

© 2018 by SAP SE or an SAP affiliate company. All rights reserved. No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE. The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary. These materials are provided by SAP SE and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty. SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices.

CONFIGURING, DEPLOYING AND MONITORING WEBUI BASED SCENARIO USING SAP CLOUD PLATFORM INTEGRATION

Exercise05: Using OData Provisioning of SAP Cloud Platform Integration



The Best-Run Businesses Run SAP™

INTEGRATION SCENARIO

In this exercise, you will learn how to create and expose OData services from various data sources using SAP Cloud Platform Integration.

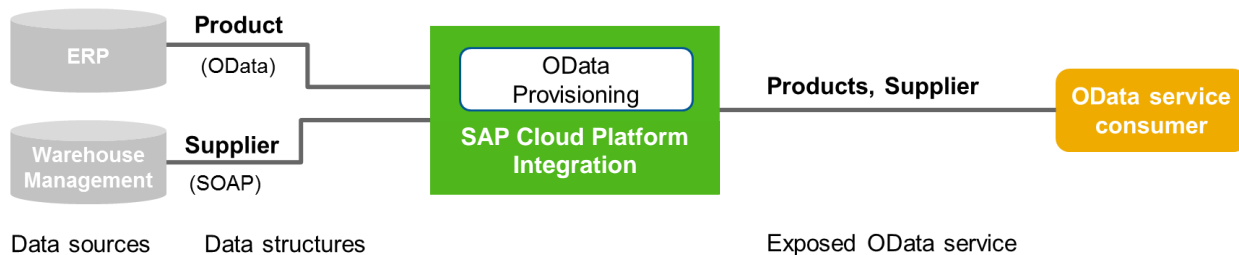
The capabilities of OData provisioning allow you to support new integration use cases with SAP Cloud Platform Integration where you expose an OData service which can be consumed by SAP Fiori applications, SAP Cloud Platform, mobile services or any other custom applications to fulfil user-centric scenarios.

As an example, you will create an OData services out of two different type of services as data sources – an OData service which delivers product data and a SOAP service which delivers supplier data.

This example depicts a typical business scenario where you need to check the current stock such as details of product/material in stock, units ordered from your legacy warehouse system which you would like to combine with supplier master data of those products which are available in your ERP system.

For the exercise, you will combine product data from a public and free accessible OData service while the supplier data from SOAP service which is mocked on the CPI tenant for you.

The tooling will guide you through the simple configuration steps and will auto-generate templates for developing the OData service in SAP Cloud Platform Integration for you.



During this exercise, you will conduct the following high-level steps:

1. Develop an OData service imported from an OData data source
2. Enrich your OData service by importing data from an SOAP data source
3. Expose a single OData service comprising out of two entity sets.

Along with this hands-on exercise you will learn the following:

How to ...

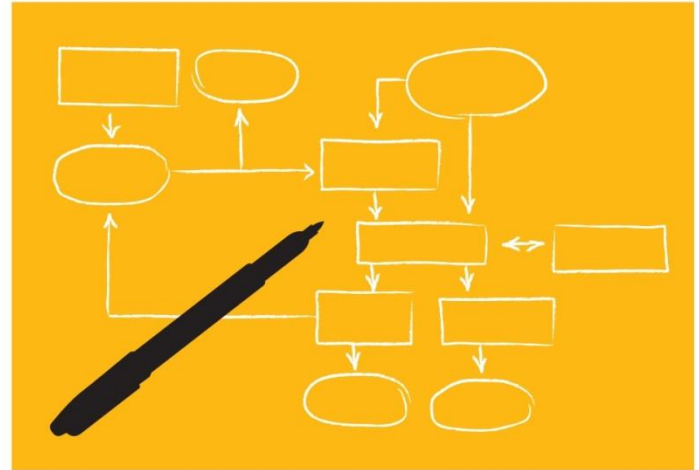
- import an OData model from an OData and SOAP data source using simple modelling steps.
- enhance/change an existing OData model using the OData Model Editor.
- use the Graphical OData Model Viewer to analyse an OData service in CPI
- deploy and monitor the OData service
- test the OData service using a Non-SAP test tool.

Welcome to the Integration Flow challenge!

In this exercise, your aim is to create an integration flow that solves a challenge (*described in the Integration scenario*).

And when you work your way through the exercise, our aim is that you learn:

1. How to access and work with the Cloud Platform Integration Web tooling
2. How to speed up your integration project by leveraging reference integration flows
3. How to customize a reference integration flow by configuring its connectivity and flow steps
4. Basic monitoring of an integration flow
5. Using Open-source tools to test your integration flow



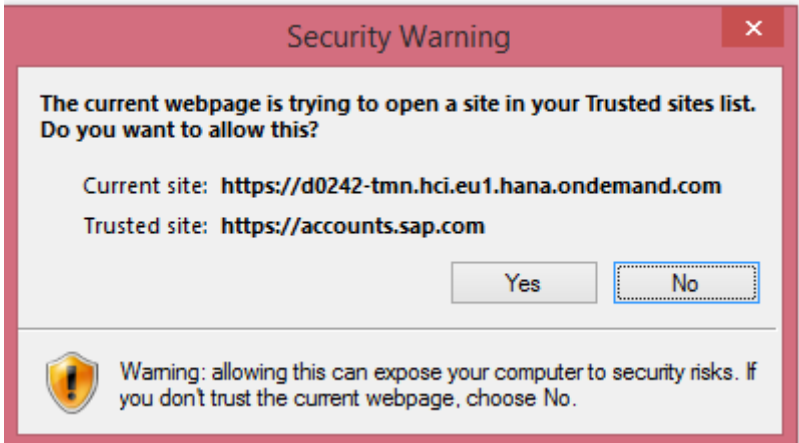
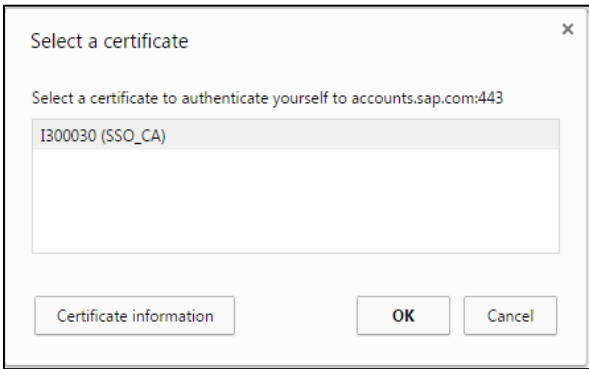
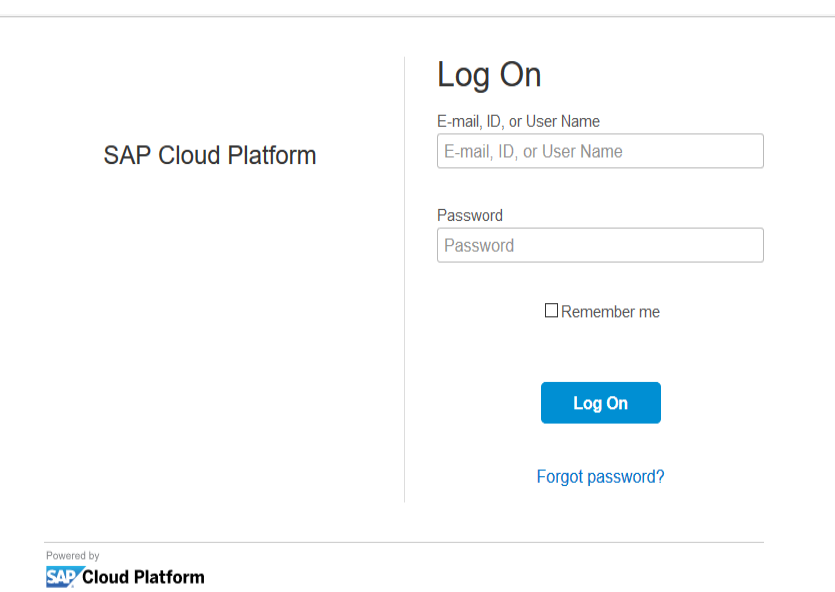
Exercise Files

Exercise Files are provided by the session moderator. Download and Extract them into one of your local folders for use later in your exercises and setup:

Note

1. In the exercise, we have used the notation XX to refer to the content created by you or your group uniquely. The session moderator shall assist you with assigning you/group the unique ID that you can use to replace XX.
2. In the exercise, we have used **d0242/d0243** as the example tenants. The final tenant details shall be provided by the session moderator.
3. Please note that colours and other visual appearance might differ slightly from the screenshot screens, as the CPI editor might have received feature upgrades since production of this content.

EXERCISE 05

Explanation	Screenshot
<p>Follow these steps for integration flow creation in Web UI.</p>	
<p>1) Access the Web application using Google Chrome (or IE10 Browser)</p> <ol style="list-style-type: none"> Open Google Chrome Copy the application URL or The URL that you can use will be assigned to you by the instructor. You will get a security warning if you want to open this site in your trusted sites list. <p>Accept it with 'Yes'.</p>	 <p>A 'Security Warning' dialog box with a red border and a close button (X) in the top right corner. The text inside reads: 'The current webpage is trying to open a site in your Trusted sites list. Do you want to allow this?'. Below this, it shows 'Current site: https://d0242-tmn.hci.eu1.hana.ondemand.com' and 'Trusted site: https://accounts.sap.com'. At the bottom, there are two buttons: 'Yes' and 'No'. Below the buttons, there is a warning icon (a shield with an exclamation mark) and the text: 'Warning: allowing this can expose your computer to security risks. If you don't trust the current webpage, choose No.'</p>
<p>2) If you get displayed your Certificate List, click on 'Cancel'</p>	 <p>A 'Select a certificate' dialog box with a close button (X) in the top right corner. The text inside reads: 'Select a certificate to authenticate yourself to accounts.sap.com:443'. Below this, there is a list box containing 'I300030 (SSO_CA)'. At the bottom, there are three buttons: 'Certificate information', 'OK', and 'Cancel'.</p>
<p>3) You will now see the log-on screen for the SAP ID Service/SAP Cloud Platform.</p> <p>Enter the following credentials:</p> <p><i>User:</i> <Provided by instructor> <i>Password:</i> <Provided by instructor></p>	 <p>The SAP Cloud Platform Log On screen. On the left, it says 'SAP Cloud Platform'. On the right, under the heading 'Log On', there are two input fields: 'E-mail, ID, or User Name' and 'Password'. Below these fields is a checkbox labeled 'Remember me'. At the bottom right, there is a blue 'Log On' button and a link that says 'Forgot password?'. At the very bottom, it says 'Powered by SAP Cloud Platform'.</p>

- 4) Now you will have access to the Web application. Familiarize yourself with the environment and choose the **Design** area. Here, you will create an Integration Package to store your iFlows for this Exercise

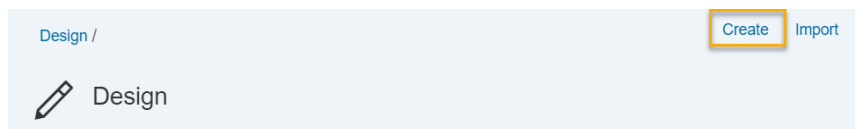
Click on **Create**

Enter Details


- a) **Name:**
OData_Services_Package_XX

NOTE: Replace XX with the group number provided by the instructor

- b) **Short Description:** EKT Cloud Platform, OData Service
- c) **Version:** 1.0.0
- d) **Vendor:** SAP



Design / New Package /

 OData_Services_Package_00

Header Overview Artifacts Documents Tags

*Name:

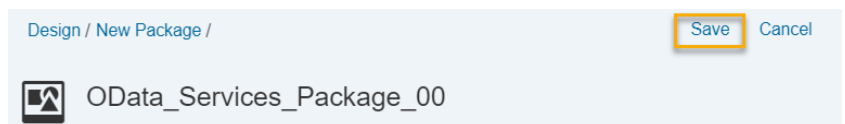
*Technical Name: ⓘ

*Short Description:

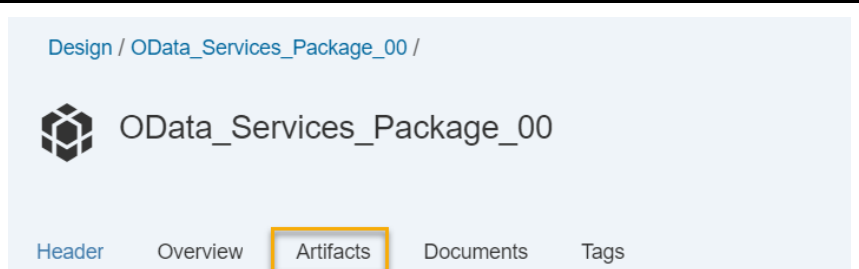
Version:

Vendor:

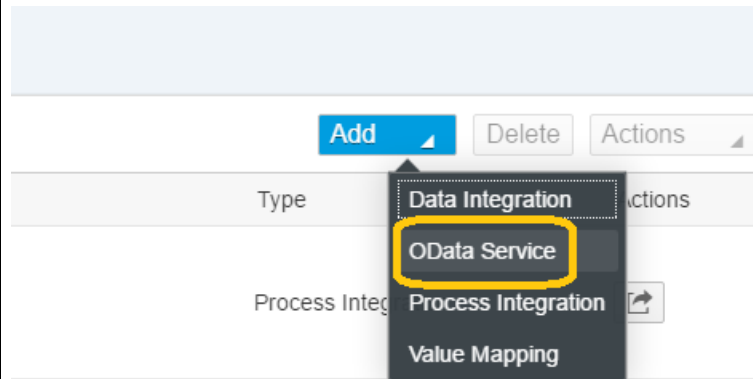
- 5) Click on Save



- 6) Navigate to **Artifacts** Tab



7) Click on **Add -> OData Service**




8) Enter as name for your OData service **Exercise05_ODP_XX** and click on **OK**.

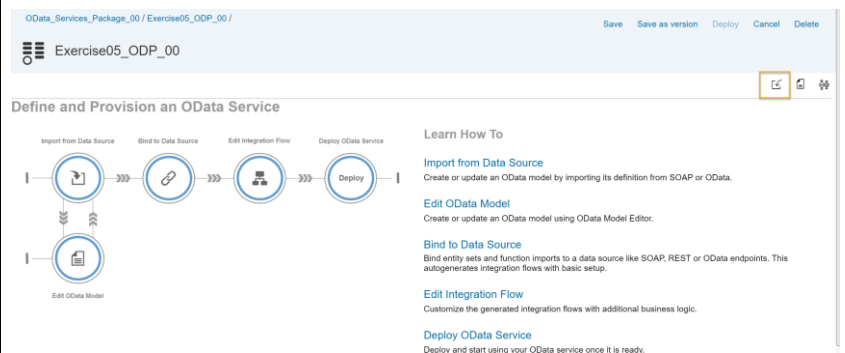
Hint: ODP stands for OData provisioning.

