

openSAP Simplify Integration with SAP Cloud platform Integration Suite

Week 2 Unit 7: Expose Back-End Services as OData APIs

PUBLIC

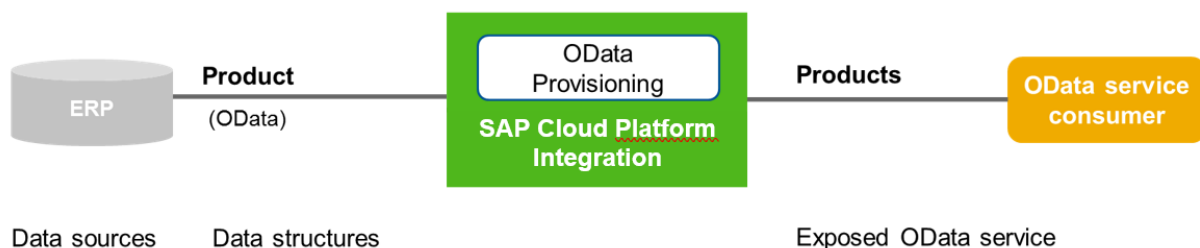


INTEGRATION SCENARIO

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 Mobiles Services or any other custom applications to fulfil user-centric scenarios.

In this exercise, you will learn how to create and expose OData API from an existing data source using OData Provisioning in SAP Cloud Platform Integration.

You will create an OData service out of a publicly available free OData service which delivers product data. You can also extend this exercise by adding additional data sources and do a mash up of services as shown in the demo of Week 2 Unit 7.



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:

How to ...

- import an OData model from an OData 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.

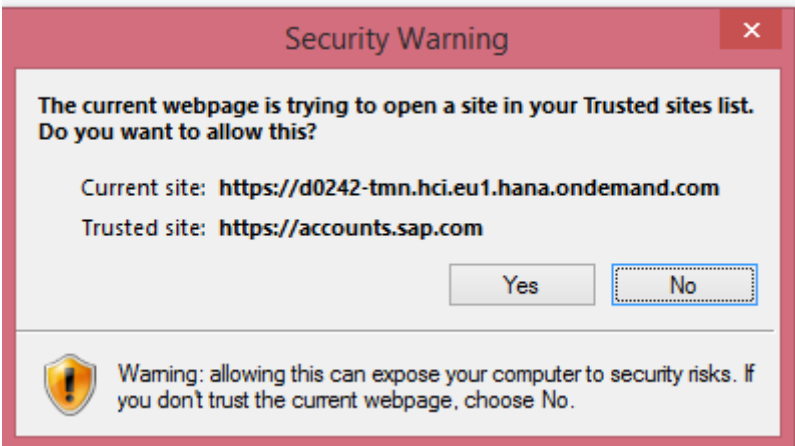
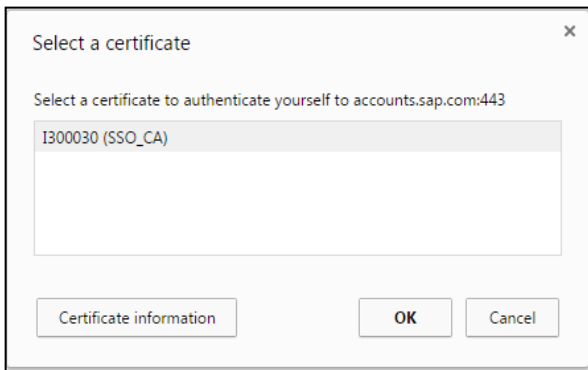
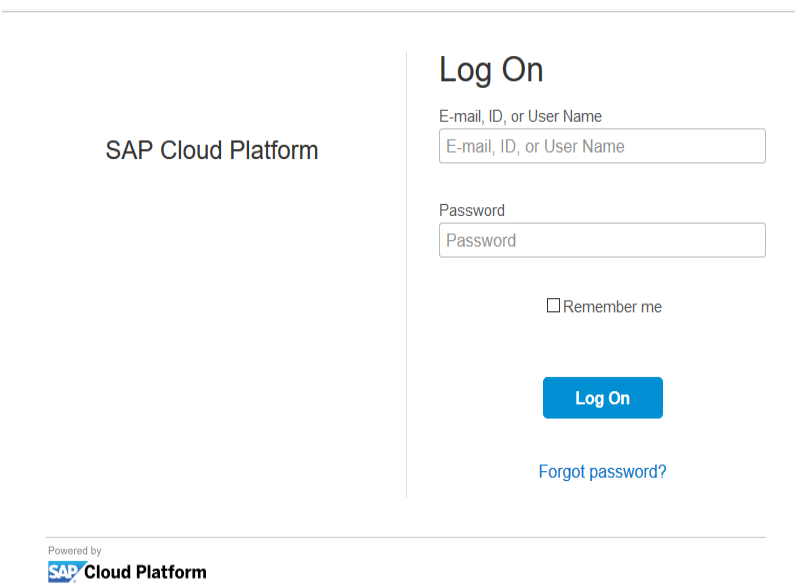
Exercise Files

