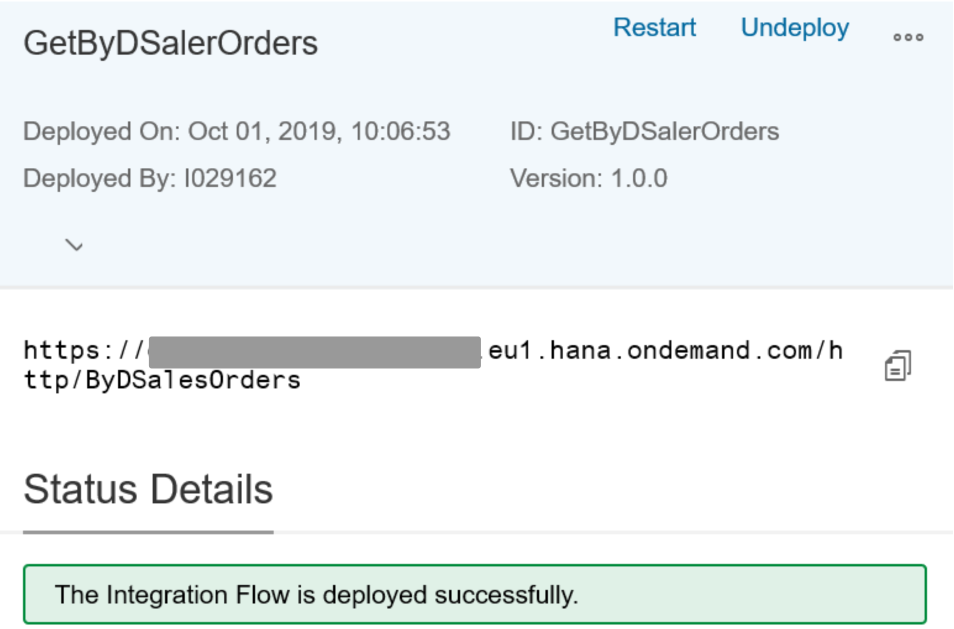
Explanation	Screenshot
For ByD: The system connects to the **hksalesorder** ByD oData service and retrieves the metadata from its OData API. Choose the Search icon in the Select Entity field and choose **SalesOrderCollection**. Change the number of **Sub Levels** to 2, you can then select sub collection properties like Item, BuyerParty,… Select the **Fields** you want to retrieve. You can set a Top value to reduce the size of the response. Press **Step 3** button.	 Model Operation 1 Connect to System — 2 Select Entity & Define Operation *Operation : Query (GET) Sub Levels : 2 *Select Entity : SalesOrderCollection □ Select All Fields Fields □ DataOriginTypeCodeText □ SalesOrganisationID □ DistributionChannelCode □ DistributionChannelCodeText □ PostingDateTime □ ConsistencyStatusCode □ ConsistencyStatusCodeText □ ReleaseStatusCode □ ReleaseStatusCodeText □ DeliveryStatusCode Top : 1 Skip : Query Copy SalesOrderCollection?$select=UUID,ID,Name,ExternalReference,BuyerParty/PartyID,Item/ItemProduct/ProductID,Item/ItemScheduleLine/Quantity&$expand=BuyerParty,Item/ItemProduct,Item,Item/ItemScheduleLine&$top=1

Explanation	Screenshot
For ByD: Press **Finish** on the last step.	 Model Operation 1 Connect to System 2 Select Entity & Define Operation 3 Configure Filter & Sorting Remove All Remove All Copy Finish Cancel 3. Configure Filter & Sorting Filter By: Select field / Lesser Than or Equal Sort By: Select field / Ascending Query SalesOrderCollection?$select=UUID,ID,Name,ExternalReference,BuyerParty/PartyID,Item/ItemProduct/ProductID,Item/ItemScheduleLine/Quantity&$expand=BuyerParty,Item/ItemProduct,Item,Item/ItemScheduleLine&$top=1

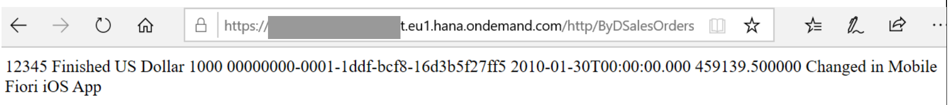
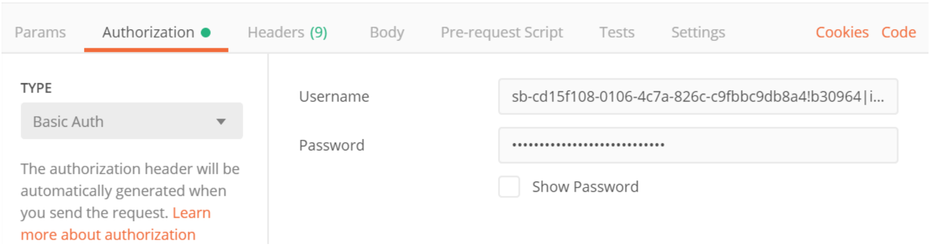
iv. Deploy the integration flow

Explanation	Screenshot
Click **Save**. On successful save, a corresponding status message is displayed.	 Design / CPI_exercises / GetByDSalerOrders / Save Save as version Deploy Cancel Delete GetByDSalerOrders Sender Integration Process HTTPS Start Request Reply 1 End OData Receiver 94%
Click **Deploy**. A message is displayed that asks you to confirm this action.	 Design / CPI_exercises / GetByDSalerOrders / Save Save as version Deploy Cancel Delete GetByDSalerOrders Confirmation Do you want to deploy?

Explanation	Screenshot
Chose the Operations/Monitor view to check the status of the deployment.	 SAP Discover Design Monitor Settings Operations view
Click the first tile in the **Manage Integration Content** section.	
You can check the deployment status of your integration flow. At the beginning the status of your flow will be **Starting** and afterwards move to **Started**.	 Overview / Manage Integration Content Integration Content (6) Name · Status GetByDSalerOrders · Started Integration Flow

Explanation	Screenshot
Copy the endpoint URL to the clipboard. The URL should end with `/http/<Address specified in the HTTPS adapter>.`	 GetByDSalerOrders Deployed On: Oct 01, 2019, 10:06:53 ID: GetByDSalerOrders Deployed By: I029162 Version: 1.0.0 `https://[redacted]eu1.hana.ondemand.com/http/ByDSalesOrders` Status Details The Integration Flow is deployed successfully.

v. Test your integration flow

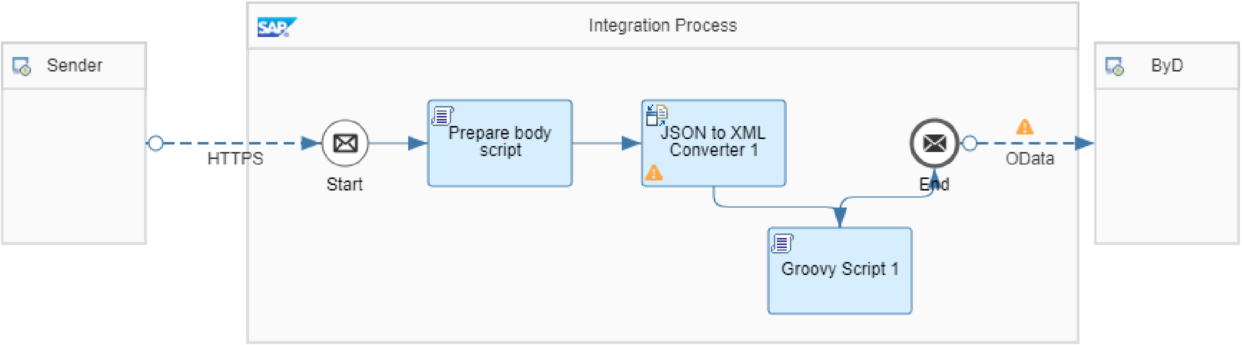
Explanation	Screenshot
Using an HTTP client, send a GET request to the endpoint address. You will get details of your Sales Orders in ByD.	 https://[redacted]t.eu1.hana.ondemand.com/http/ByDSalesOrders 12345 Finished US Dollar 1000 00000000-0001-1ddf-bcf8-16d3b5f27ff5 2010-01-30T00:00:00.000 459139.500000 Changed in Mobile Fiori iOS App
Enter your clientid and clientsecret on your http client.	 Params Authorization ● Headers (9) Body Pre-request Script Tests Settings Cookies Code TYPE: Basic Auth The authorization header will be automatically generated when you send the request. Learn more about authorization Username: sb-cd15f108-0106-4c7a-826c-c9fbbc9db8a4!b30964|i... Password: •••••••••••••••••••••••••••• □ Show Password
If you don’t know your clientid and clientsecret please check that you have previously generated	Authorize the HTTP Client to Call the Integration Flow Endpoint in the Cloud Foundry Environment:

Explanation	Screenshot
a **Process Integration Runtime** instance and its corresponding **Service Key**.	https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US/7ac1d3ab712a49f793e0a2af168ac22d.html
Note1: If you CPI instance is running on a Neo environment the user and password will be simply your CPI account user and password.	Send the HTTP Request and Process the Integration Flow: https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US/f08cca6af4fd48b687d3f86c329efaed.html
Note2: If you have set the CRSF Protected flag for your HTTPS connection, please follow the instructions here to configure your http sender.	Send the HTTP Request and Process the Integration Flow: https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US/f08cca6af4fd48b687d3f86c329efaed.html

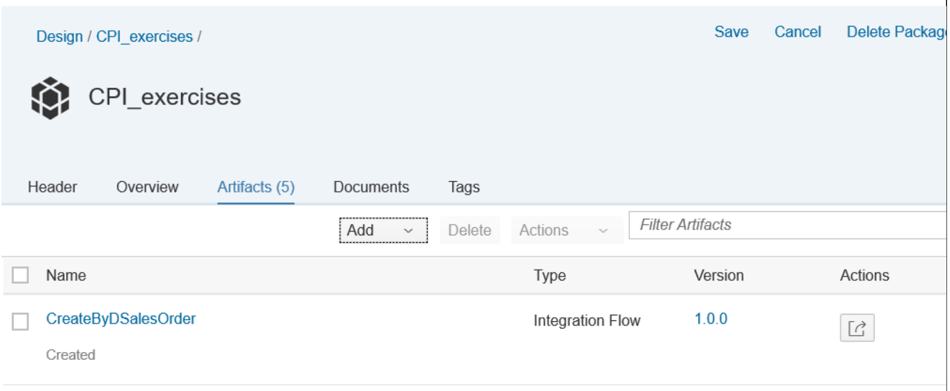
STEP 5: HTTPS SENDER INITIATED SCENARIO, POST REQUEST

Implementation of an integration scenario initiated by an HTTPS sender (like the previous scenario) sending a SalesOrder POST request sent to SAP Business ByDesign or SAP Business One.