NOTE: Replace XX with the group number provided by the instructor.

A screenshot of the 'Add OData Service Artifact' dialog box. It has two tabs: 'Create' (selected) and 'Upload'. The 'Create' tab contains the following fields: '*Name:' with the value 'Exercise05_ODP_00', '*Namespace:' with the value 'SAP', '*ID:' with the value 'Exercise05_ODP_00_SAP_1', and a 'Description:' field with a placeholder '<Description>'. At the bottom, there is a '*OData Version:' dropdown menu set to 'OData V2'. 'OK' and 'Cancel' buttons are at the bottom right.

9) The overview page of OData provisioning launches.

In the top right corner, click on **Import Model Wizard** () for importing data structures from an OData or SOAP data source.



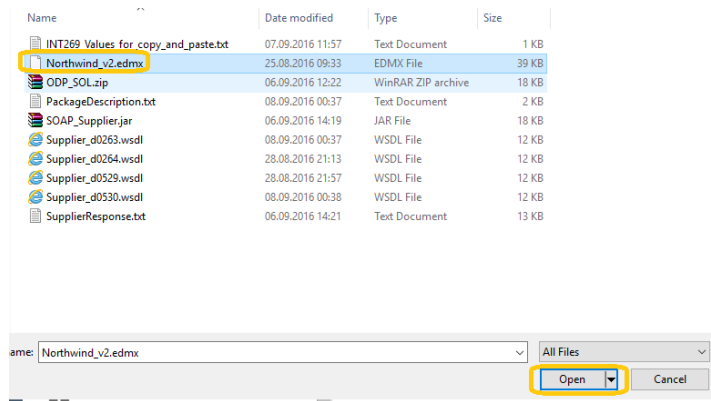
10) First you select a data source and import the respective definition file from your file system.

For the *Data Source Type* choose **ODATA** and click on **Browse**.

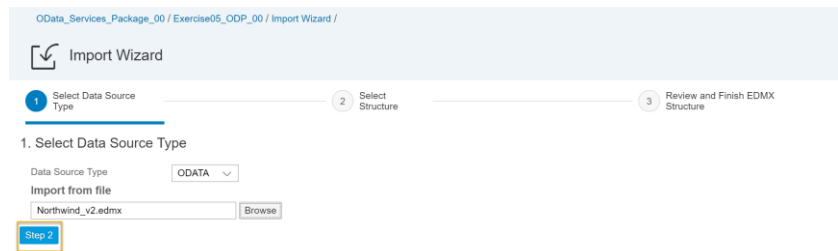
A screenshot of the 'Import Wizard' dialog box. The title bar shows 'OData_Services_Package_00 / Exercise05_ODP_00 / Import Wizard /'. The main content area has a progress bar with three steps: '1. Select Data Source Type', '2. Select Structure', and '3. Review and Finish EDMX Structure'. Under step 1, there is a 'Data Source Type' dropdown menu set to 'ODATA'. Below it, there is a 'Browse' button and a list of data source types: 'ODATA', 'SOAP', 'ODATA', and 'ODC'. The 'ODATA' option is highlighted with a yellow rectangle.

11) Now let's import an OData definition file (.edmx) of a public OData service which includes data structures for Products:

Go to exercises folder, select file **Northwind_v2.edmx** and click on **Open**.

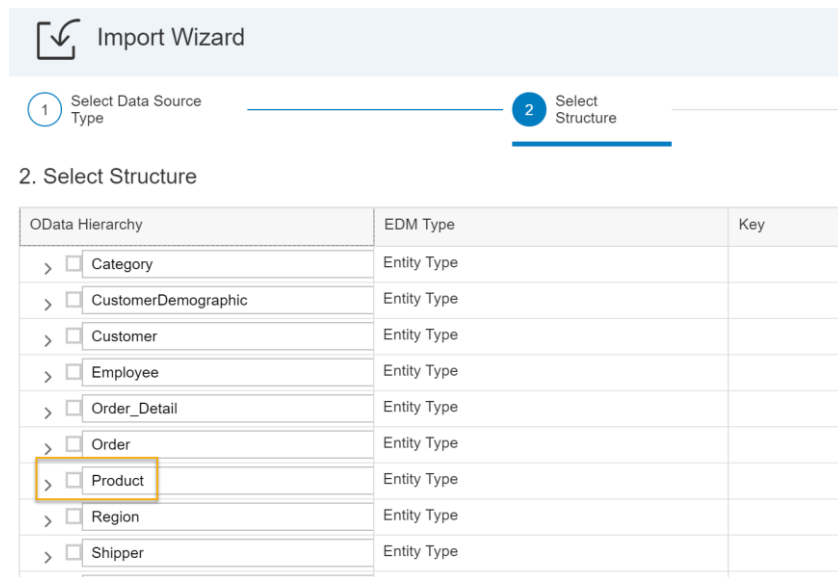


12) The button **Step 2** appears. Click on the same.



13) The structure of the OData service is shown. This OData structure comprises of several entities but we want to implement just one for now.

Therefore, click on ► next to the **Product** to expand this entity set.



14) Select the following properties of the Product Entity from the list:

- **ProductID**
- **ProductName**
- **SupplierID**
- **CategoryID**
- **QuantityPerUnit**
- **UnitPrice**
- **UnitsInStock**

Click on button **Step 3**.

2. Select OData Structure

OData Hierarchy	EDM Type
▼ <input checked="" type="checkbox"/> Product	Entity Type
<input checked="" type="checkbox"/> ProductID	Edm.Int32
<input checked="" type="checkbox"/> ProductName	Edm.String
<input checked="" type="checkbox"/> SupplierID	Edm.Int32
<input checked="" type="checkbox"/> CategoryID	Edm.Int32
<input checked="" type="checkbox"/> QuantityPerUnit	Edm.String
<input checked="" type="checkbox"/> UnitPrice	Edm.Decimal
<input checked="" type="checkbox"/> UnitsInStock	Edm.Int16
<input type="checkbox"/> UnitsOnOrder	Edm.Int16
<input type="checkbox"/> ReorderLevel	Edm.Int16

Step 3

15) In **Step 3**, you could change some definitions and names of the EDMX structure but you leave the configuration as it is.

Click on button **Finish**.

3. Review and Finish EDMX Structure


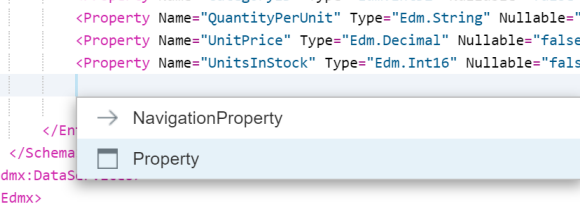
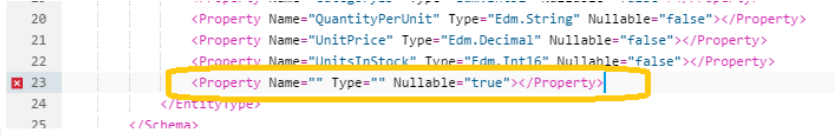
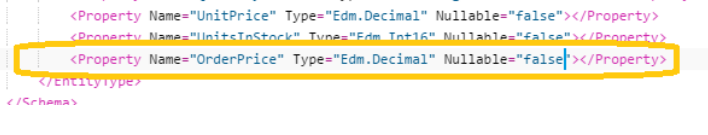
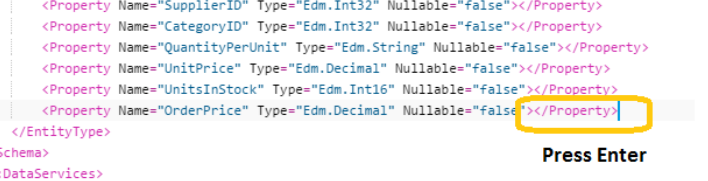
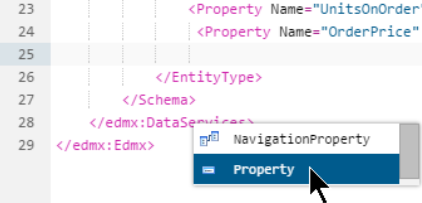
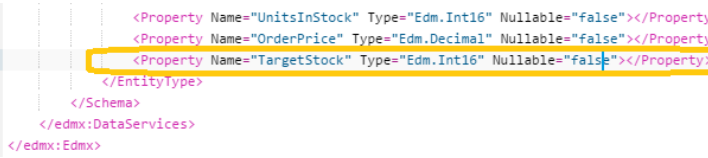
Restore Default Values			
EDM Name	EDM Type		Key
▼ <input checked="" type="checkbox"/> Product			
<input type="checkbox"/> ProductID	Int32	▼	<input checked="" type="checkbox"/>
<input type="checkbox"/> ProductName	String	▼	<input type="checkbox"/>
<input type="checkbox"/> SupplierID	Int32	▼	<input type="checkbox"/>
<input type="checkbox"/> CategoryID	Int32	▼	<input type="checkbox"/>
<input type="checkbox"/> QuantityPerUnit	String	▼	<input type="checkbox"/>
<input type="checkbox"/> UnitPrice	Decimal	▼	<input type="checkbox"/>
<input type="checkbox"/> UnitsInStock	Int16	▼	<input type="checkbox"/>

Finish

16) As a result, operations (query, create, read, update, delete) for the OData structure are generated.

Click on icon of the **OData Model Editor** (📄) in the top right corner. By using this modeler, you can modify the OData structure which was automatically created before.

OData_Services_Package_00 / Exercise05_ODP_00 /				Save	Save as version	Deploy	Cancel	Delete
Exercise05_ODP_00								
Name	Data Source	Details	Action					
▼ ProductSet	ODATA ▼							
• Query								
• Create								
• Read								
• Update								
• Delete								

<p>17) In this exercise, we will add two new properties to the OData structure. Place your cursor after the last property end tag (<code></Property></code>) and press ENTER. This will add an empty row.</p>	
<p>18) Press keys CTRL + SPACEBAR on your keyboard for launching the <i>Schema-Based Code Assist</i> of the <i>OData Model Editor</i>.</p> <p>Select Property from the context menu.</p>	
<p>19) The parameters for defining a property are inserted automatically.</p>	
<p>20) Enter the following values:</p> <ul style="list-style-type: none"> Name: OrderPrice Type: Edm.Decimal Nullable: false 	
<p>21) Let's add another new property to the EDMX file. Again, place the cursor after the last property end tag (<code></Property></code>) and press ENTER for adding a new row.</p>	
<p>22) Press keys CTRL + SPACEBAR on your keyboard for launching the <i>Schema-Based Code Assist</i> of the <i>OData Model Editor</i> once more.</p> <p>Select Property from the context menu.</p>	
<p>23) Enter the following values:</p> <ul style="list-style-type: none"> Name: TargetStock Type: Edm.Int16 Nullable: false 	

24) Click on OK in the right corner at the top of the OData Model Editor. With that you have finished the data structure definition of the new OData service.

```

5  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" m:DataServiceVersion="2.0"
6  <Schema
7    xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="S1"
8    <EntityContainer Name="EC1" m:IsDefaultEntityContainer="true"
9      <EntitySet Name="ProductSet" EntityType="S1.Product"/></EntitySet>
10    </EntityContainer>
11    <EntityType Name="Product">
12      <Documentation/
13      <Key>
14        <PropertyRef Name="ProductID"/></PropertyRef>
15      </Key>
16      <Property Name="ProductID" Type="Edm.Int32" Nullable="false"/></Property>
17      <Property Name="ProductName" Type="Edm.String" Nullable="false"/></Property>
18      <Property Name="SupplierID" Type="Edm.Int32" Nullable="false"/></Property>
19      <Property Name="CategoryID" Type="Edm.Int32" Nullable="false"/></Property>
20      <Property Name="QuantityPerUnit" Type="Edm.String" Nullable="false"/></Property>
21      <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="false"/></Property>
22      <Property Name="UnitsInStock" Type="Edm.Int16" Nullable="false"/></Property>
23      <Property Name="OrderPrice" Type="Edm.Decimal" Nullable="false"/></Property>
24      <Property Name="TargetStock" Type="Edm.Int16" Nullable="false"/></Property>
25    </EntityType>
26  </Schema>
27  </edmx:DataServices>
28  </edmx:Edmx>

```

25) Click on Navigation Path “Exercise05_ODP_00” to go back to main integration scenario screen

```

5  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" m:DataServiceVersion="2.0"
6  <Schema
7    xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="S1"
8    <EntityContainer Name="EC1" m:IsDefaultEntityContainer="true"
9      <EntitySet Name="ProductSet" EntityType="S1.Product"/></EntitySet>
10    </EntityContainer>
11    <EntityType Name="Product">
12      <Documentation/
13      <Key>
14        <PropertyRef Name="ProductID"/></PropertyRef>
15      </Key>
16      <Property Name="ProductID" Type="Edm.Int32" Nullable="false"/></Property>
17      <Property Name="ProductName" Type="Edm.String" Nullable="false"/></Property>
18      <Property Name="SupplierID" Type="Edm.Int32" Nullable="false"/></Property>
19      <Property Name="CategoryID" Type="Edm.Int32" Nullable="false"/></Property>
20      <Property Name="QuantityPerUnit" Type="Edm.String" Nullable="false"/></Property>
21      <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="false"/></Property>
22      <Property Name="UnitsInStock" Type="Edm.Int16" Nullable="false"/></Property>
23      <Property Name="OrderPrice" Type="Edm.Decimal" Nullable="false"/></Property>
24      <Property Name="TargetStock" Type="Edm.Int16" Nullable="false"/></Property>
25    </EntityType>
26  </Schema>
27  </edmx:DataServices>
28  </edmx:Edmx>

```

26) Next you need to bind the operation to a data source. Let's do that for the **Query** operation to get a list of Products. Click on the **Bind** icon (🔗) in the row of the **Query** operation.

Name	Data Source	Details	Action
ProductSet	ODATA		
Query			
Create			
Read			
Update			
Delete			

27) You now configure the data source of the OData service. Some setting is already preconfigured. For the missing ones add the following:

- **Entity Sets:** Products
- **End Point:** <http://services.odata.org/V2/Northwind/Northwind.svc>

Finally click on **OK** button.

Configure ODATA Data Source

Entity Set: ProductSet

CRUDQ Operation: Query

Upload OData Model: Northwind_v2.edmx Browse...

Existing Files: Northwind_v2


metadata


*Entity Sets: Products

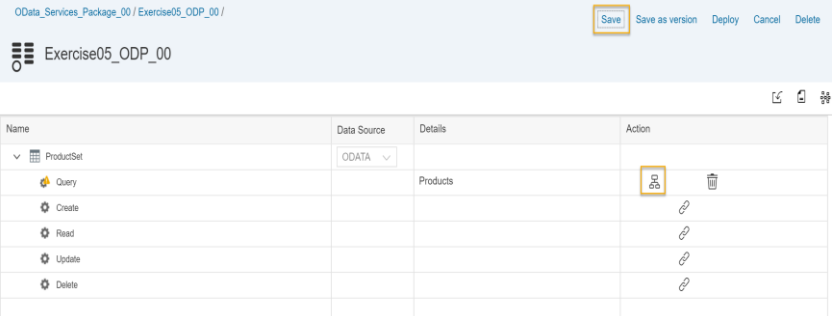
*End Point: http://services.odata.org/V2/Northwind/Northwind.svc







OK Cancel

28) Click on **Save**.

You can see that there is an integration flow () added to the **Query** operation. This is an auto generated integration flow which you will configure in the next steps for connecting your new OData service with the data source and implement needed mappings.

Click on the integration flow icon ().

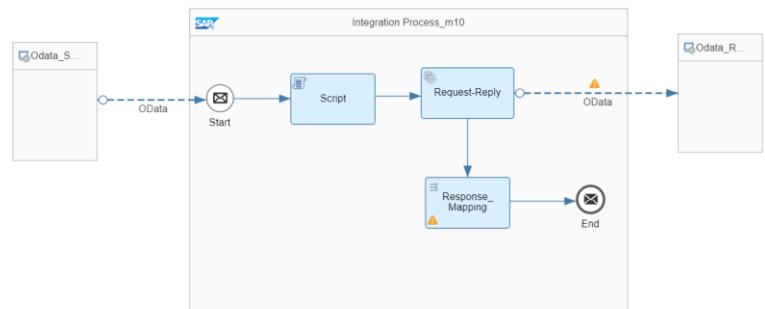


Name	Data Source	Details	Action
ProductSet	ODATA	Products	
Query			
Create			
Read			
Update			
Delete			

29) In case of OData sources, the integration flow is already preconfigured to a large extend.

You only need to implement the following:

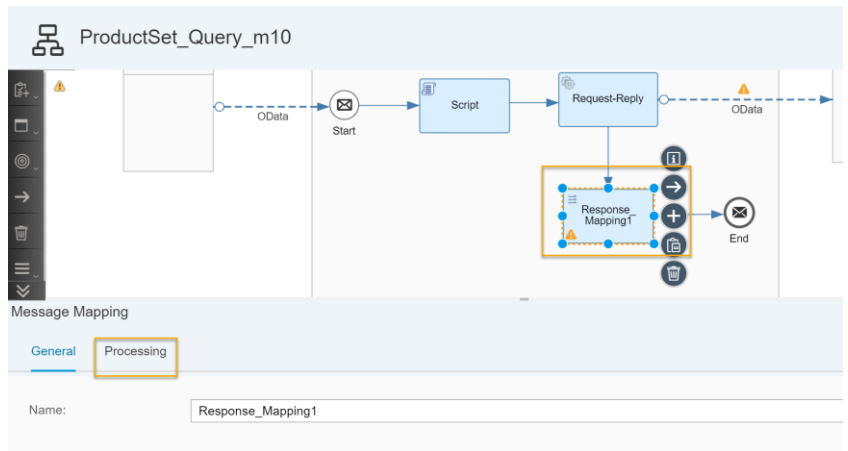
- Response Mapping
- Configure the OData channel to connect with the OData source



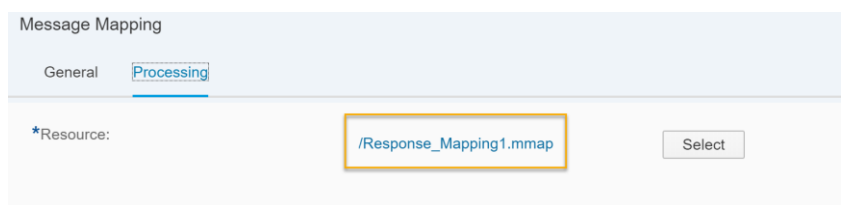
30) Let's implement the response mapping to map the message (here: Products) from the OData source to the new OData service (here: ProductSet).

Select **Response_Mapping** step in the design canvas.

Navigate to **Processing** tab.



31) In **Processing** tab, click on the mapping's name which is **Response_Mapping1.mmap**. This will launch the mapping editor.



32) On both sides, expand the nodes by clicking on the expand icons (▶) next to **Products** on the left and **ProductSet** on the right-hand side which would display the attributes.

Response_Mapping1

Products		ProductSet	
Structure	Occur...	Structure	Occur...
Products	1..1	ProductSet	1..1
Product	0..*	Product	0..*
ProductID	1..1	ProductID	1..1
ProductName	1..1	ProductName	1..1
SupplierID	0..1	SupplierID	1..1
CategoryID	0..1	CategoryID	1..1
QuantityPerUnit	0..1	QuantityPerUnit	1..1

33) Implement the mapping program:

- Click **Products** on the left and drag & drop it to the corresponding field of **ProductSet** on the right.
- Repeat the mapping of following elements:
Product
ProductID,
ProductName,
SupplierID,
CategoryID,
QuantityPerUnit,
UnitPrice,
UnitsInStock

Products		ProductSet	
Structure	Occurrence	Structure	Occurrence
Products	1..1	ProductSet	1..1
Product	0..*	Product	0..*
ProductID	1..1	ProductID	1..1
ProductName	1..1	ProductName	1..1
SupplierID	0..1	SupplierID	1..1
CategoryID	0..1	CategoryID	1..1
QuantityPerUnit	0..1	QuantityPerUnit	1..1
UnitPrice	0..1	UnitPrice	1..1
UnitsInStock	0..1	UnitsInStock	1..1
UnitsOnOrder	0..1	OrderPrice	1..1
ReorderLevel	0..1	TargetStock	1..1
Discontinued	1..1		

34) For the two new elements that we added to the EDMX file using the OData Model Editor, we need to implement a mapping using a predefined mapping function.

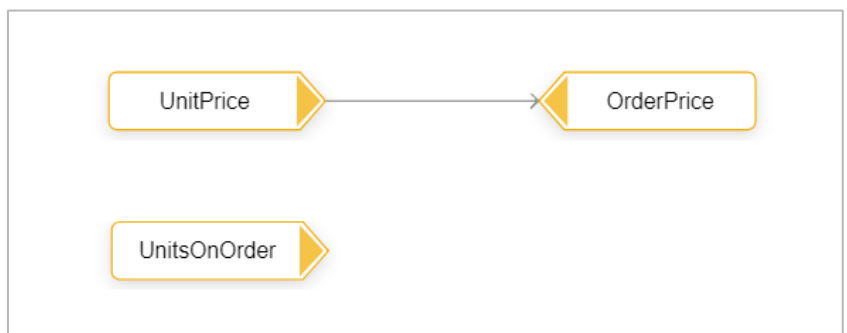
Let's start with element **OrderPrice** of the target message, which we can define by multiplying the value of the source elements **UnitPrice** and **UnitsOnOrder**.

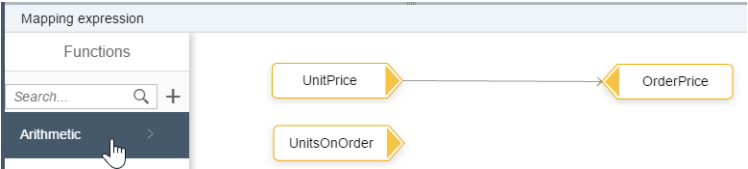
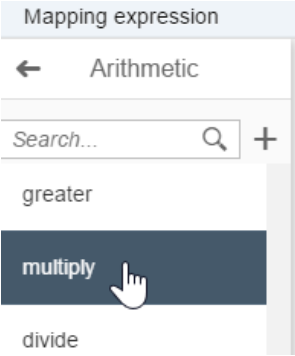
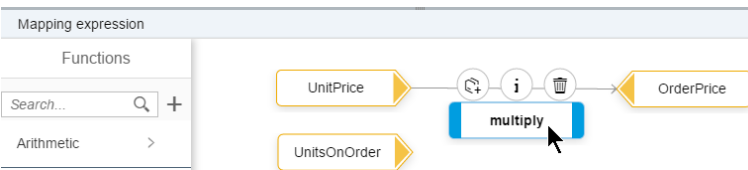
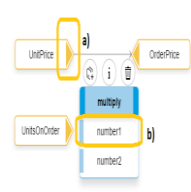
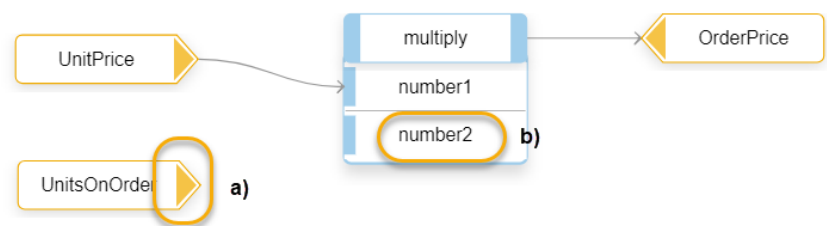
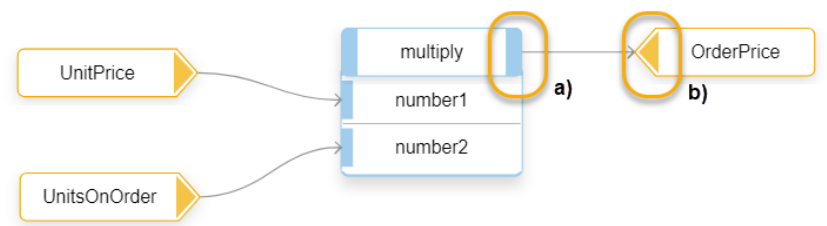
First, drag & drop **UnitPrice** of the source message to **OrderPrice**.

Second, drag & drop **UnitsOnOrder** of the source structure to **OrderPrice** of the target structure

In the Mapping Expression Editor **UnitPrice** and **UnitsOnOrder** are shown on the left and **OrderPrice** on the right.

Products		ProductSet	
Structure	Occurrence	Structure	Occurrence
QuantityPerUnit	0..1	CategoryID	
UnitPrice	0..1	QuantityPerUnit	
UnitsInStock	0..1	UnitPrice	
UnitsOnOrder	0..1	UnitsInStock	
ReorderLevel	0..1	OrderPrice	
Discontinued	1..1	TargetStock	



<p>Hint: Move OrderPrice a bit to the right for adding more space to add the mapping function.</p>	
<p>35) From the list of <i>Functions</i> click on Arithmetic.</p>	
<p>36) The list of arithmetical mapping functions appears.</p> <p>Click on multiply.</p>	
<p>37) The multiply function is added to the canvas of the <i>Mapping Expression Editor</i>. Select the same and move it in the middle between all three elements.</p>	
<p>38) Click on the yellow triangle of element UnitPrice (a) and click on multiply function for connecting them. The context menu of this function will appear: Choose number1 (b).</p>	
<p>39) Next, click on the yellow triangle of element UnitsOnOrder (a) and click on multiply function for connecting them. The context menu of this function will appear: Choose number2 (b).</p>	
<p>40) Finally select multiply function (a) and click on the yellow triangle in front of OrderPrice (b).</p> <p>With that the mapping for calculating OrderPrice out of the source message is implemented.</p>	

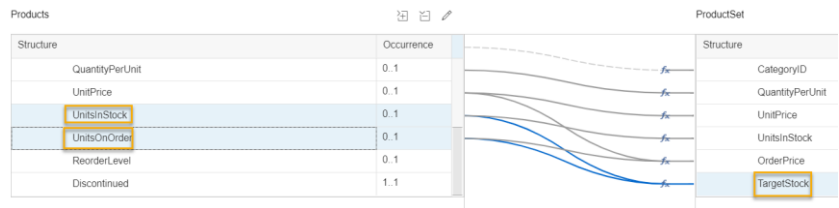
41) Now let's calculate the value for **TargetStock** using **UnitsInStock** and **UnitsOnOrder** elements of the source structure.

First, drag & drop **UnitsInStock** of the source message to **TargetStock**.

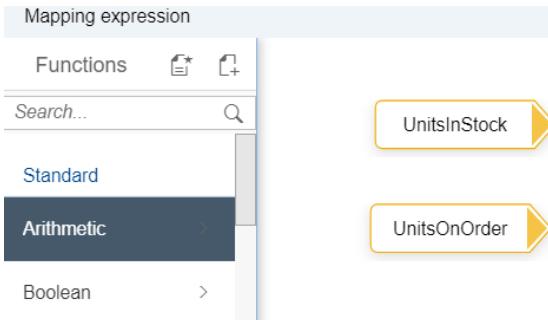
Second, drag & drop **UnitsOnOrder** of the source message to **TargetStock**.

In the Mapping Expression Editor **UnitsInStock** and **UnitsOnOrder** are shown on the left and **TargetStock** on the right.

Again, move **TargetStock** a bit to the right for adding more space to add the mapping function.

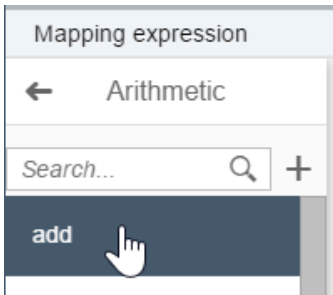


42) From the list of *Functions* click on **Arithmetic**.



43) The list of arithmetical mapping functions appears.

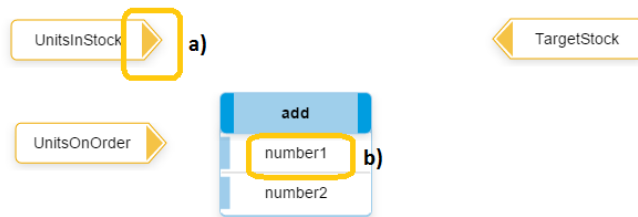
Click on **add**.



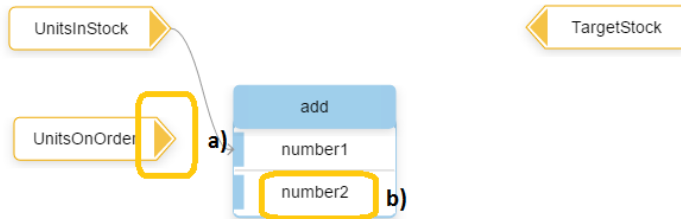
44) The **add** function is added to the canvas of the *Mapping Expression Editor*. Select the same and move it in the middle between all three elements.