The final flow will look like this:

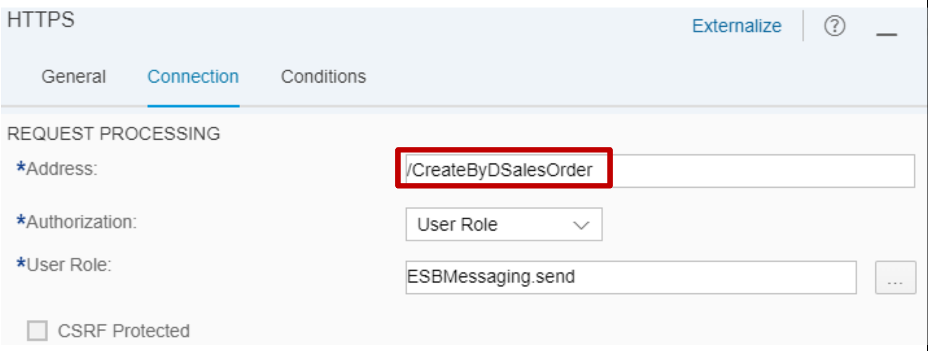


i. Create the Integration Flow

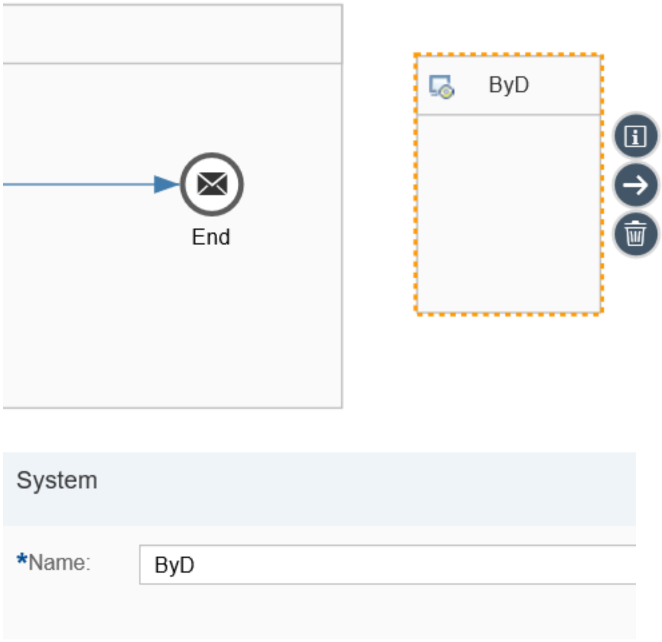
Explanation	Screenshot
Create a new Integration Flow by following STEP 2: CREATE AN INTEGRATION FLOW. Call the new Integration Flow something like: **CreateByDSalesOrder** or **CreateB1SalesOrder**.	 Design / CPI_exercises / Save Cancel Delete Packag CPI_exercises Header Overview Artifacts (5) Documents Tags Add Delete Actions Filter Artifacts □ Name · Type · Version · Actions □ CreateByDSalesOrder · Created · Integration Flow · 1.0.0

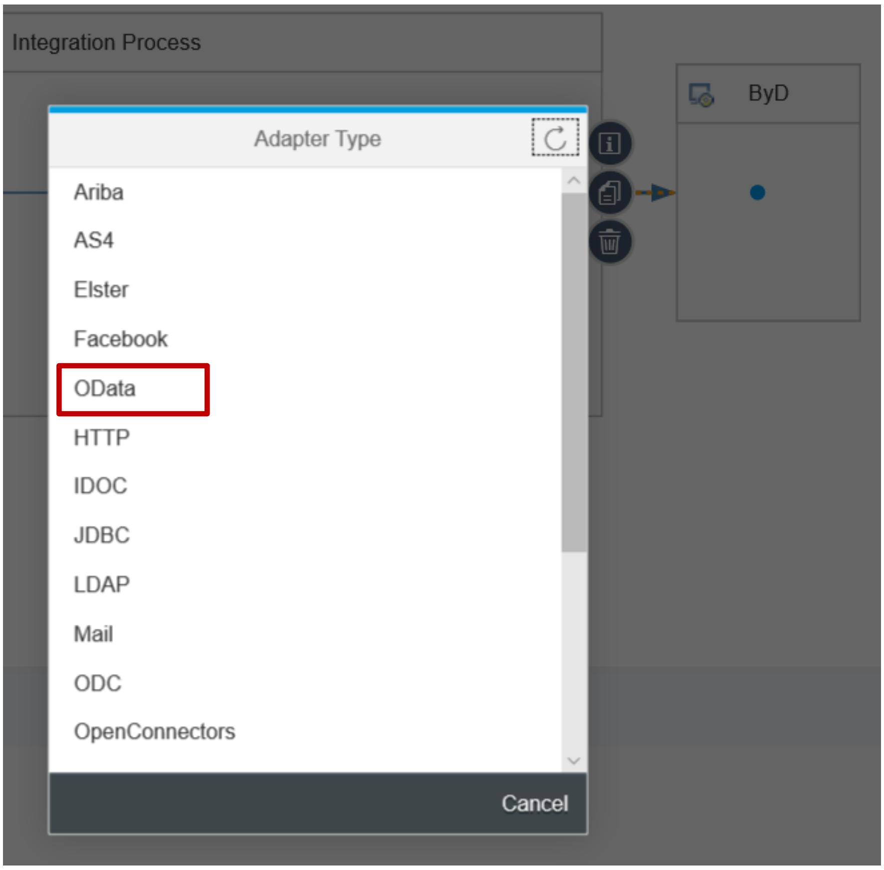
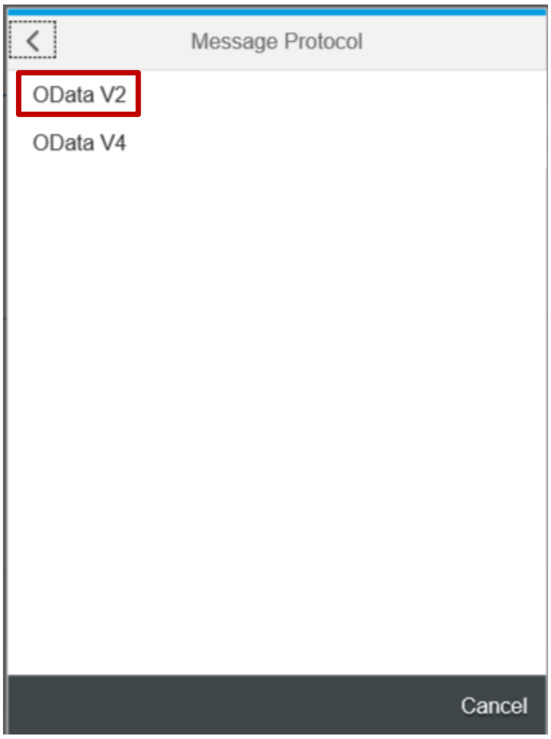
ii. Create the HTTPS Sender Channel

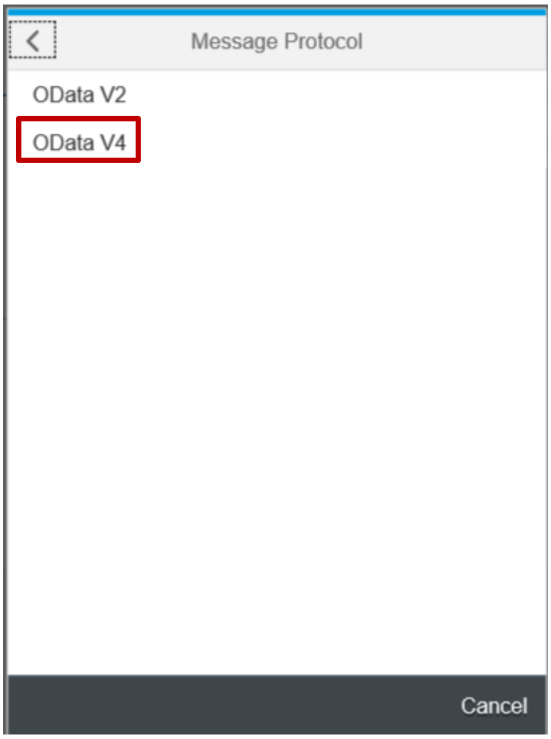
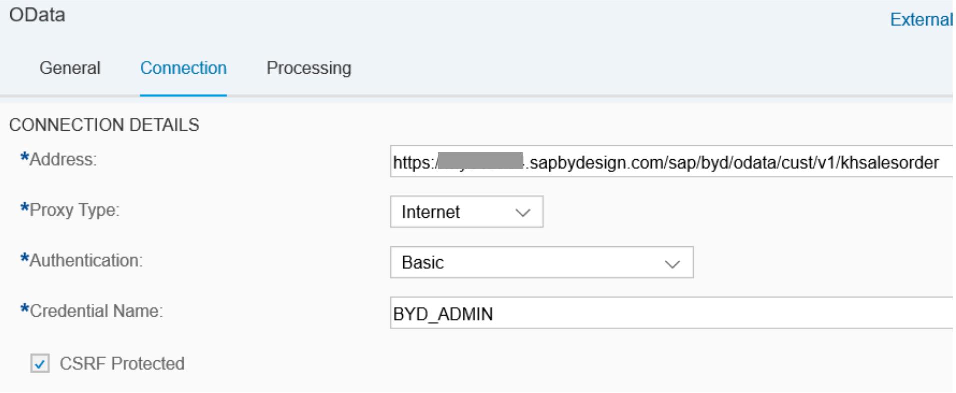
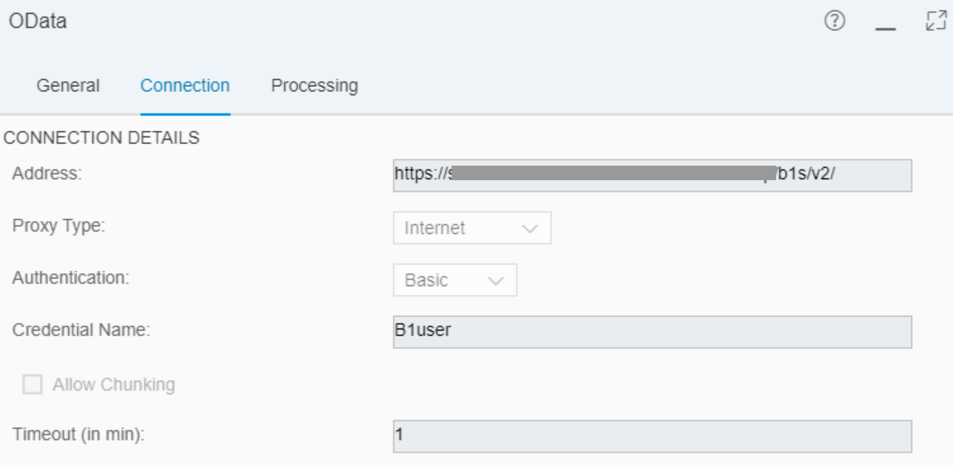
Add an HTTPS sender channel to enable the integration flow to receive HTTP requests.

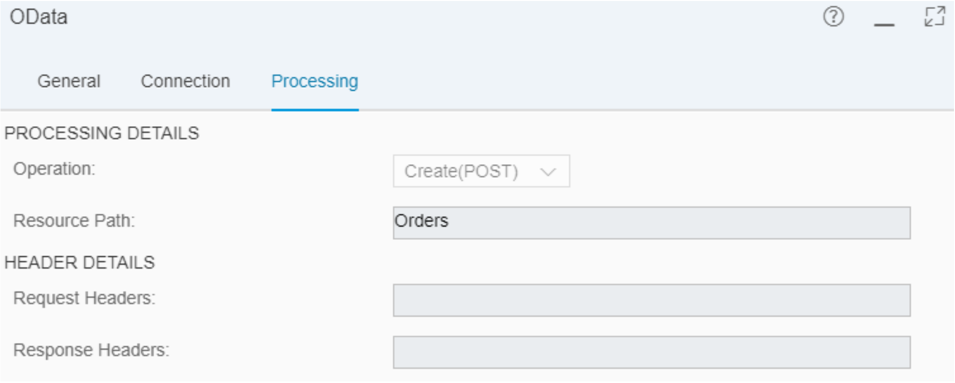
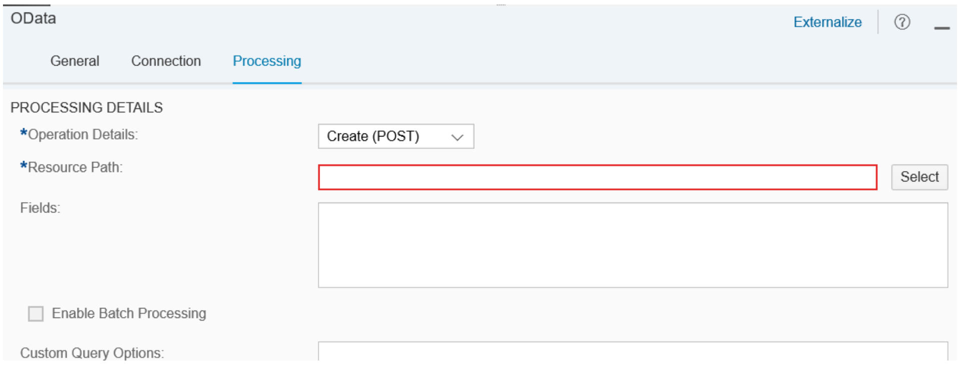
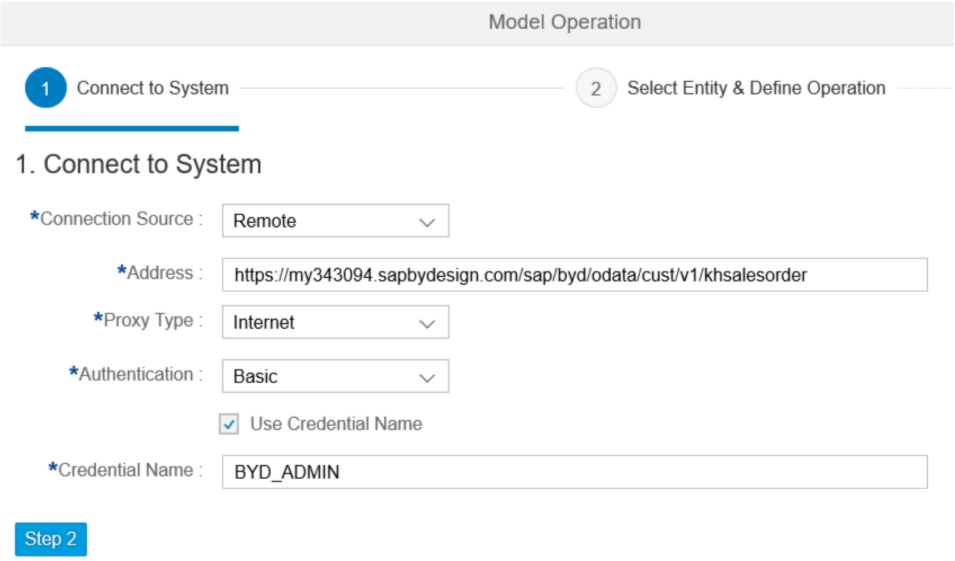
Explanation	Screenshot
Follow the steps at Create the HTTPS Sender Channel. Enter **/CreateByDSalesOrder** or **/CreateB1SalesOrder** as address. Uncheck the flag CSRF Protected.	 HTTPS General · Connection · Conditions REQUEST PROCESSING *Address: /CreateByDSalesOrder *Authorization: User Role *User Role: ESBMessaging.send □ CSRF Protected

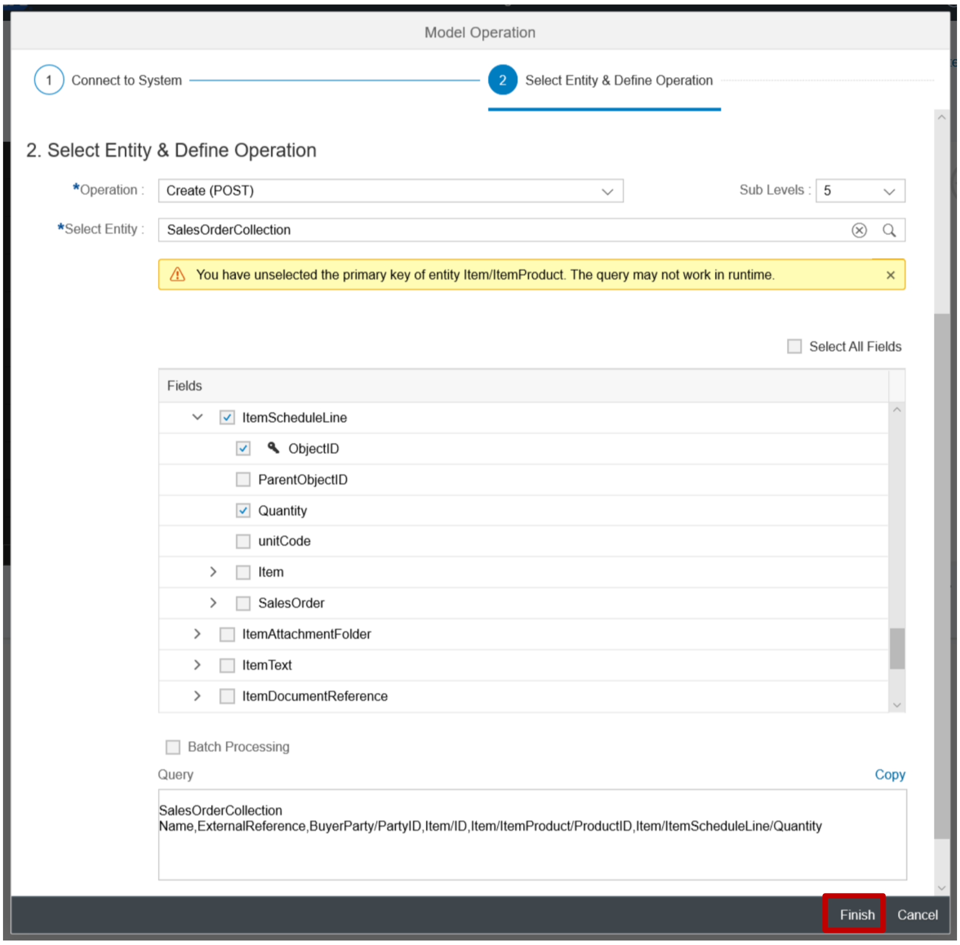
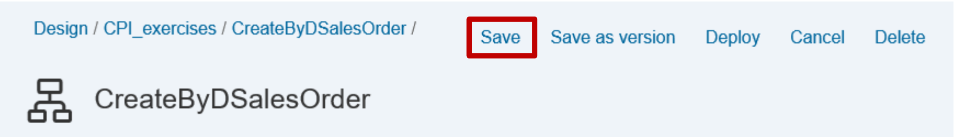
iii. Configure the B1/ByD Receiver

Explanation	Screenshot
Select the Receiver system and change its name to **ByD** or **B1**.	 System *Name: ByD

Explanation	Screenshot
Connect the End event shape to the ByD system. In the next dialog, choose adapter type **OData**.	 Integration Process ByD Adapter Type Ariba AS4 Elster Facebook OData HTTP IDOC JDBC LDAP Mail ODC OpenConnectors Cancel
<span>**For ByD:**</span> Select **OData V2** as Message Protocol.	

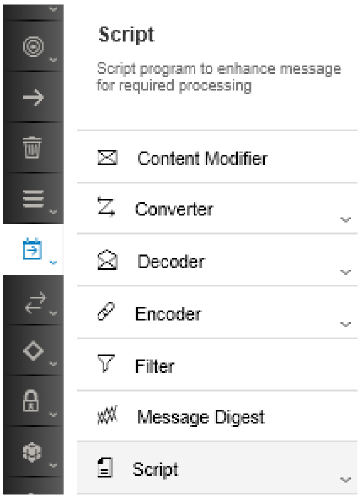
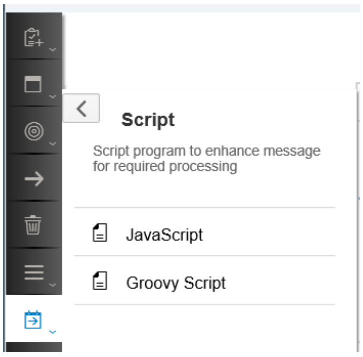
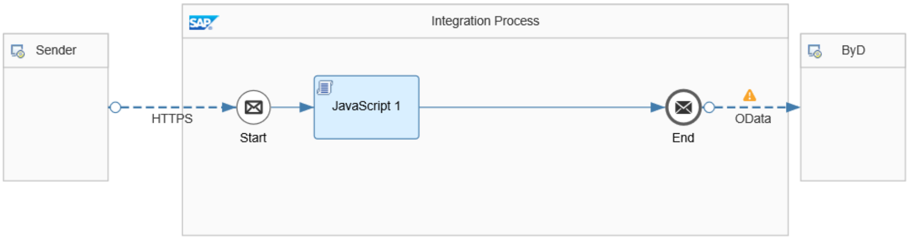
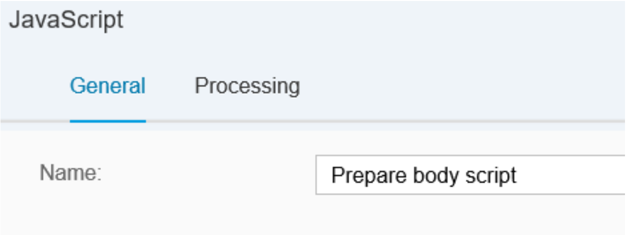
Explanation	Screenshot
For B1: Select **OData V4** as Message Protocol.	 Message Protocol OData V2 OData V4 Cancel
Go to the **Connection** tab of the OData adapter and enter the following as the **Address**: **For ByD:** https://MY_BYD_TENANT/sap/byd/odata/cust/v1/khsalesorder **For B1:** https://BY_B1_BACKEND/b1s/v2/ Select **Internet** as proxy type and **Basic Authentication.** Enter the User Credentials name created in Store B1 and/or ByD User Credentials at SAP Cloud Platform Integration tenant prerequisite.	 OData External General Connection Processing CONNECTION DETAILS *Address: https:/[redacted].sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder *Proxy Type: Internet *Authentication: Basic *Credential Name: BYD_ADMIN ☑ CSRF Protected  OData General Connection Processing CONNECTION DETAILS Address: https://[redacted]b1s/v2/ Proxy Type: Internet Authentication: Basic Credential Name: B1user □ Allow Chunking Timeout (in min): 1