- 45) Click on the yellow triangle of element **UnitsInStock** (a) and click on **add** function for connecting them. The context menu of this function will appear: Choose **number1** (b).



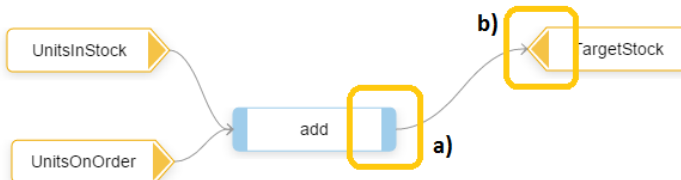
- 46) Next, click on the yellow triangle of element **UnitOnOrder** (a) and click on **add** function for connecting them. The context menu of this function will appear: Choose **number2** (b).



- 47) Finally select **add** function (a) and click on the yellow triangle in front of **TargetStock** (b).

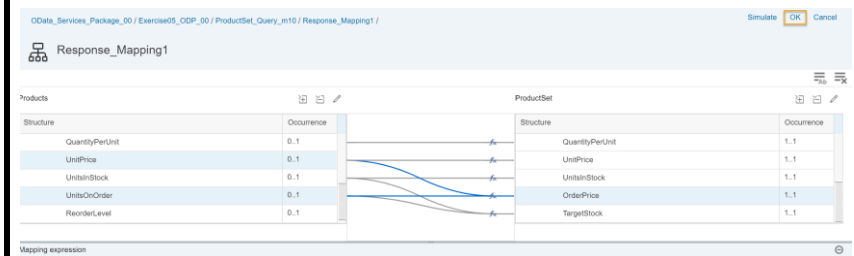
With that the mapping for calculating **TargetStock** out of the source message is implemented.

Now all elements of the target message (new OData service) are mapped.



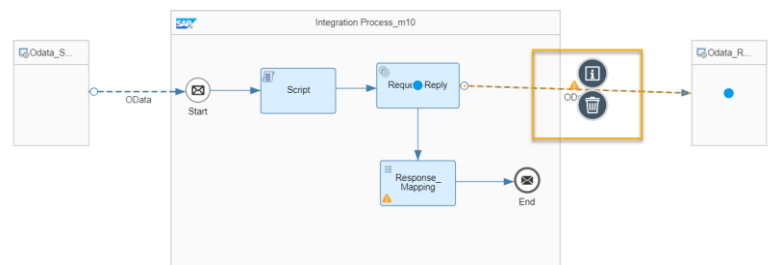
- 48) Click on **Ok** in the top right corner for saving your mapping.

You will be directed to the preconfigured integration flow.



- 49) Next, you configure the OData channel for pulling the needed order data out of the existing OData source.

Move the mouse pointer over the connection (channel) between **Request-Reply** step and **OData_Receiver** to select it (it should get highlighted in orange). Channel configuration should be visible in the bottom.



50) Switch to **Adapter-Specific** tab.

OData

General **Adapter Specific**

CONNECTION DETAILS

*Address: http://services.odata.org/V2/Northwind/Northwind.svc

*Proxy Type: Internet

*Authentication: None

PROCESSING DETAILS

*Operation Details:

Operation: Query(GET)
ResourcePath : \${header.odataEntity}?\${header.odataURI}
Path to edmx : /edmx/metadata_odata.edmx

51) Modify the resource path of the OData channel for defining which data to query from the OData source:

Specify the value of **Resource path** as follows:

Products?\$select=ProductID,ProductName,SupplierID,CategoryID,QuantityPerUnit,UnitPrice,UnitsInStock,UnitsOnOrder

Leave the other values as it is.

OData

General **Adapter Specific**

CONNECTION DETAILS

*Address: http://services.odata.org/V2/Northwind/Northwind.svc

*Proxy Type: Internet

*Authentication: None

PROCESSING DETAILS

*Operation Details:

Operation: Query(GET)
ResourcePath : Products?\$select=ProductID,ProductName,SupplierID,CategoryID,QuantityPerUnit,UnitPrice,UnitsInStock,UnitsOnOrder
Path to edmx : /edmx/metadata_odata.edmx

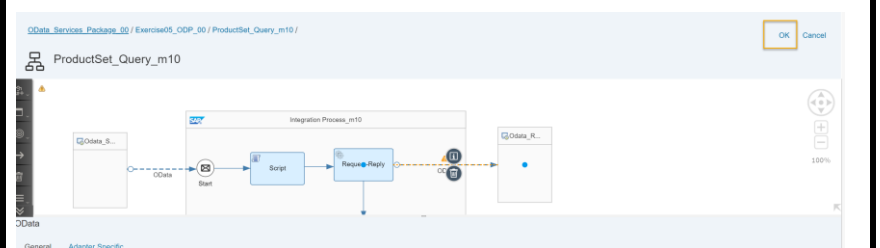
Custom Query Options:

Content Type: Atom

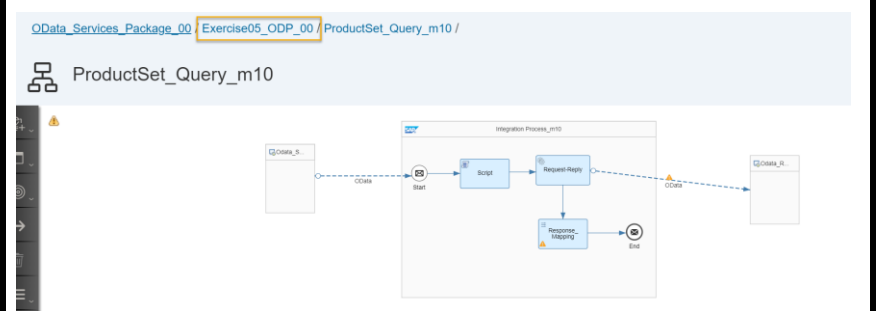
Content Type Encoding: None

Page Size:

52) Click on **OK** in the right corner at the top to save the OData adapter configuration



53) Click on Navigation Path **“Exercise05_ODP_00”** to go back to main integration scenario screen




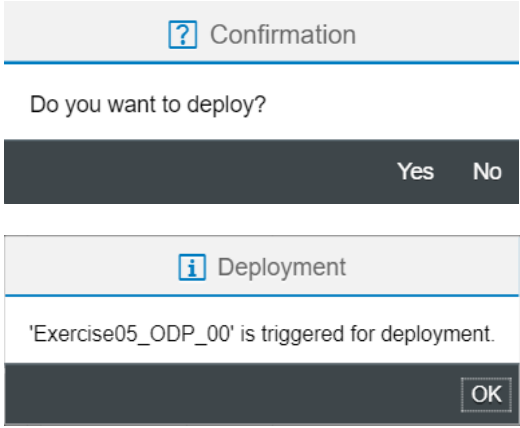

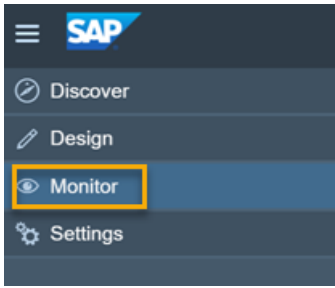
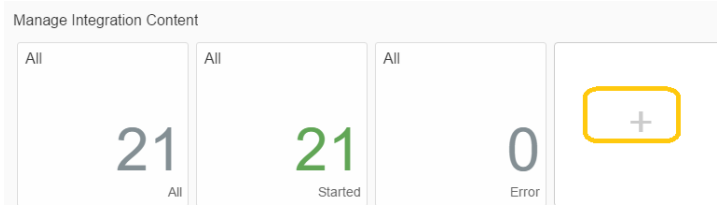
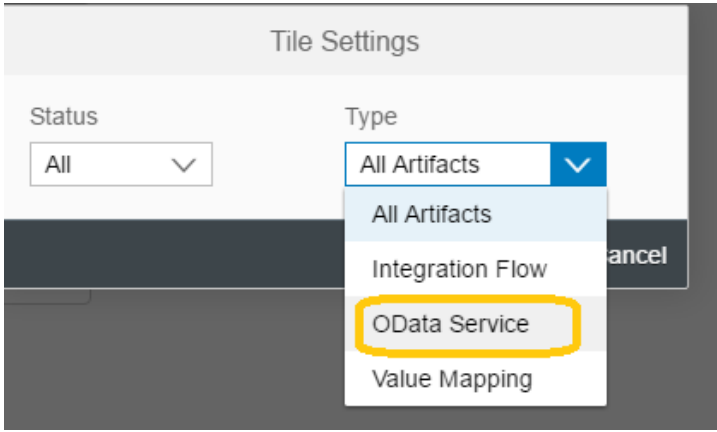
54) Click on **Save** to save the changes.

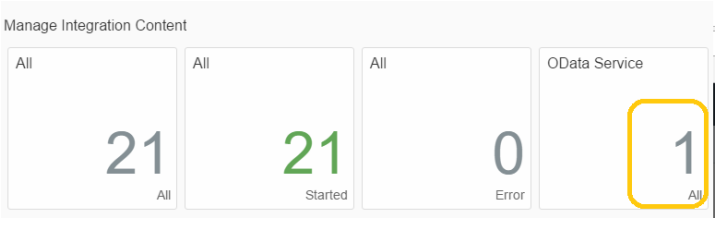
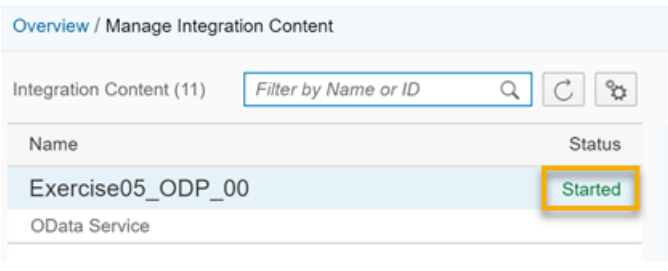

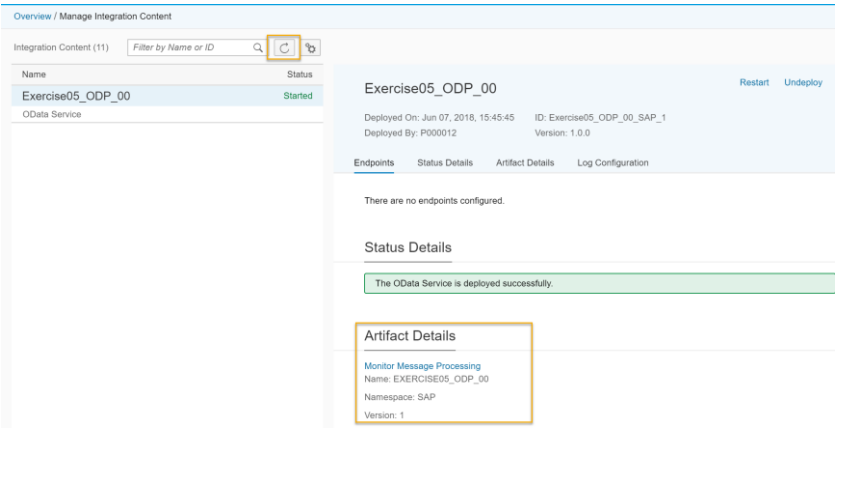

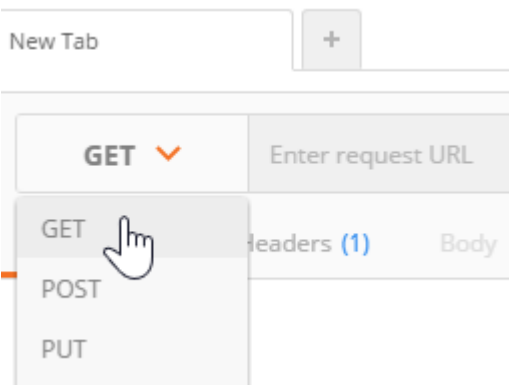
OData_Services_Package_00 / Exercise05_ODP_00 /

Exercise05_ODP_00

Save Save as version Deploy Cancel Delete

Name	Data Source	Details	Action
ProductSet	ODATA		
Query			
Create			
Read			
Update			
Delete			

Deploy Integration Project on tenant:	Follow this step to deploy Integration Project on tenant.
55) Press Deploy in the OData Service Task Bar.	
56) You will receive a confirmation. Click on Yes . Once the deployment is completed successfully you will receive a 2 nd notification.	
57) Verify if the deployment is successful: Go to the Monitoring of CPI to check the status of the deployed OData service: Click on  to navigate to the Monitor section.	
58) Under the Manage Integration Content Section, if there are only three tiles shown, click on the empty one at the end of the row to add the OData service view.	
59) The <i>Tile Settings</i> configuration will show up. Expand the <i>Type</i> list and choose OData Service from the dropdown menu. Click on OK . With that you created an extra tile for monitoring OData services.	