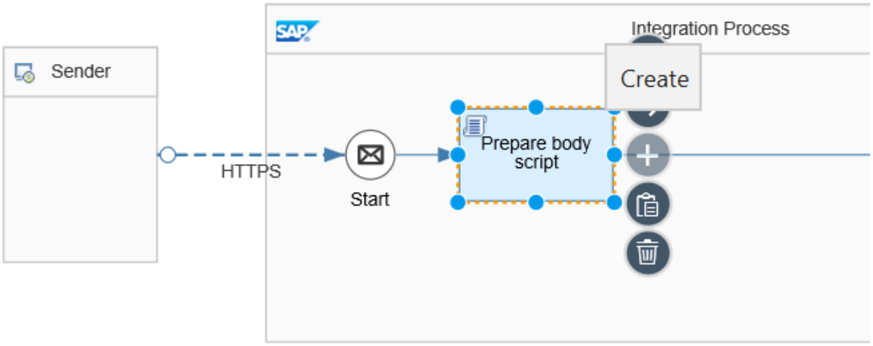
Explanation	Screenshot
For B1: Go to the **Processing** tab. Change the Operation Details to **Create (POST).** Enter **Orders** as Resource Path.	 OData General Connection Processing PROCESSING DETAILS Operation: Create(POST) Resource Path: Orders HEADER DETAILS Request Headers: Response Headers:
For ByD: Go to the **Processing** tab. Change the Operation Details to **Create (POST).** Make sure Content Type is set to **Atom**. Next to **Resource Path**, choose **Select**.	 OData Externalize General Connection Processing PROCESSING DETAILS *Operation Details: Create (POST) *Resource Path: Select Fields: □ Enable Batch Processing Custom Query Options:
For ByD: The Query Editor opens. The **Address** field is already populated. Make sure that **Remote** is selected as the Connection Source and Authentication is set to **Basic**. Press **Step 2**.	 Model Operation 1 Connect to System 2 Select Entity & Define Operation 1. Connect to System *Connection Source : Remote *Address : https://my343094.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder *Proxy Type : Internet *Authentication : Basic ☑ Use Credential Name *Credential Name : BYD_ADMIN Step 2

Explanation	Screenshot
For ByD: The system connects to the **hksalesorder** ByD oData service and retrieves the metadata from its OData API. Choose the Search icon in the Select Entity field and choose **SalesOrderCollection**. Change the number of **Sub Levels** to 5. Select the following **Fields**: - Name - ExternalReference - BuyerParty/PartyID - Item/ID - Item/ItemProduct /ProductID - Item /ItemScheduleLine /Quantity Press **Finish**.	 Model Operation 1 Connect to System — 2 Select Entity & Define Operation 2. Select Entity & Define Operation *Operation : Create (POST) Sub Levels : 5 *Select Entity : SalesOrderCollection You have unselected the primary key of entity Item/ItemProduct. The query may not work in runtime. □ Select All Fields Fields ☑ ItemScheduleLine ☑ ObjectID □ ParentObjectID ☑ Quantity □ unitCode □ Item □ SalesOrder □ ItemAttachmentFolder □ ItemText □ ItemDocumentReference □ Batch Processing Query Copy SalesOrderCollection Name,ExternalReference,BuyerParty/PartyID,Item/ID,Item/ItemProduct/ProductID,Item/ItemScheduleLine/Quantity Finish Cancel
Press **Save**.	 Design / CPI_exercises / CreateByDSalesOrder / Save Save as version Deploy Cancel Delete CreateByDSalesOrder

iv. Create the Script Step

Add a Script Step that will read the Sender request body and prepare the ByD Sales Order body.

Explanation	Screenshot
Go to the palette, choose the **Message Transformers** icon and select the **Script** icon.	 Script Script program to enhance message for required processing Content Modifier Converter Decoder Encoder Filter Message Digest Script
In the **Script** submenu, select **JavaScript**.	 Script Script program to enhance message for required processing
Place the Script Step shape between the Start and End event.	 Integration Process Sender HTTPS Start JavaScript 1 End OData ByD
In the Script properties, go to the General tab and change the Name to **Prepare body script**.	 JavaScript Name: Prepare body script

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Select the Script shape and press the + icon to create a new script file.</td><td>
</td></tr>
<tr><td><b>For ByD:</b>
The JavaScript file opens with some default code.
Replace the code to with the proposed here.
Press <b>OK</b> to close the code editor.</td><td><pre>
importClass(com.sap.gateway.ip.core.customdev.util.Message);
importClass(java.util.HashMap);
function processData(message) {
  var body = String(message.getBody(new
java.lang.String().getClass()));
  //create json object from string object
  body = JSON.parse(body);

  // get data from the HTTPS request body
  var name = body.Name;
  var buyerPartyID = body.BuyerPartyID;
  var reference = body.Reference;
  var productID1 = body.ProductID1;
  var quantity1 = body.Quantity1;
  var productID2 = body.ProductID2;
  var quantity2 = body.Quantity2;

  // create the ByD SalesOrder request body
  var soBody = {
    SalesOrder: {
      ExternalReference: "FromCPI" + reference,
      Name: name,
      DataOriginTypeCode: "1",
      BuyerParty: { BuyerParty: {PartyID: buyerPartyID }},
      Item: [{
        Item: {
          ID: "10",
          ItemProduct: {
            ItemProduct: {ProductID: productID1}
          },
          ItemScheduleLine: [{
            ItemScheduleLine: {Quantity: quantity1}
          }]
        }
      },
      {
        Item: {
          ID: "20",
          ItemProduct: {
            ItemProduct: {ProductID: productID2}
          },
          ItemScheduleLine: [{
            ItemScheduleLine: {Quantity: quantity2}
          }]
        }
      }]
</pre></td></tr>
</table>

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td></td><td>

```
   }
 };

  message.setBody(JSON.stringify(soBody));

  return message;
}
```

</td></tr>
<tr><td>

For B1:

The JavaScript file opens with some default code.

Replace the code to with the proposed here.

Press **OK** to close the code editor.

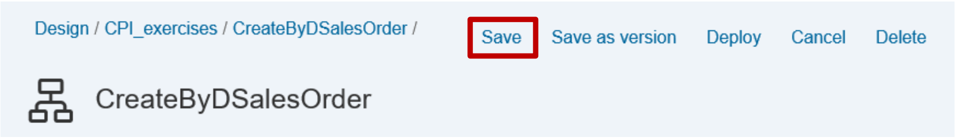
</td><td>

```
importClass(com.sap.gateway.ip.core.customdev.util.Message);
importClass(java.util.HashMap);
function processData(message) {
  var body = String(message.getBody(new
java.lang.String().getClass()));
  //create json object from string object
  body = JSON.parse(body);

  // get data from the HTTPS request body
  var buyerPartyID = body.BuyerPartyID;
  var reference = body.Reference;
  var productID1 = body.ProductID1;
  var quantity1 = body.Quantity1;
  var productID2 = body.ProductID2;
  var quantity2 = body.Quantity2;

  var soBody = {
    Document: {
      Comments: "FromCPI" + reference,
      DocDueDate: "2020-01-05T00:00:00", // + reference,
      CardCode: buyerPartyID,
      DocumentLines: [{
        DocumentLine: {
          ItemCode: productID1,
          Quantity: quantity1
        }
       },
       {
         DocumentLine: {
           ItemCode: productID2,
           Quantity: quantity2
         }
       }]
     }
   };


  message.setBody(JSON.stringify(soBody));

  return message;
}
```

</td></tr>
<tr><td>

Press **Save**.

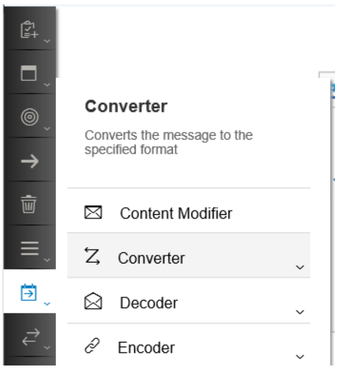
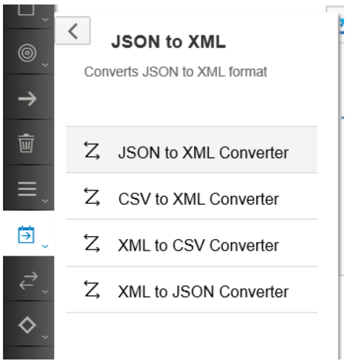
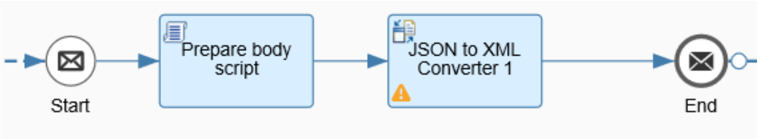
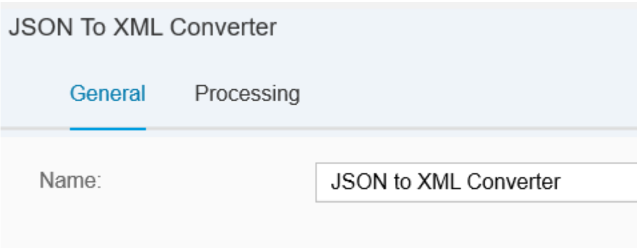
</td><td>