<p>60) Next, click on the tile OData Service (All).</p>	
<p>61) Your OData service should be listed there.</p>	
<p>62) Make sure that the status of the service is Started and that Artifact Details such as <i>Name</i>, <i>Namespace</i> and <i>Version</i> are shown.</p> <p>If the status of your OData Services is other than Started, then wait a while and refresh the monitoring data by clicking on the <i>refresh</i> () icon at the top to check if the status changed in the meantime.</p>	
<p>Execute end to end scenario:</p>	<p>Follow steps to execute end to end scenario.</p>
<p>63) Finally test the service using a test tool for OData services. For this hands-on we will use Postman. This has been installed and configured as a pre-requisite of this training.</p>	
<p>64) First let's make sure that the OData service was created successfully.</p> <p>Select GET as operation option in Postman</p>	

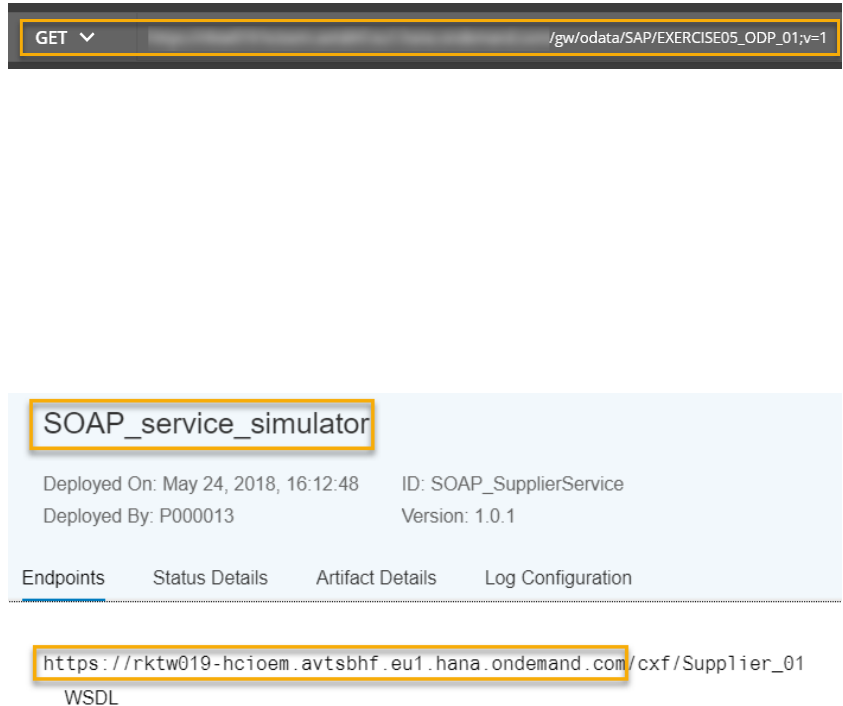
65) Enter the URL pointing to your OData Service. Replace the values highlighted in **RED**

https://<your runtime node hostname>/gw/odata/SAP/EXERCISE05_ODP_XX;v=1

Note: Please replace XX with your Group_No

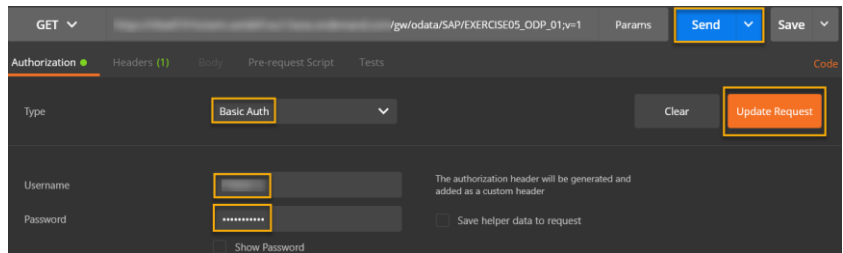
How to get runtime node hostname?

1. Go to monitoring section
2. Select Manage Integration Content(Started)
3. Search and select **SOAP_service_simulator** integration content that is already deployed.
4. Copy hostname from the **Endpoints** section which is highlighted in yellow.

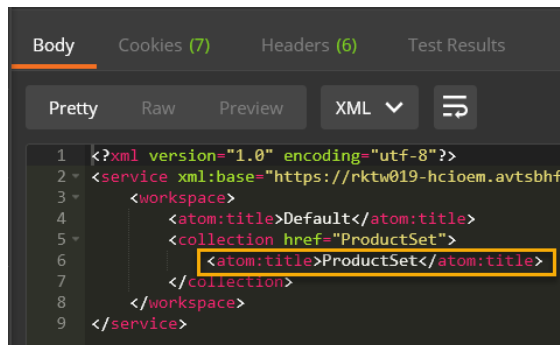


66) For Authorization choose Type as **Basic Auth** from the dropdown list.

Enter your user Id and password of CPI tenant. Don't forget to click on **Update Request**. And finally click on **Send** for sending out the request to your new OData service on CPI



67) If you get a response which shows that a collection **ProductSet** is available, then the OData service is registered on the CPI tenant.



68) Next, call GET operation of this OData service for retrieving the list of orders out of the OData source.

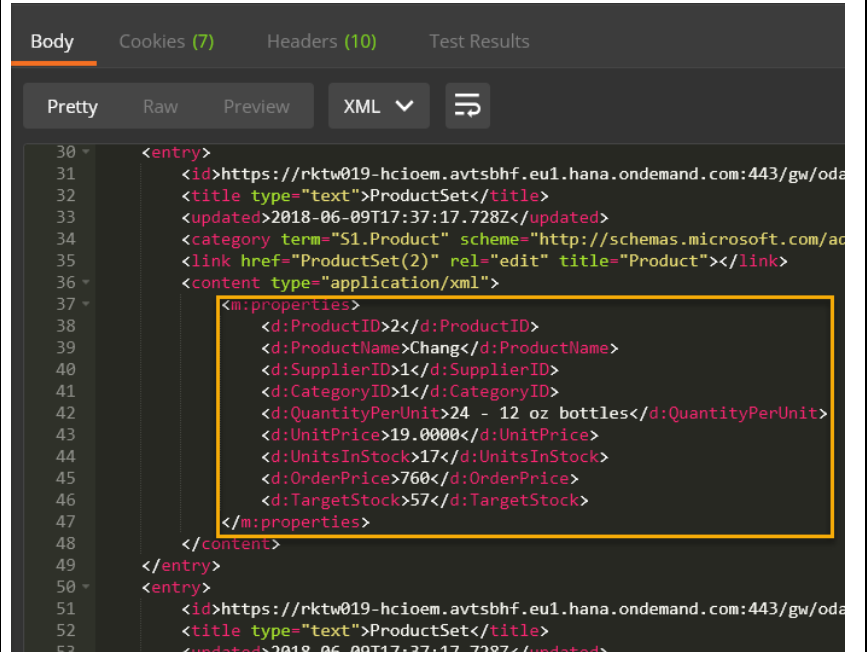
In Postman

- append **/ProductSet** to the previous URL
- click on **Send**



As response, you can see several Products listed which include the added properties for **OrderPrice** and **TargetStock** as well.

Congratulations, you have completed the first part of exercise successfully!

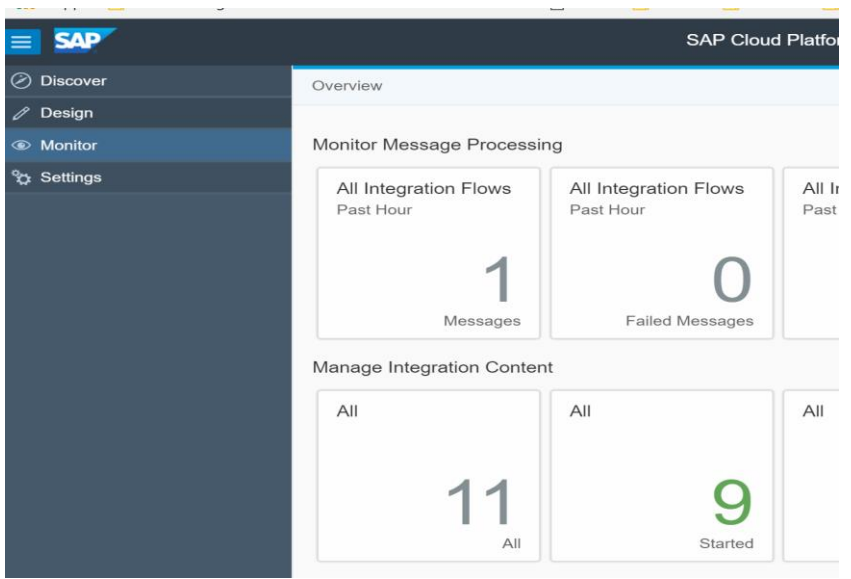
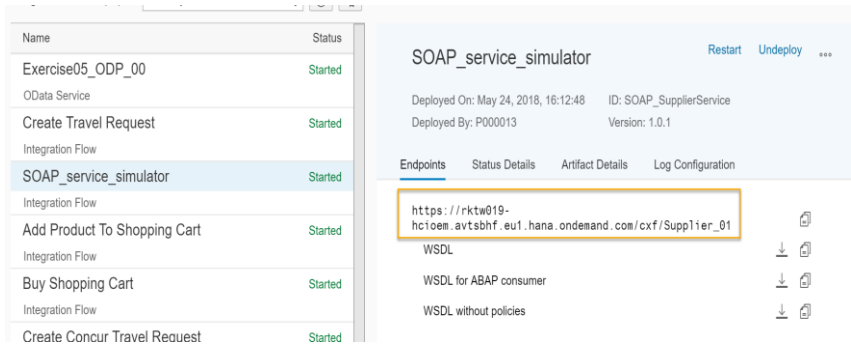


```
Body    Cookies (7)    Headers (10)    Test Results
Pretty  Raw    Preview    XML  ↕
30 <entry>
31   <id>https://rktw019-hcioem.avtsbhf.eu1.hana.ondemand.com:443/gw/oda
32   <title type="text">ProductSet</title>
33   <updated>2018-06-09T17:37:17.728Z</updated>
34   <category term="S1.Product" scheme="http://schemas.microsoft.com/ad
35   <link href="ProductSet(2)" rel="edit" title="Product"></link>
36   <content type="application/xml">
37     <m:properties>
38       <d:ProductID>2</d:ProductID>
39       <d:ProductName>Chang</d:ProductName>
40       <d:SupplierID>1</d:SupplierID>
41       <d:CategoryID>1</d:CategoryID>
42       <d:QuantityPerUnit>24 - 12 oz bottles</d:QuantityPerUnit>
43       <d:UnitPrice>19.0000</d:UnitPrice>
44       <d:UnitsInStock>17</d:UnitsInStock>
45       <d:OrderPrice>760</d:OrderPrice>
46       <d:TargetStock>57</d:TargetStock>
47     </m:properties>
48   </content>
49 </entry>
50 <entry>
51   <id>https://rktw019-hcioem.avtsbhf.eu1.hana.ondemand.com:443/gw/oda
52   <title type="text">ProductSet</title>
53   <updated>2018-06-09T17:37:17.728Z</updated>
```

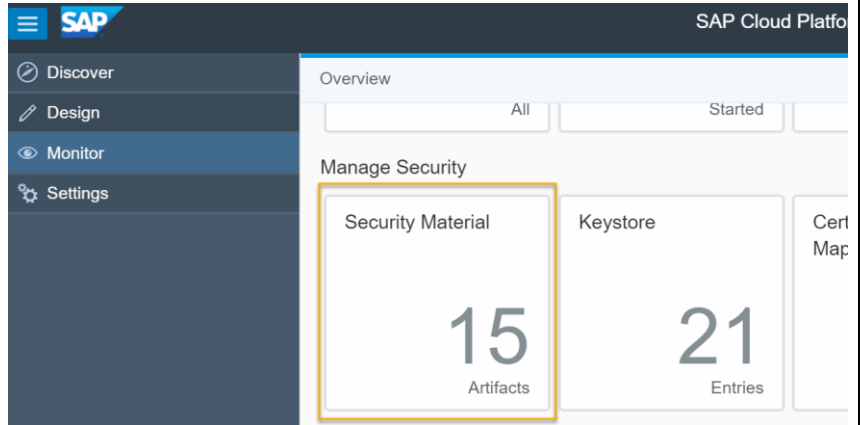
Enrich your OData service by importing data from a SOAP data source

In the second exercise, you are going to enhance your OData service and add structures from another data source. More precisely from a SOAP service for querying supplier master data (such as address data, contact information) from an ERP system (simulator SOAP service deployed on tenant) which are related to the products on stock of the legacy warehouse system. This will result in a single OData service retrieving data structures out of two different data sources. The implementation sequence is the same as of the previous exercise. But the configuration steps vary.

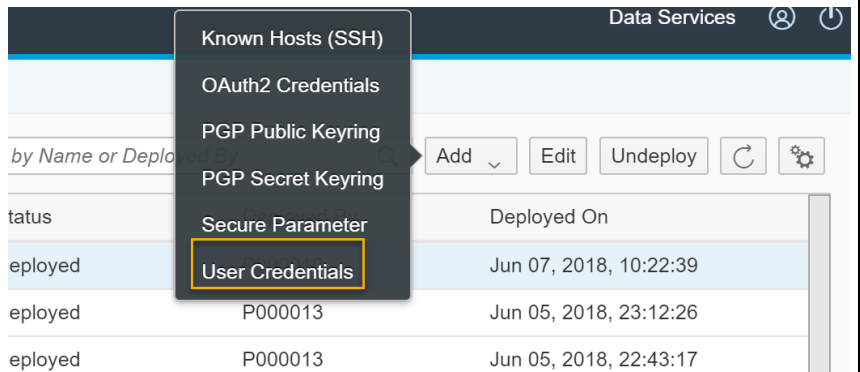
There is an integration flow pre-deployed on your tenant which acts as a SOAP data source in this exercise.

Explanation	Screenshot
Follow these steps for integration flow creation in Web UI.	
<p>1) Go back to Web UI in Google Chrome browser and click on the Monitor icon</p>	
<p>2) Search for integration scenario "SOAP_service_simulator". It can be found in the Started section of Manage Integration Content.</p> <p>Copy and save the SOAP service endpoint. It will be used in later steps in SOAP adapter while calling this SOAP service.</p> <p>Note: This SOAP service is simulator of ERP for providing the supplier data and is already deployed on the tenant</p>	

- 3) Go back to **Overview** screen, scroll down and click on **Security Material** in the Manage Security section.



- 4) Click on **Add→User Credentials**. This user credentials will be used in next steps to call the SOAP simulator service via basic authentication.



- 5) Enter the following values:

- Name:**
CPI_SERVICE_USER_XX
- User:** Your user id that you used to logon to tenant
- Password:** Your password

NOTE: Please replace XX with your Group_No

Click on Deploy and verify that user credentials are created.

Add User Credentials

*Name:

Description:

*User:

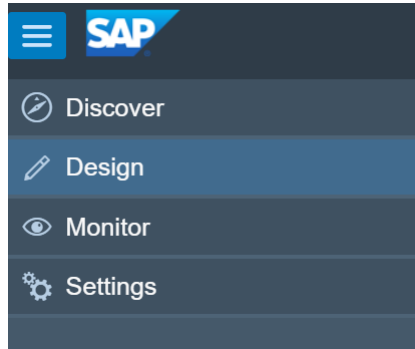
Password:

Repeat Password:

☐ SuccessFactors

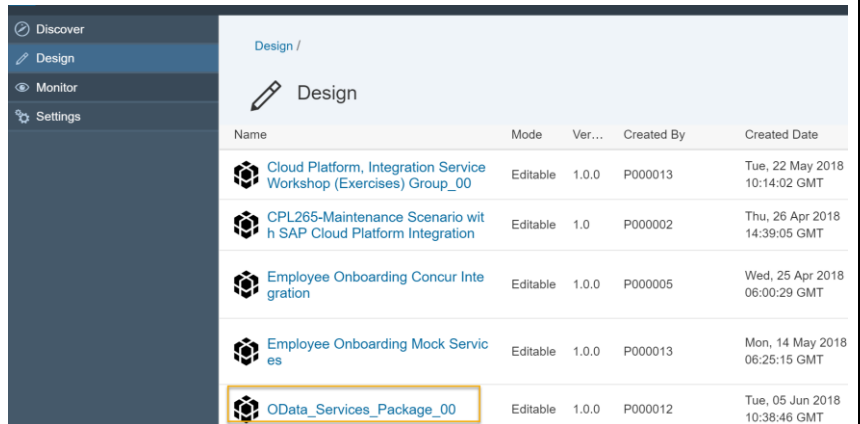
Name	Type	Status
CPI_SERVICE_USER_00	Credentials	Deployed

- 6) Switch to the **Design** section by clicking on the **Design** icon



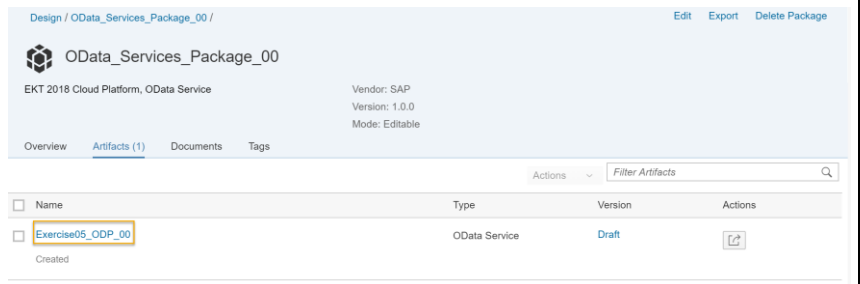
- 7) Search for your integration package and click on the **OData_Services_Package_XX** package name.