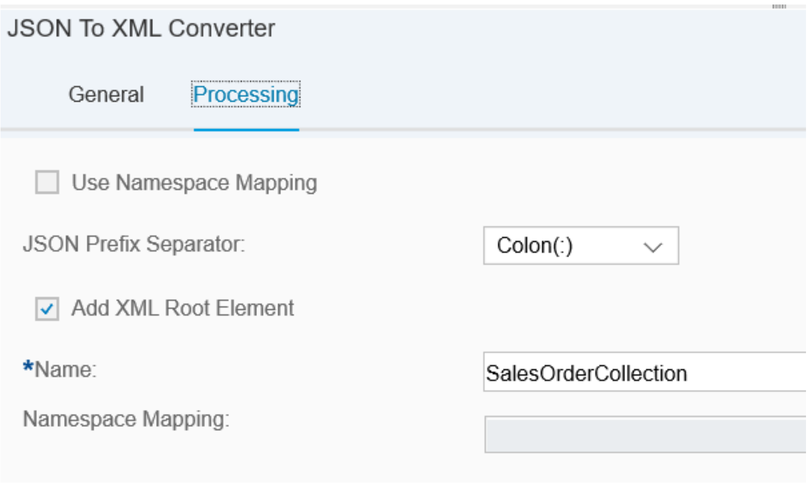
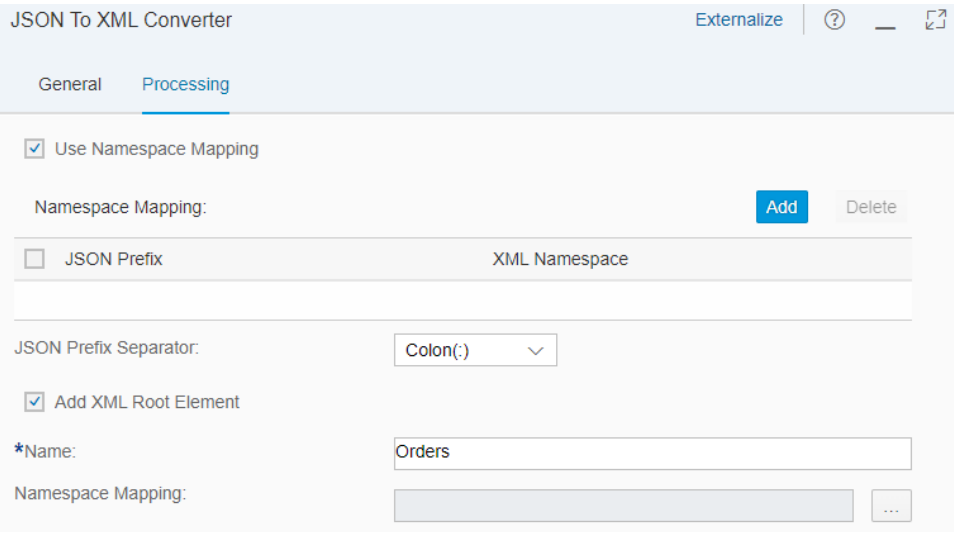
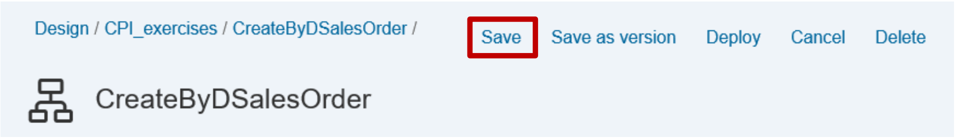


</td></tr>
</table>

v. Add a Converter JSON to XML format

OData receivers only accept XML format, we need to transform our JSON body to XML format.

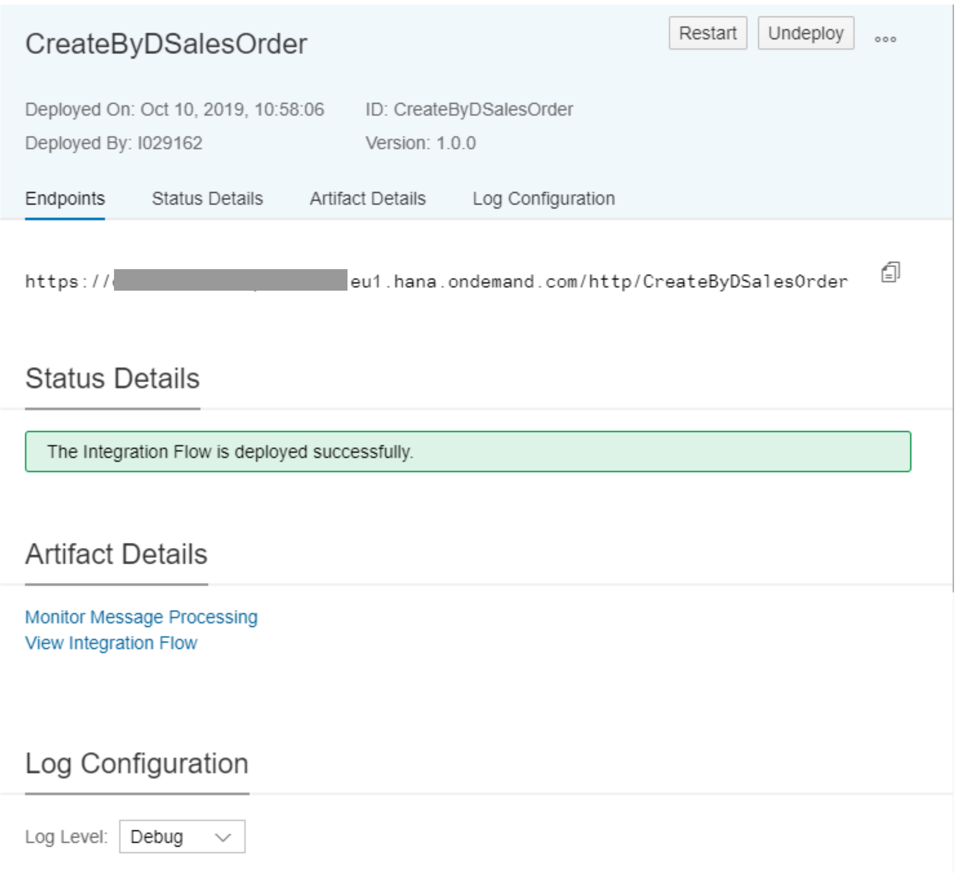
Explanation	Screenshot
Go to the palette, choose the **Message Transformers** icon and select the **Converter** icon.	 **Converter** Converts the message to the specified format Content Modifier Converter Decoder Encoder
In the **Converter** submenu, select **JSON to XML Converter**.	 **JSON to XML** Converts JSON to XML format JSON to XML Converter CSV to XML Converter XML to CSV Converter XML to JSON Converter
Place the JSON to XML converter Step shape between the Prepare body script and End event shapes.	 Start Prepare body script JSON to XML Converter 1 End
In the JSON to XML Converter properties, go to the General tab and change the Name to **JSON to XML Converter**.	 JSON To XML Converter General Processing Name: JSON to XML Converter

Explanation	Screenshot
For ByD: In the Processing tab check the **Add XML Root Element**. Enter the Name **SalesOrderCollection**. Make sure the JSON Prefix Separator is set to **Colon(:)**.	 JSON To XML Converter □ Use Namespace Mapping JSON Prefix Separator: Colon(:) ☑ Add XML Root Element *Name: SalesOrderCollection Namespace Mapping:
For B1: In the Processing tab check the **Add XML Root Element**. Enter the Name **Orders**. Make sure the JSON Prefix Separator is set to **Colon(:)**.	 JSON To XML Converter ☑ Use Namespace Mapping Namespace Mapping: □ JSON Prefix | XML Namespace JSON Prefix Separator: Colon(:) ☑ Add XML Root Element *Name: Orders Namespace Mapping:
Press **Save**.	 Design / CPI_exercises / CreateByDSalesOrder / CreateByDSalesOrder

vi. (Optional) Create a Script Step to Log the Payload

Follow the same steps as we did at Create a Script Step to Log the Payload.

vii. Deploy the integration flow

Explanation	Screenshot
Follow the steps proposed in the previous exercise to Deploy the integration flow CreateByDSalesOrder.	

CreateByDSalesOrder

Deployed On: Oct 10, 2019, 10:58:06

ID: CreateByDSalesOrder

Deployed By: I029162

Version: 1.0.0

```
https://[redacted]eu1.hana.ondemand.com/http/CreateByDSalesOrder
```

Status Details

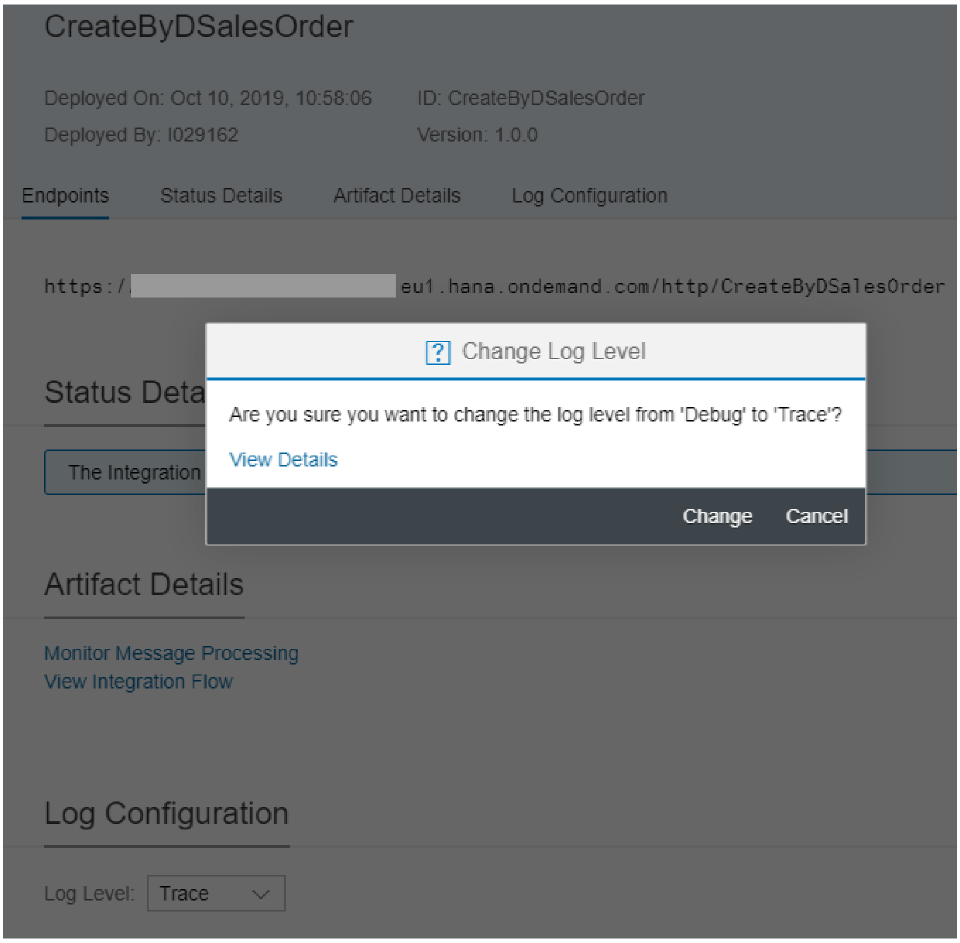
The Integration Flow is deployed successfully.