NOTE: Replace XX with the group number provided by the instructor

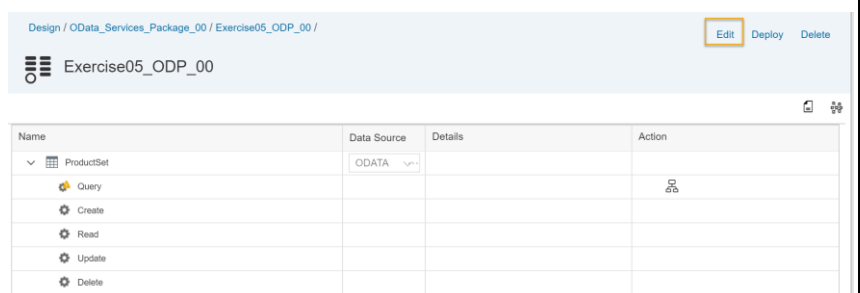


- 8) In the list of **Artifacts** select your OData service **Exercise05_ODP_XX**

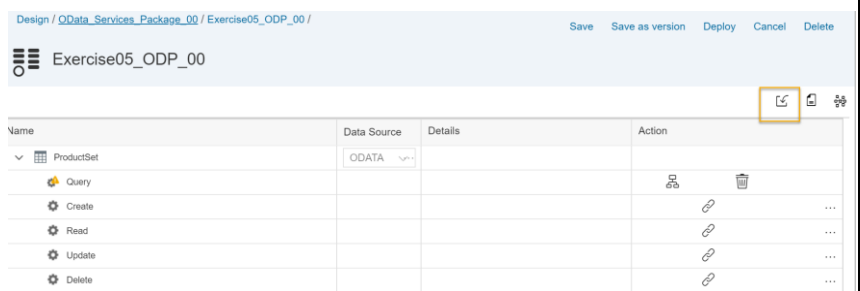
NOTE: Replace XX with the group number provided by the instructor



- 9) The **ProductSet** structure from previous exercise will show up on the screen. Click on **Edit** in the right top corner



- 10) Choose **Import Model Wizard**



11) This time you select **SOAP** as Data Source from the dropdown list and click on **Browse**.

Import Wizard

1 Select Data Source Type ————— 2 Select Structure

1. Select Data Source Type

Data Source Type SOAP ▾

Import from file

Browse

12) Go to exercises folder and select “**Supplier.wsdl**” file and click on button **Step 2**.

Import Wizard

1 Select Data Source Type —————

1. Select Data Source Type

Data Source Type SOAP ▾

Import from file

Supplier.wsdl Browse

Step 2

13) The structure of the SOAP service is shown. This SOAP service comprises of several operations but we want to implement just one for now.

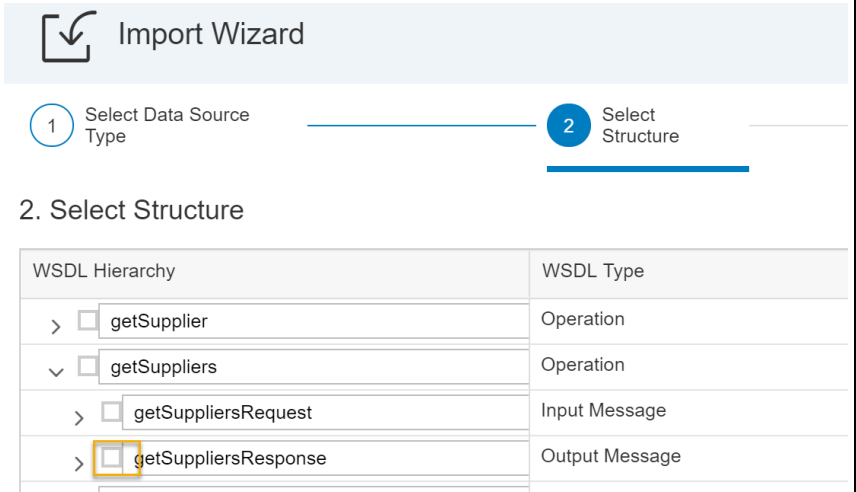
Expand (▸) operation **getSuppliers**

CAUTION: Don't choose getSupplier.

2. Select Structure

WSDL Hierarchy	WSDL Type
> <input type="checkbox"/> getSupplier	Operation
▾ <input type="checkbox"/> getSuppliers	Operation
> <input type="checkbox"/> getSuppliersRequest	Input Message
> <input type="checkbox"/> getSuppliersResponse	Output Message
> <input type="checkbox"/> generateData	Operation
> <input type="checkbox"/> deleteSupplier	Operation
> <input type="checkbox"/> updateSupplier	Operation
> <input type="checkbox"/> createSupplier	Operation

14) Check segment **getSuppliersResponse**.

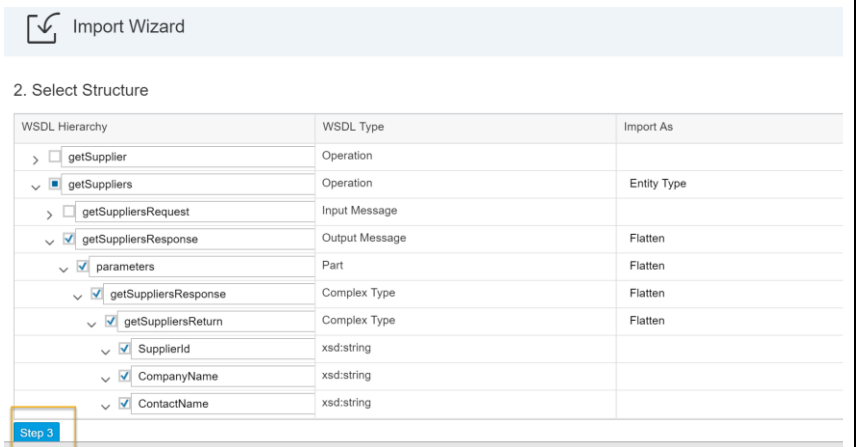


The screenshot shows the 'Import Wizard' interface at step 2, 'Select Structure'. At the top, there are two progress indicators: '1 Select Data Source Type' and '2 Select Structure', with the second one being active. Below the indicators, the title '2. Select Structure' is displayed. A table lists WSDL elements and their types. The element 'getSuppliersResponse' is highlighted with a yellow box.

WSDL Hierarchy	WSDL Type
> <input type="checkbox"/> getSupplier	Operation
✓ <input type="checkbox"/> getSuppliers	Operation
> <input type="checkbox"/> getSuppliersRequest	Input Message
> <input type="checkbox"/> getSuppliersResponse	Output Message

15) With that, all sub elements are selected automatically. Leave the other settings as it is.

Click on button **Step 3**.

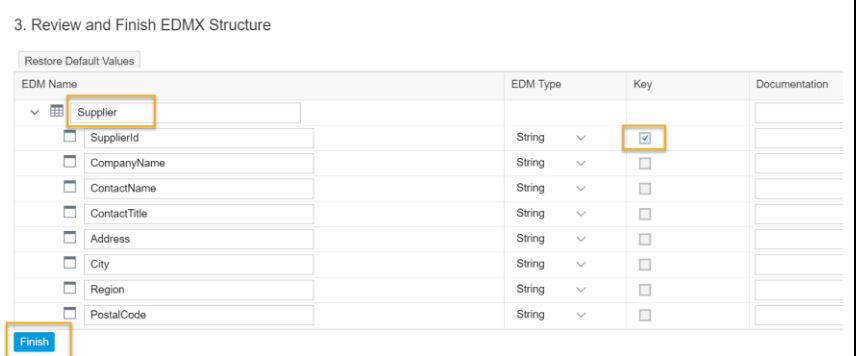


The screenshot shows the 'Import Wizard' interface at step 2, 'Select Structure'. The table now includes an 'Import As' column. The 'getSuppliersResponse' element and its sub-elements are selected, and their 'Import As' values are set to 'Flatten'. The 'Step 3' button is highlighted with a yellow box.

WSDL Hierarchy	WSDL Type	Import As
> <input type="checkbox"/> getSupplier	Operation	
✓ <input type="checkbox"/> getSuppliers	Operation	Entity Type
> <input type="checkbox"/> getSuppliersRequest	Input Message	
✓ <input checked="" type="checkbox"/> getSuppliersResponse	Output Message	Flatten
✓ <input checked="" type="checkbox"/> parameters	Part	Flatten
✓ <input checked="" type="checkbox"/> getSuppliersResponse	Complex Type	Flatten
✓ <input checked="" type="checkbox"/> getSuppliersReturn	Complex Type	Flatten
✓ <input checked="" type="checkbox"/> SupplierId	xsd:string	
✓ <input checked="" type="checkbox"/> CompanyName	xsd:string	
✓ <input checked="" type="checkbox"/> ContactName	xsd:string	

16) Change the default settings as follows:

- Change **getSuppliers** to **Supplier**
- Check **Key** for field **SupplierID**
- Click on button **Finish**.



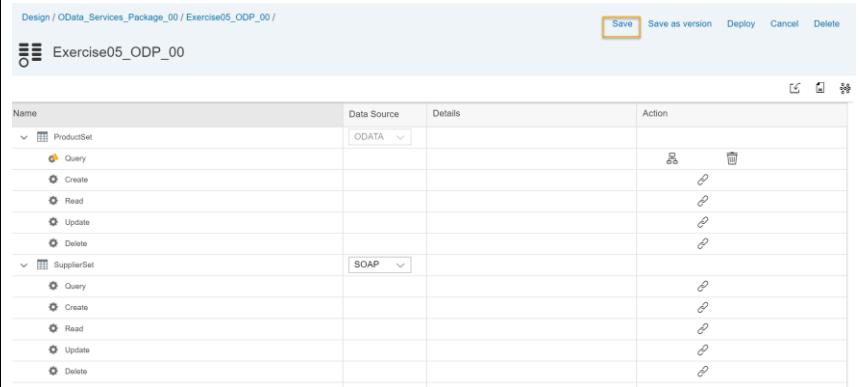
The screenshot shows the 'Import Wizard' interface at step 3, 'Review and Finish EDMX Structure'. A table lists EDM entities and their fields. The 'Supplier' entity is highlighted with a yellow box, and the 'SupplierID' field is marked as a 'Key'. The 'Finish' button is highlighted with a yellow box.


EDM Name	EDM Type	Key	Documentation
Supplier		<input checked="" type="checkbox"/>	
SupplierId	String	<input type="checkbox"/>	
CompanyName	String	<input type="checkbox"/>	
ContactName	String	<input type="checkbox"/>	
ContactTitle	String	<input type="checkbox"/>	
Address	String	<input type="checkbox"/>	
City	String	<input type="checkbox"/>	
Region	String	<input type="checkbox"/>	
PostalCode	String	<input type="checkbox"/>	

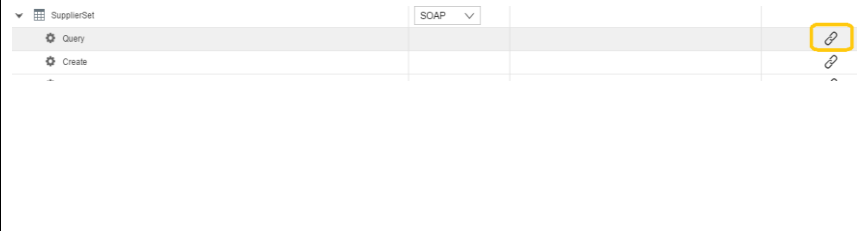
17) As a result, the **SupplierSet** structure from SOAP data source is added to the **ProductSet** structure of previous exercise.

Before moving on let's save configurations made.

Click on the **Save** button.



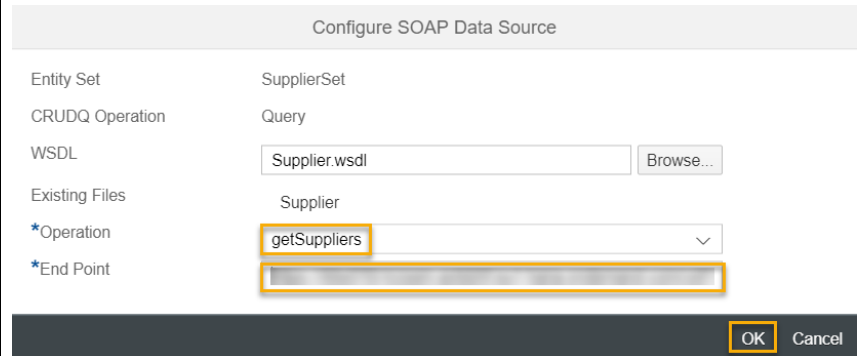
18) Next you need to bind the operation to a data source. Let's do that for the **Query** operation to get a list of customers.
In **SupplierSet** click on the **Bind** icon () in the row of the **Query** operation.



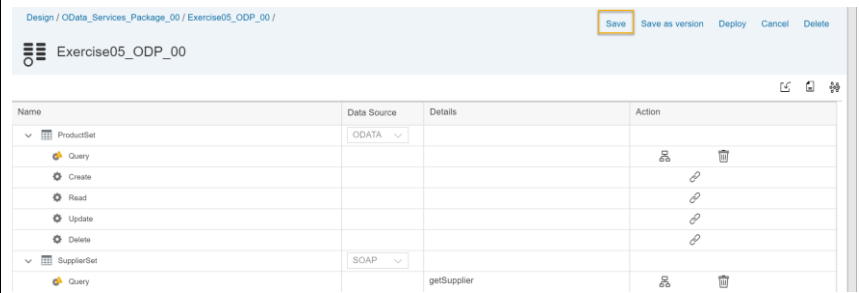
19) You now configure the data source of the SOAP service. Some values are already populated.


For the missing ones do the following:


- **Operation:** getSuppliers
- **End Point:** Change the **End Point** information with the endpoint URL of the SOAP simulator service integration scenario deployed on your tenant.
Refer to Step 2
- Finally click on **OK** button.









20) Save your changes by clicking on **Save** option in the top right corner.



21) You can see that there is an integration flow () added to the query operation. This is an auto generated integration flow which you will configure in the next steps for connecting your new OData service with the SOAP data source and implement needed mappings.

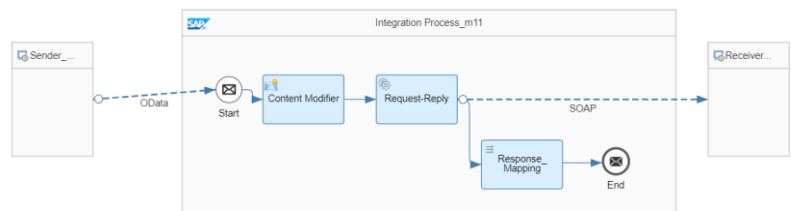
Click on the integration flow icon ().

▼ SupplierSet	SOAP ▼		
Query		getSuppliers	 
Create			
Read			
Update			
Delete			

22) Even though the preconfigured integration flow for SOAP sources looks like the one of an OData source, the configuration steps are slightly different.

The following steps need to be performed:

- Configure **Content Modifier** to define the message format of the request message
- Implement **Response Mapping**
- Configure **SOAP channel** to connect with the SOAP data source



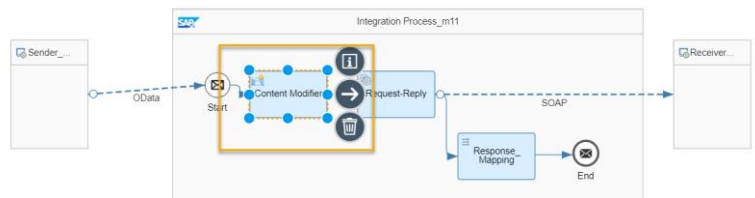
23) Let's first configure the **Content Modifier**

Click on the **Content Modifier** step within the integration flow.

Switch to the **Message Body** tab

Copy and paste the following XML structure into the editor of the Message Body:

```
<web:getSuppliers
xmlns:web="http://webservice.supplier.test.org">
  <web:City></web:City>
  <web:PostalCode>
</web:PostalCode>
</web:getSuppliers>
```



Content Modifier

General Message Header Exchange Property

Message Body

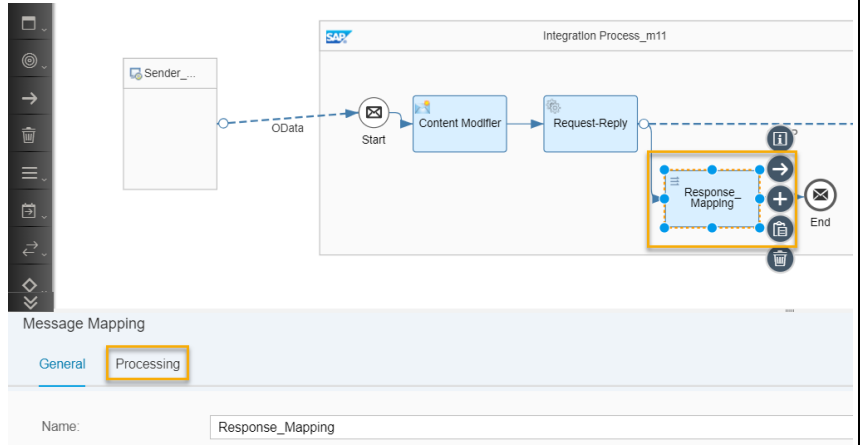
Body:

```
<web:getSuppliers xmlns:web="http://webservice.supplier.test.org">
<web:City></web:City>
<web:PostalCode></web:PostalCode>
</web:getSuppliers>
```

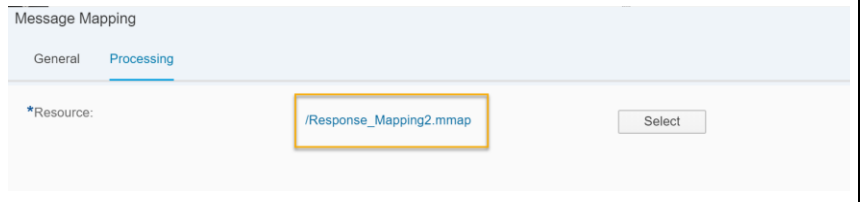
24) Let's implement the response mapping to map the message (here: Supplier) from the SOAP data source to the new OData service (here: SupplierSet).

Select **Response_Mapping** step in the design canvas.

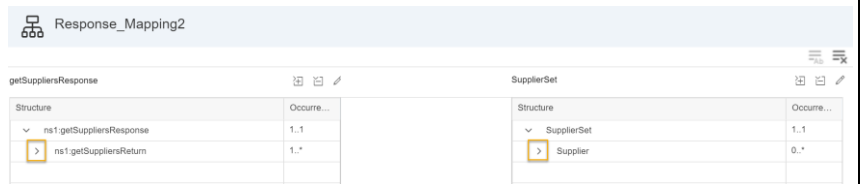
Navigate to **Processing** tab.



25) In **Processing** tab, click on the mapping's name which is **Response_Mapping2.mmap**. This will launch the mapping editor.



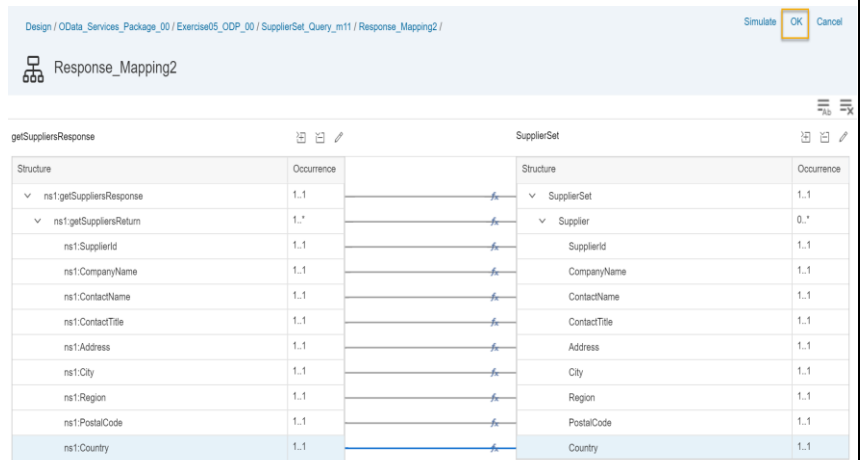
26) On both sides expand the nodes by clicking on the expand icons (➤) for showing up all elements to map.



27) Implement the following mapping program:

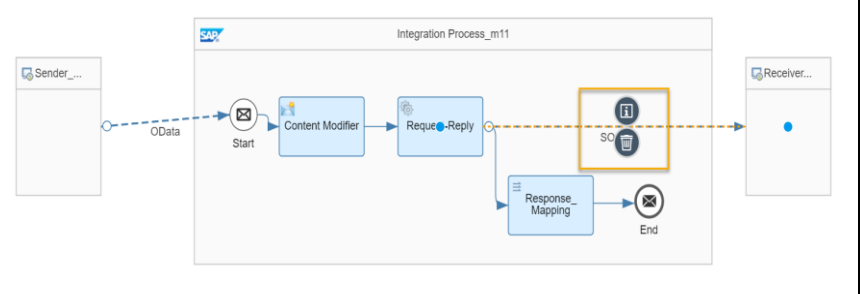
- Click **ns1:getSuppliersResponse** on the left and drag & drop it to the corresponding field **SupplierSet** on the right.
- Repeat these steps until all fields on the left of the **SupplierSet** message on the right are mapped.

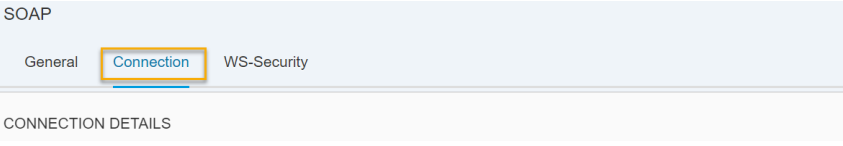

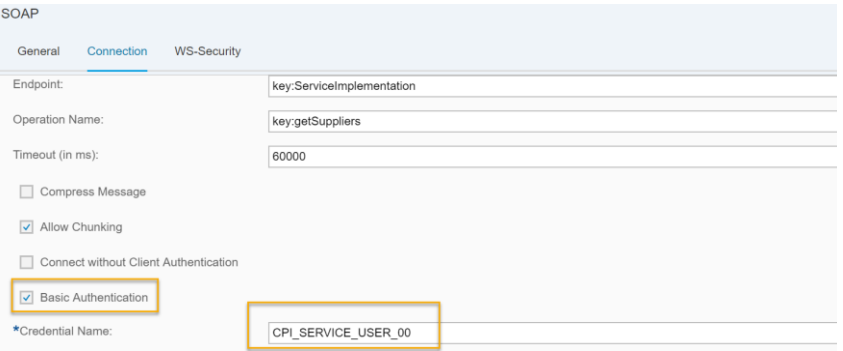
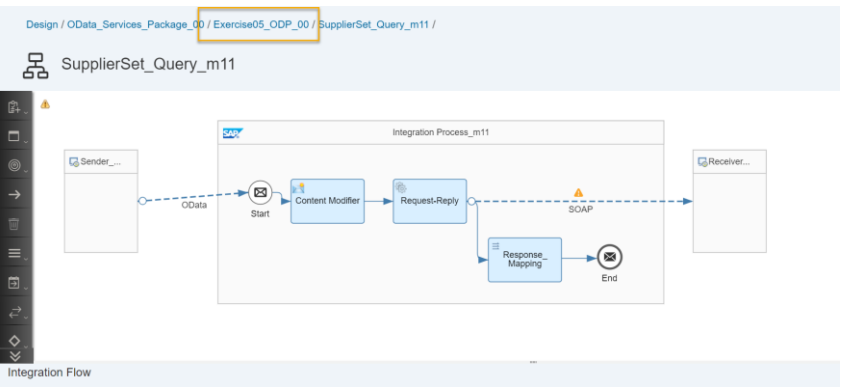
Click on **OK** for saving the mapping. You are directed back to the Integration Flow Designer.

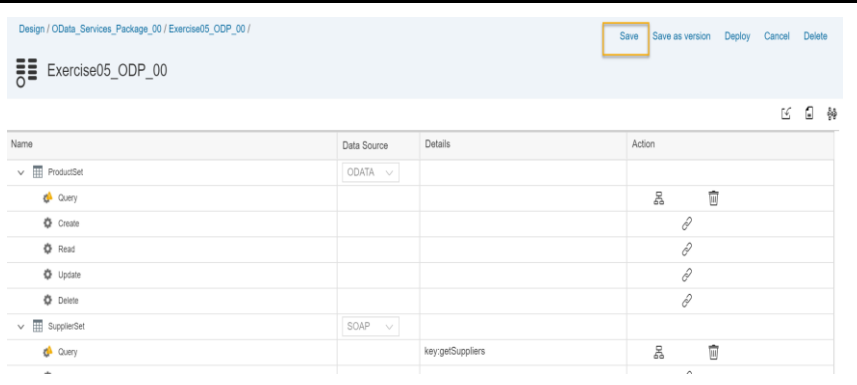
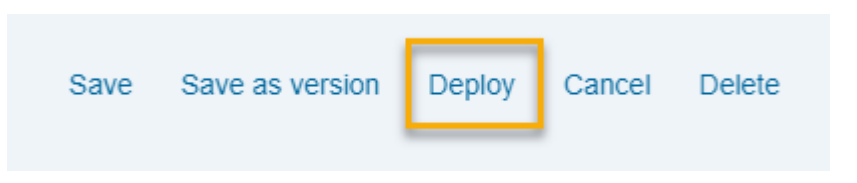
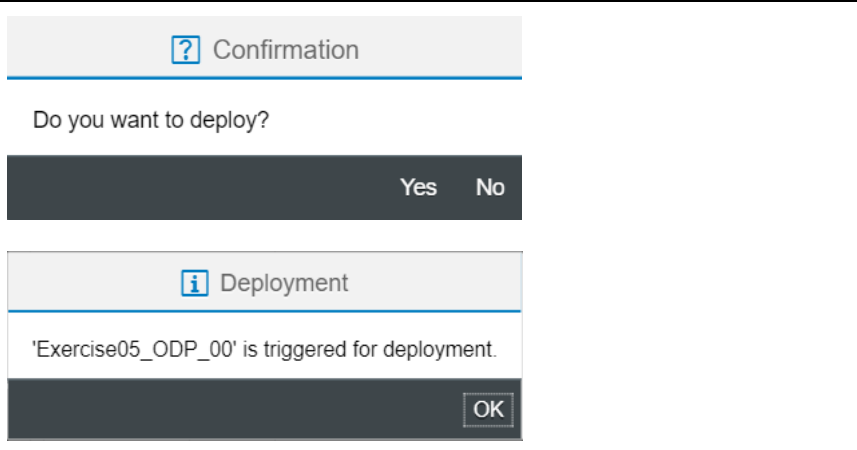




28) Next you need to configure the SOAP channel for getting the needed order data from the SOAP data source.

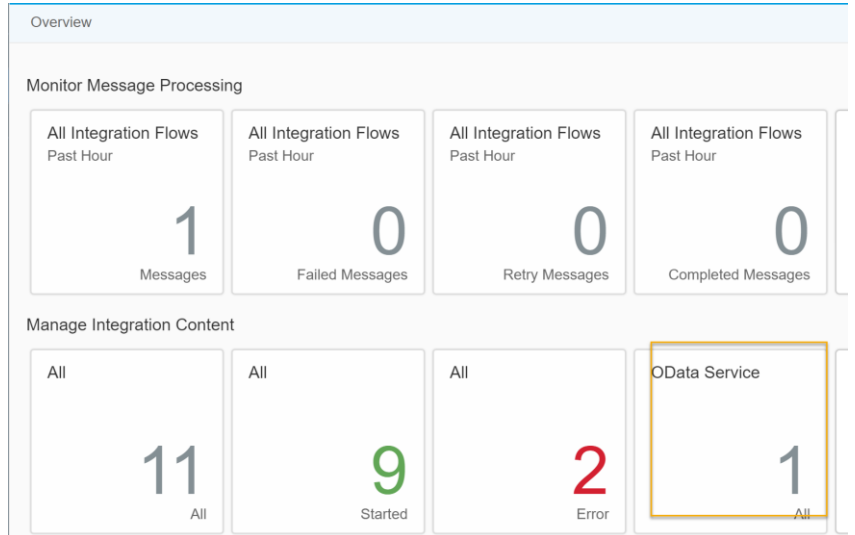
Select the connection (channel) between **Request-Reply** step and **SOAP_Receiver**.