Artifact Details

Monitor Message Processing
View Integration Flow

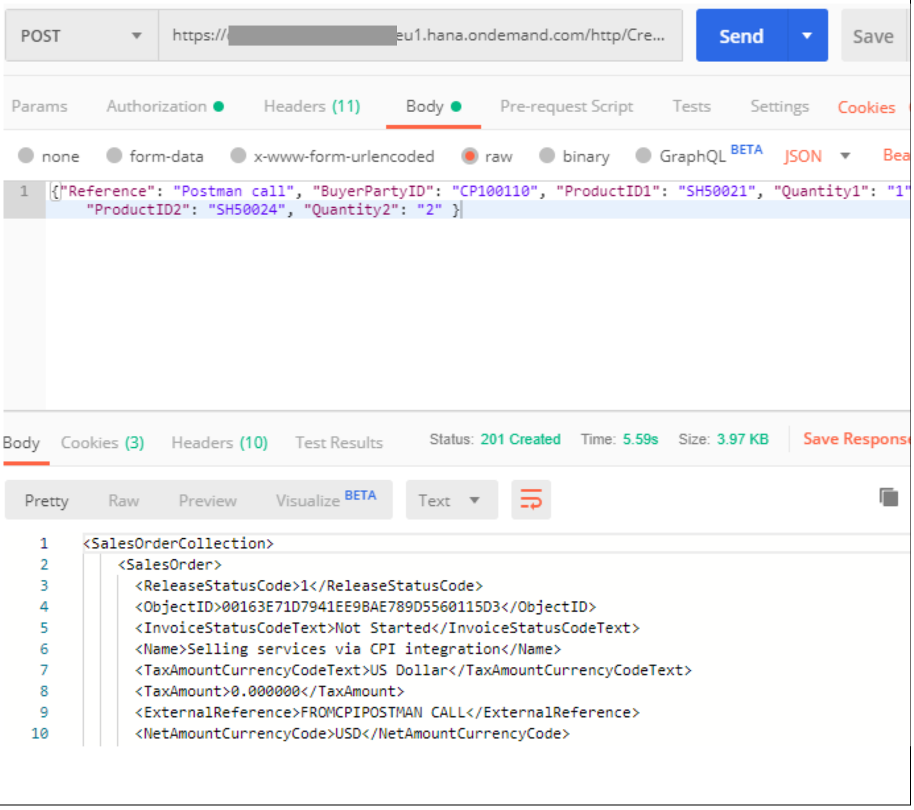
Log Configuration

Log Level: Debug

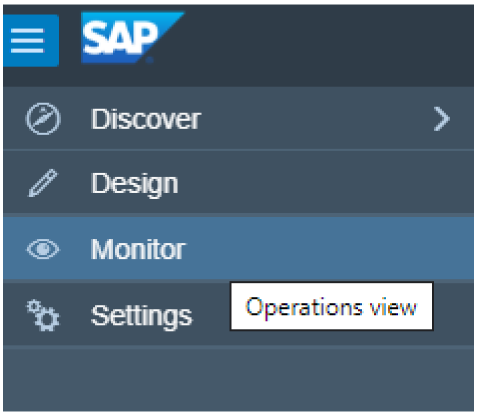
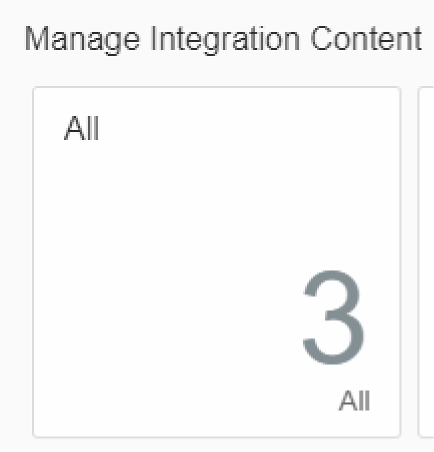
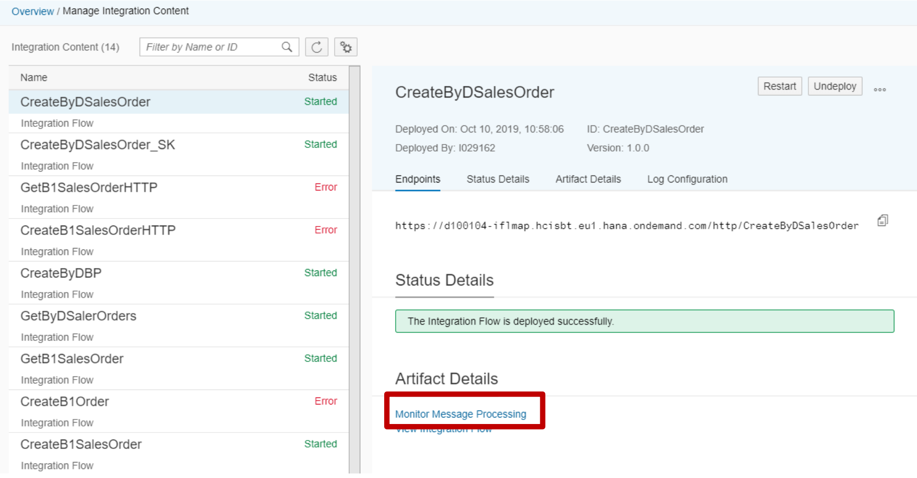
Explanation	Screenshot
If you want to explore all the details of your flow you can change the Log Level to “Trace”. This level will remain active for some minutes only to avoid performances issues.	 CreateByDSalesOrder Deployed On: Oct 10, 2019, 10:58:06 ID: CreateByDSalesOrder Deployed By: I029162 Version: 1.0.0 Endpoints Status Details Artifact Details Log Configuration https:/[redacted]eu1.hana.ondemand.com/http/CreateByDSalesOrder Status Deta The Integration Change Log Level Are you sure you want to change the log level from 'Debug' to 'Trace'? View Details Artifact Details Monitor Message Processing View Integration Flow Log Configuration Log Level: Trace

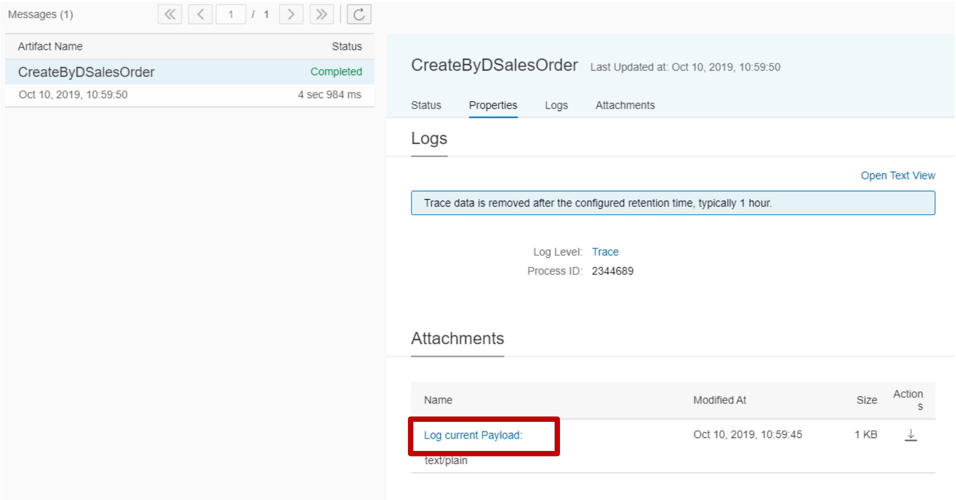
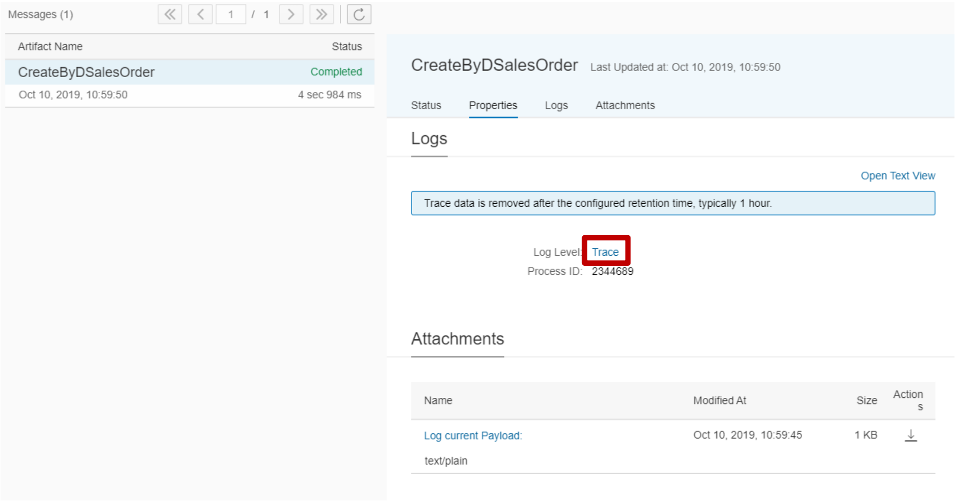
viii. Test your integration flow

Explanation	Screenshot

Using an HTTP client, send a POST request to the endpoint address.
With a body like:
{
"Name": "SO creation via CPI",
"Reference": "Postman call",
"BuyerPartyID": "CP100110",
"ProductID1": "SH50021",
"Quantity1": "1",
"ProductID2": "SH50024",
"Quantity2": "2"
}
You will create a new Sales Order and get the response message from B1/ByD.
	 Send
	Save
	Params
	Authorization
	Headers (11)
	Body
	Pre-request Script
	Tests
	Settings
	Cookies
	none
	form-data
	x-www-form-urlencoded
	binary
	GraphQL BETA
	Bea
	Body
	Cookies (3)
	Headers (10)
	Test Results
	Save Respons
	Pretty
	Raw
	Preview
	Visualize BETAPOST https://[redacted]eu1.hana.ondemand.com/http/Cre...
	Body: ● raw JSON
	`{"Reference": "Postman call", "BuyerPartyID": "CP100110", "ProductID1": "SH50021", "Quantity1": "1"`
	`"ProductID2": "SH50024", "Quantity2": "2" }`
	Status: 201 Created Time: 5.59s Size: 3.97 KB
	Text
	`<SalesOrderCollection>`
	`  <SalesOrder>`
	`    <ReleaseStatusCode>1</ReleaseStatusCode>`
	`    <ObjectID>00163E71D7941EE9BAE789D5560115D3</ObjectID>`
	`    <InvoiceStatusCodeText>Not Started</InvoiceStatusCodeText>`
	`    <Name>Selling services via CPI integration</Name>`
	`    <TaxAmountCurrencyCodeText>US Dollar</TaxAmountCurrencyCodeText>`
	`    <TaxAmount>0.000000</TaxAmount>`
	`    <ExternalReference>FROMCPIPOSTMAN CALL</ExternalReference>`
	`    <NetAmountCurrencyCode>USD</NetAmountCurrencyCode>`

ix. Check the traces

Explanation	Screenshot
Chose the Operations/Monitor view to check the status of the deployment.	 SAP Discover Design Monitor Settings Operations view
Click the first tile in the **Manage Integration Content** section.	
In the new page **select your Integration Flow**. Press on the link **Monitor Message Processing**.	Overview / Manage Integration Content Integration Content (14) Name — Status CreateByDSalesOrder — Started — Integration Flow CreateByDSalesOrder_SK — Started — Integration Flow GetB1SalesOrderHTTP — Error — Integration Flow CreateB1SalesOrderHTTP — Error — Integration Flow CreateByDBP — Started — Integration Flow GetByDSalerOrders — Started — Integration Flow GetB1SalesOrder — Started — Integration Flow CreateB1Order — Error — Integration Flow CreateB1SalesOrder — Started — Integration Flow CreateByDSalesOrder Deployed On: Oct 10, 2019, 10:58:06 ID: CreateByDSalesOrder Deployed By: I029162 Version: 1.0.0 `https://d100104-iflmap.hcisbt.eu1.hana.ondemand.com/http/CreateByDSalesOrder` Status Details The Integration Flow is deployed successfully. Artifact Details Monitor Message Processing Filter by Name or ID Restart Undeploy Endpoints Status Details Artifact Details Log Configuration

Explanation	Screenshot
If you implemented the script for logging the payload, on the new page select **Attachments** section of your deployed flow you will be able to see the logged payload.	 Messages (1) 1 / 1 Artifact Name Status CreateByDSalesOrder Completed Oct 10, 2019, 10:59:50 4 sec 984 ms CreateByDSalesOrder Last Updated at: Oct 10, 2019, 10:59:50 Status Properties Logs Attachments Logs Open Text View Trace data is removed after the configured retention time, typically 1 hour. Log Level: Trace Process ID: 2344689 Attachments Name / Modified At / Size / Actions Log current Payload: text/plain / Oct 10, 2019, 10:59:45 / 1 KB
On the same page you can click on the **Trace** link to open a detailed trace page.	 Messages (1) 1 / 1 Artifact Name Status CreateByDSalesOrder Completed Oct 10, 2019, 10:59:50 4 sec 984 ms CreateByDSalesOrder Last Updated at: Oct 10, 2019, 10:59:50 Status Properties Logs Attachments Logs Open Text View Trace data is removed after the configured retention time, typically 1 hour. Log Level: Trace Process ID: 2344689 Attachments Name / Modified At / Size / Actions Log current Payload: text/plain / Oct 10, 2019, 10:59:45 / 1 KB

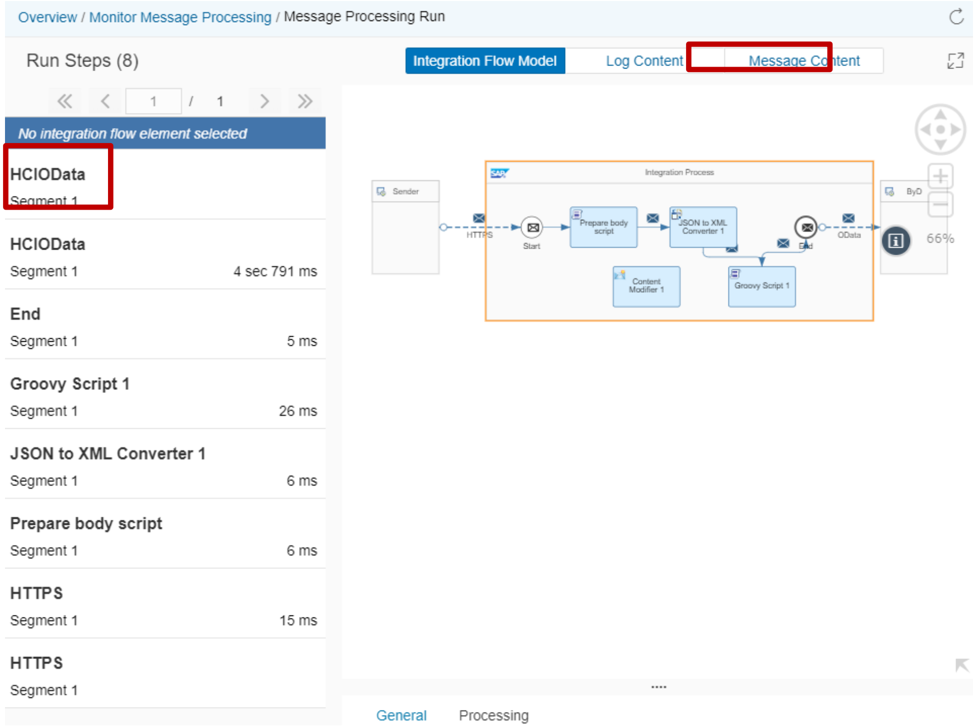
Explanation	Screenshot

In the new screen you will see all the steps run.

You can click on any of the steps to see detailed information.

Select for example the last Step on the top **HCIOdata**.

Click on the **Message Content**.



Overview / Monitor Message Processing / Message Processing Run

Run Steps (8)

1 / 1

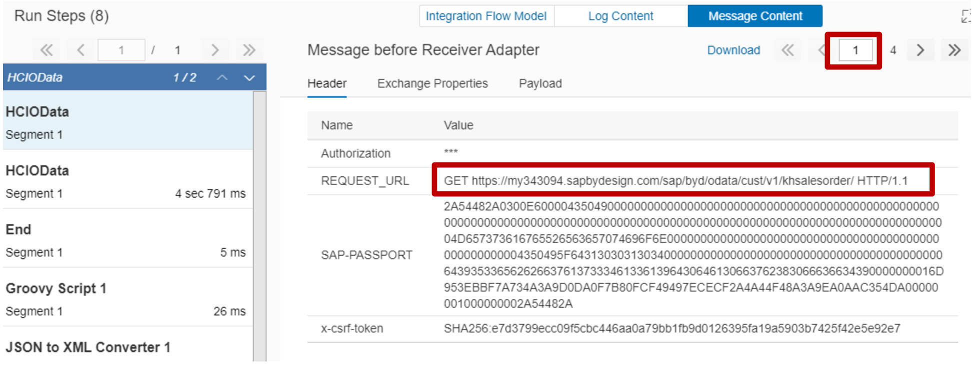
No integration flow element selected

HCIOData Segment 1	
HCIOData Segment 1	4 sec 791 ms
End Segment 1	5 ms
Groovy Script 1 Segment 1	26 ms
JSON to XML Converter 1 Segment 1	6 ms
Prepare body script Segment 1	6 ms
HTTPS Segment 1	15 ms
HTTPS Segment 1	

For ByD:

Select **page 1**.

You can see it is a GET request done automatically by CPI OData adapter in order to retrieve the csrf token.



Run Steps (8)

1 / 1

HCIOData 1 / 2

HCIOData Segment 1	
HCIOData Segment 1	4 sec 791 ms
End Segment 1	5 ms
Groovy Script 1 Segment 1	26 ms
JSON to XML Converter 1	

Message before Receiver Adapter

Download 1 4

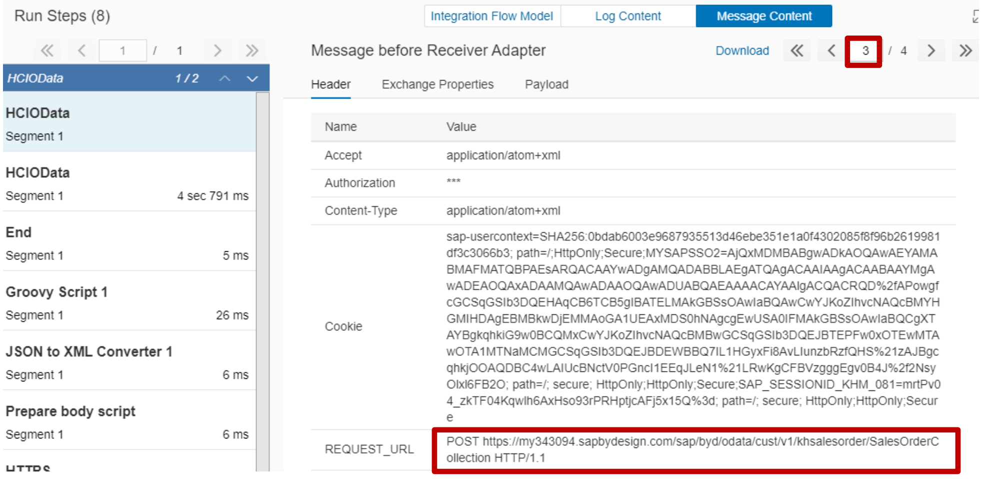
Header | Exchange Properties | Payload

Name	Value
Authorization	***
REQUEST_URL	GET https://my343094.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder/ HTTP/1.1
SAP-PASSPORT	2A54482A0300E6000043504900000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000004D6573736167655265636570746696F6E0000000000000000000000000000000000000000000000004350495F64313030313034000000000000000000000000000000000006439353365626266376137333461336139643064613066376238306663663439000000016D953EBBF7A734A3A9D0DA0F7B80FCF49497ECECF2A4A44F48A3A9EA0AAC354DA00000001000000002A54482A
x-csrf-token	SHA256:e7d3799ecc09f5cbc446aa0a79bb1fb9d0126395fa19a5903b7425f42e5e92e7

For ByD:

Select **page 3**.

It shows the POST request sent to ByD.