<p>29) Switch to Connection tab.</p>	
<p>30) The connection details of the SOAP channel are already prefilled with the settings derived from the WSDL file.</p>	
<p>31) Add Authentication details:</p> <ol style="list-style-type: none"> Scroll below to check Basic Authentication option. For Credential Name enter : CPI_SERVICE_USER_XX Note: Replace XX with your Group No. You had already deployed this credential in Step 5 Click on OK in the top right corner to save the changes 	
<p>32) Click on the Navigation Path “Exercises05_ODP_00” to go back to main integration scenario screen.</p>	

<p>33) Click on Save to save the changes.</p>	 <p>Design / OData_Services_Package_00 / Exercise05_ODP_00 /</p> <p>Exercise05_ODP_00</p> <table><thead><tr><th>Name</th><th>Data Source</th><th>Details</th><th>Action</th></tr></thead><tbody><tr><td>ProductSet</td><td>ODATA</td><td></td><td></td></tr><tr><td>Query</td><td></td><td></td><td>Query</td></tr><tr><td>Create</td><td></td><td></td><td>Create</td></tr><tr><td>Read</td><td></td><td></td><td>Read</td></tr><tr><td>Update</td><td></td><td></td><td>Update</td></tr><tr><td>Delete</td><td></td><td></td><td>Delete</td></tr><tr><td>SupplierSet</td><td>SOAP</td><td></td><td></td></tr><tr><td>Query</td><td></td><td>key getSuppliers</td><td>Query</td></tr></tbody></table>	Name	Data Source	Details	Action	ProductSet	ODATA			Query			Query	Create			Create	Read			Read	Update			Update	Delete			Delete	SupplierSet	SOAP			Query		key getSuppliers	Query
Name	Data Source	Details	Action																																		
ProductSet	ODATA																																				
Query			Query																																		
Create			Create																																		
Read			Read																																		
Update			Update																																		
Delete			Delete																																		
SupplierSet	SOAP																																				
Query		key getSuppliers	Query																																		
<p>Deploy Integration Project on tenant:</p>	<p>Follow this step to deploy Integration Project on tenant.</p>																																				
<p>34) Press Deploy in the OData Service Task Bar.</p>	 <p>Save Save as version Deploy Cancel Delete</p>																																				
<p>35) You will receive a confirmation. Click on Yes.</p> <p>Once the deployment is completed successfully you will receive a 2nd notification.</p>	 <p>Confirmation</p> <p>Do you want to deploy?</p> <p>Yes No</p> <p>Deployment</p> <p>'Exercise05_ODP_00' is triggered for deployment.</p> <p>OK</p>																																				
<p>36) Verify if the deployment is successful:</p> <p>Go to the Monitoring of CPI to check the status of the deployed OData service: Click on  to navigate to the Monitor section.</p>	 <p>SAP</p> <p>Discover</p> <p>Design</p> <p>Monitor</p> <p>Settings</p>																																				

37) In the Monitor section go to *Manage Integration Content* and click on **OData Service (All)**.



38) Your OData service should be listed there.

Overview / Manage Integration Content

Integration Content (11) 🔍 🔄 ⚙️

Name	Status
Exercise05_ODP_00	Started
OData Service	

39) Make sure that the status of the service is **Started** and that Artifact Details such as *Name*, *Namespace* and *Version* are shown.

If the status of your OData Services is other than **Started**, then wait a while and refresh the monitoring data by clicking on the *refresh* (🔄) icon at the top to check if the status changed in the meantime.

Overview / Manage Integration Content

Integration Content (11) 🔍 🔄 ⚙️

Name	Status
Exercise05_ODP_00	Started
OData Service	

Exercise05_ODP_00 Restart Undeploy

Deployed On: Jun 07, 2018, 15:45:45 ID: Exercise05_ODP_00_SAP_1
Deployed By: P000012 Version: 1.0.0

Endpoints Status Details Artifact Details Log Configuration

There are no endpoints configured.

Status Details

The OData Service is deployed successfully.

Artifact Details

Monitor Message Processing
Name: EXERCISE05_ODP_00
Namespace: SAP
Version: 1

Execute end to end scenario:

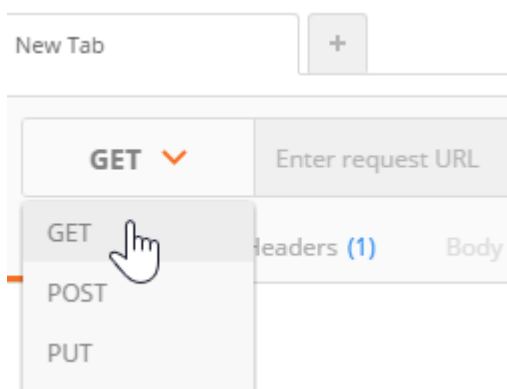
Follow steps to execute end to end scenario.

40) Go back to **Postman**.



41) First let's make sure that the OData service was created successfully.

Select **GET** as operation option in Postman



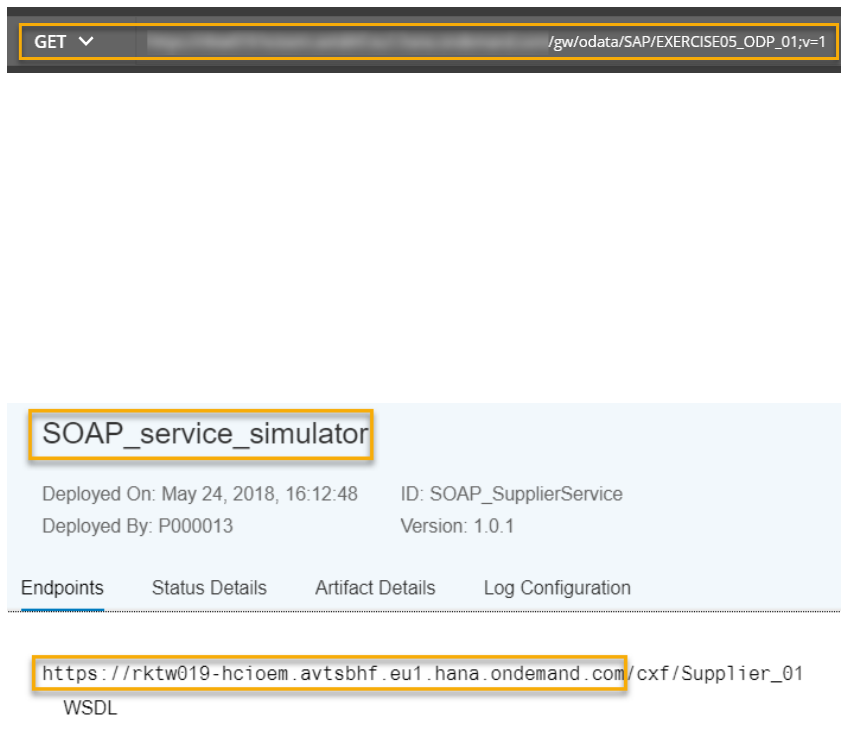
42) Enter the URL pointing to your OData Service. Replace the values highlighted in **RED**

https://<your runtime node hostname>/gw/odata/SAP/EXERCISE05_ODP_XX;v=1

Note: Please replace XX with your Group_No

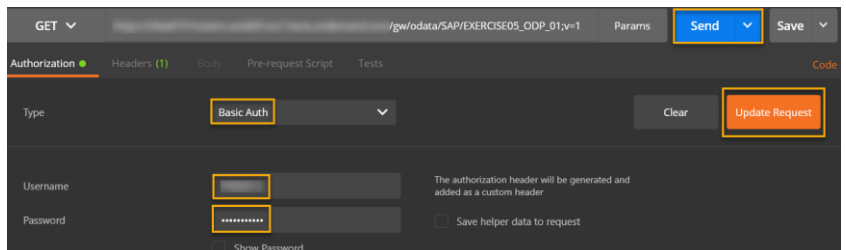
How to get runtime node hostname?

1. Go to monitoring section
2. Select Manage Integration Content(Started)
3. Search and select **SOAP_service_simulator** integration content that is already deployed.
4. Copy hostname from the **Endpoints** section which is highlighted in yellow

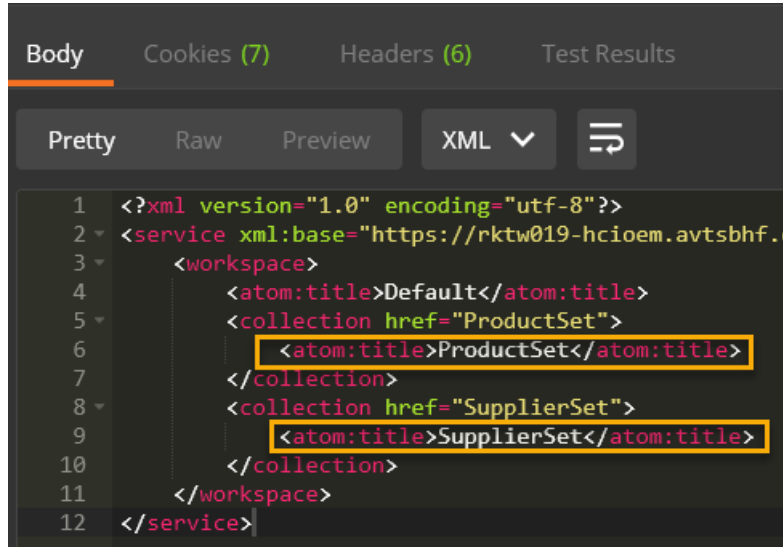


43) For Authorization choose Type as **Basic Auth** from the dropdown list.

Enter your user Id and password of CPI tenant. Don't forget to click on **Update Request**. And finally click on **Send** for sending out the request to your new OData service on CPI



44) If you get a response which shows that a collection **ProductSet** and **SupplierSet** are available, then both entity sets are registered on the CPI tenant.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <service xml:base="https://rktw019-hcioem.avtsbhf.
3 <workspace>
4   <atom:title>Default</atom:title>
5   <collection href="ProductSet">
6     <atom:title>ProductSet</atom:title>
7   </collection>
8   <collection href="SupplierSet">
9     <atom:title>SupplierSet</atom:title>
10  </collection>
11 </workspace>
12 </service>
```

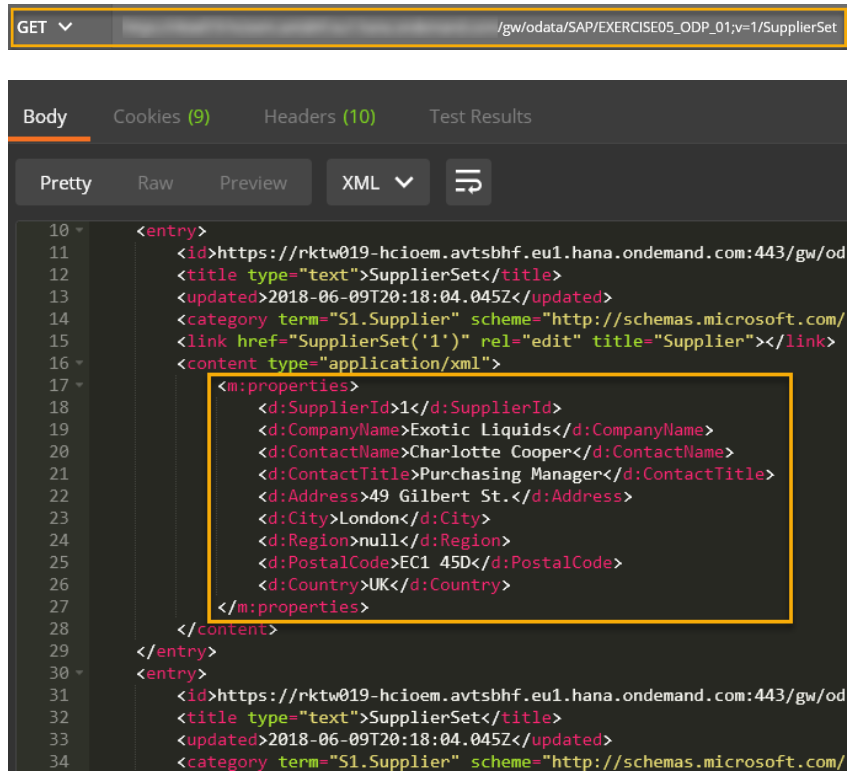
69) Next, call GET operation of this OData service for retrieving the list of suppliers out of the OData source.

In Postman

- append **/SupplierSet** to the previous URL
- click on **Send**

As response, you can see that a list of suppliers with details is listed

Congratulations, with that you have successfully created a single OData service in CPI which provide data from two different data sources i.e. OData and SOAP



```
GET /gw/odata/SAP/EXERCISE05_ODP_01;v=1/SupplierSet

Body Cookies (9) Headers (10) Test Results

10 <entry>
11   <id>https://rktw019-hcioem.avtsbhf.eu1.hana.ondemand.com:443/gw/od
12   <title type="text">SupplierSet</title>
13   <updated>2018-06-09T20:18:04.045Z</updated>
14   <category term="S1.Supplier" scheme="http://schemas.microsoft.com/
15   <link href="SupplierSet('1')" rel="edit" title="Supplier"></link>
16   <content type="application/xml">
17     <m:properties>
18       <d:SupplierId>1</d:SupplierId>
19       <d:CompanyName>Exotic Liquids</d:CompanyName>
20       <d:ContactName>Charlotte Cooper</d:ContactName>
21       <d:ContactTitle>Purchasing Manager</d:ContactTitle>
22       <d:Address>49 Gilbert St.</d:Address>
23       <d:City>London</d:City>
24       <d:Region>null</d:Region>
25       <d:PostalCode>EC1 4SD</d:PostalCode>
26       <d:Country>UK</d:Country>
27     </m:properties>
28   </content>
29 </entry>
30 <entry>
31   <id>https://rktw019-hcioem.avtsbhf.eu1.hana.ondemand.com:443/gw/od
32   <title type="text">SupplierSet</title>
33   <updated>2018-06-09T20:18:04.045Z</updated>
34   <category term="S1.Supplier" scheme="http://schemas.microsoft.com/
```