Run Steps (8)

1 / 1

HCIOData 1 / 2

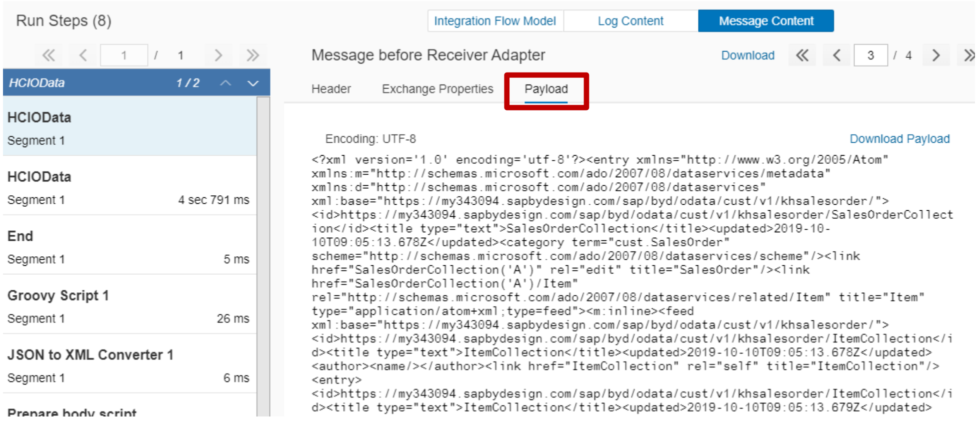
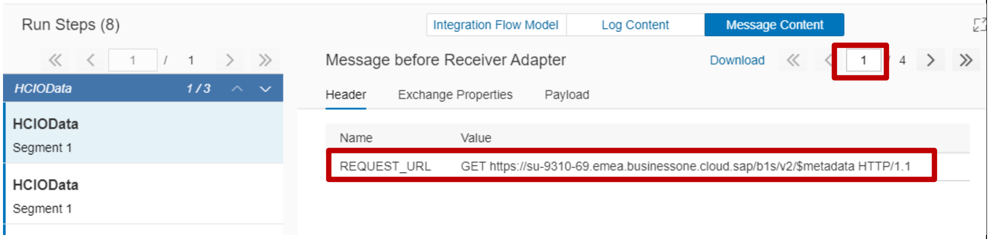
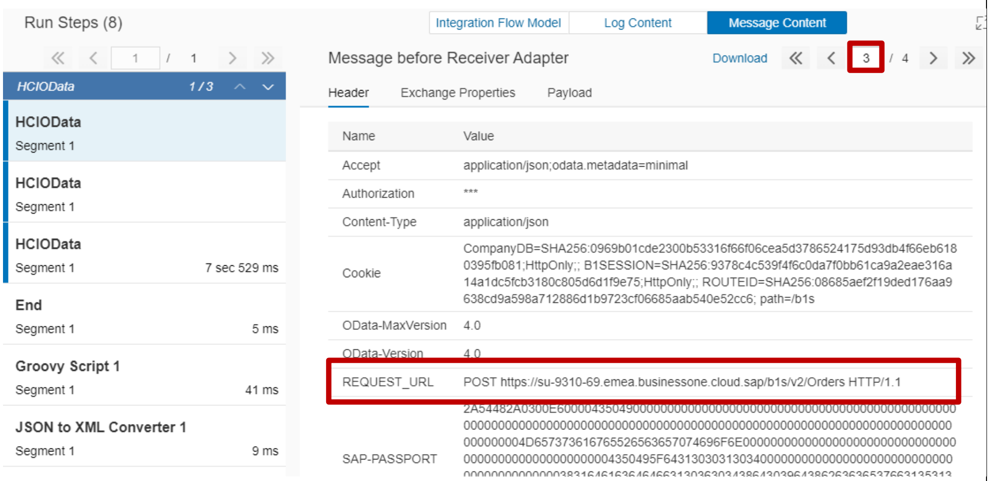
HCIOData Segment 1	
HCIOData Segment 1	4 sec 791 ms
End Segment 1	5 ms
Groovy Script 1 Segment 1	26 ms
JSON to XML Converter 1 Segment 1	6 ms
Prepare body script Segment 1	6 ms

Message before Receiver Adapter

Download 3 / 4

Header | Exchange Properties | Payload

Name	Value
Accept	application/atom+xml
Authorization	***
Content-Type	application/atom+xml
Cookie	sap-usercontext=SHA256:0bdab6003e9687935513d46ebe351e1a0f4302085f8f96b2619981df3c3066b3; path=/;HttpOnly;Secure;MYSAPSSO2=AjQxMDMBABgwADkAOQAwAEYAMABMAFMATQBPAEsARQACAAYwADgAMQADABBLAEgATQAgACAAIAAgACAABAAYMgAwADEAOQAxADAAMQAwADAAOQAwADUABQAEAAAACAYAAIgACQACRQD%2fAPowgfcGCSqGSIb3DQEHAqCB6TCB5gIBATELMAkGBSsOAwIaBQAwCwYJKoZIhvcNAQcBMYHGMIHDAgEBMBkwDjEMMAoGA1UEAxMDS0hNAgcgEwUSA0IFMAkGBSsOAwIaBQCgXTAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0xOTEwMTAwOTA1MTNaMCMGCSqGSIb3DQEJBDEWBBQ7IL1HGyxFi8AvLIunzbRzfQHS%21zAJBgcqhkjOOAQDBC4wLAIUcBNctV0PGncl1EEqJLeN1%21LRwKgCFBVzgggEgv0B4J%2f2NsyOlxl6FB2O; path=/; secure; HttpOnly;HttpOnly;Secure;SAP_SESSIONID_KHM_081=mrtPv04_zkTF04Kqwlh6AxHso93rPRHptjcAFj5x15Q%3d; path=/; secure; HttpOnly;HttpOnly;Secure
REQUEST_URL	POST https://my343094.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder/SalesOrderCollection HTTP/1.1

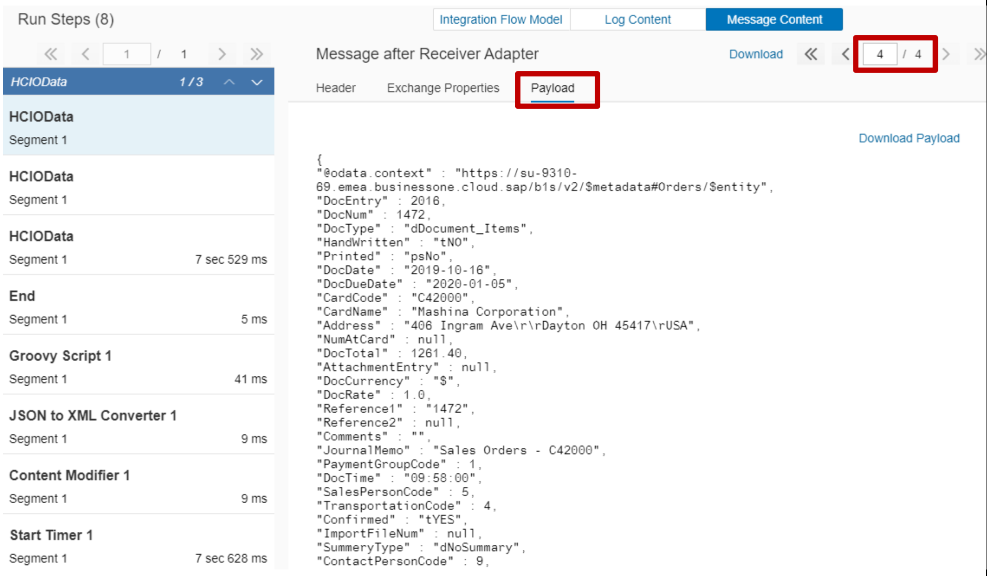
Explanation	Screenshot
For ByD: Go to the **Payload** tab. You can see the Payload that has been sent to ByD in order to create the SalesOrder.	
For B1: Select **page 1**. You can see it is a GET request done automatically by CPI OData adapter in order to retrieve metadata.	
For B1: Select **page 3**. It shows the POST request sent to B1.	

Encoding: UTF-8

```
<?xml version='1.0' encoding='utf-8'?><entry xmlns="http://www.w3.org/2005/Atom"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xml:base="https://my343094.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder/">
<id>https://my343094.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder/SalesOrderCollect
ion</id><title type="text">SalesOrderCollection</title><updated>2019-10-
10T09:05:13.678Z</updated><category term="cust.SalesOrder"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/><link
href="SalesOrderCollection('A')" rel="edit" title="SalesOrder"/><link
href="SalesOrderCollection('A')/Item"
rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Item" title="Item"
type="application/atom+xml;type=feed"><m:inline><feed
xml:base="https://my343094.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder/">
<id>https://my343094.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder/ItemCollection</i
d><title type="text">ItemCollection</title><updated>2019-10-10T09:05:13.678Z</updated>
<author><name/></author><link href="ItemCollection" rel="self" title="ItemCollection"/>
<entry>
<id>https://my343094.sapbydesign.com/sap/byd/odata/cust/v1/khsalesorder/ItemCollection</i
d><title type="text">ItemCollection</title><updated>2019-10-10T09:05:13.679Z</updated>
```

Name	Value
REQUEST_URL	GET https://su-9310-69.emea.businessone.cloud.sap/b1s/v2/$metadata HTTP/1.1

Name	Value
Accept	application/json;odata.metadata=minimal
Authorization	***
Content-Type	application/json
Cookie	CompanyDB=SHA256:0969b01cde2300b53316f66f06cea5d3786524175d93db4f66eb6180395fb081;HttpOnly;; B1SESSION=SHA256:9378c4c539f4f6c0da7f0bb61ca9a2eae316a14a1dc5fcb3180c805d6d1f9e75;HttpOnly;; ROUTEID=SHA256:08685aef2f19ded176aa9638cd9a598a712886d1b9723cf06685aab540e52cc6; path=/b1s
OData-MaxVersion	4.0
OData-Version	4.0
REQUEST_URL	POST https://su-9310-69.emea.businessone.cloud.sap/b1s/v2/Orders HTTP/1.1
SAP-PASSPORT	2A54482A0300E600004350490000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000000000 000000004D657373616765526563657074696F6E0000000000000000000000000000000 0000000000000000000004350495F6431303031303400000000000000000000000000 00000000000000383164616364646631303630343864303964386263636537663135313

Explanation	Screenshot
For B1: Select **page 4**. Check the **Payload** returned by B1 after the creation of the SalesOrder.	

Step	Segment	Time
HCIOData	Segment 1	
HCIOData	Segment 1	
HCIOData	Segment 1	7 sec 529 ms
End	Segment 1	5 ms
Groovy Script 1	Segment 1	41 ms
JSON to XML Converter 1	Segment 1	9 ms
Content Modifier 1	Segment 1	9 ms
Start Timer 1	Segment 1	7 sec 628 ms

```
{
"@odata.context" : "https://su-9310-
69.emea.businessone.cloud.sap/b1s/v2/$metadata#Orders/$entity",
"DocEntry" : 2016,
"DocNum" : 1472,
"DocType" : "dDocument_Items",
"HandWritten" : "tNO",
"Printed" : "psNo",
"DocDate" : "2019-10-16",
"DocDueDate" : "2020-01-05",
"CardCode" : "C42000",
"CardName" : "Mashina Corporation",
"Address" : "406 Ingram Ave\r\rDayton OH 45417\rUSA",
"NumAtCard" : null,
"DocTotal" : 1261.40,
"AttachmentEntry" : null,
"DocCurrency" : "$",
"DocRate" : 1.0,
"Reference1" : "1472",
"Reference2" : null,
"Comments" : "",
"JournalMemo" : "Sales Orders - C42000",
"PaymentGroupCode" : 1,
"DocTime" : "09:58:00",
"SalesPersonCode" : 5,
"TransportationCode" : 4,
"Confirmed" : "tYES",
"ImportFileNum" : null,
"SummeryType" : "dNoSummary",
"ContactPersonCode" : 9,
```

Congratulations! You have completed now the CPI exercises.

www.sap.com/contactsap

Run Simple