Exercise files are provided in the same folder as this document. 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 uniquely. Please replace **XX** with your S/P/I/D User ID to uniquely identify your integration content from other participants using the same tenant.
2. 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 STEPS

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 titled "Security Warning" with a red border and a close button (X) in the top right corner. The main text 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 right are "Yes" and "No" buttons, with "No" highlighted by a dashed blue border. At the bottom left is a yellow shield icon with an exclamation mark. To its right, a warning message states: "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 dialog box titled "Select a certificate" with a close button (X) in the top right corner. The text inside says: "Select a certificate to authenticate yourself to accounts.sap.com:443". Below this is a list box containing "I300030 (SSO_CA)". At the bottom 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. Enter the following credentials:</p> <p><i>User: S/P/I/D User ID</i> <i>Password: S/P/I/D's Password</i></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 for "E-mail, ID, or User Name" and "Password". Below these is a checkbox for "Remember me". A blue "Log On" button is positioned below the checkbox. At the bottom right, there is a link for "Forgot password?". At the very bottom, it says "Powered by SAP Cloud Platform" with the SAP logo.</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 OData provisioning project for this Exercise

Click on **Create**

Enter Details

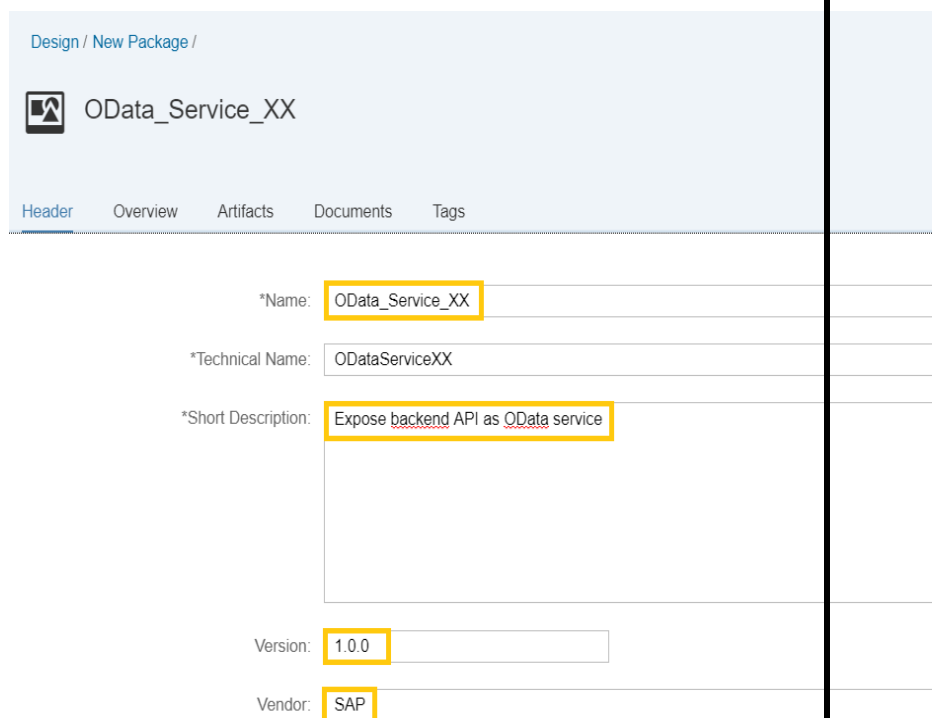
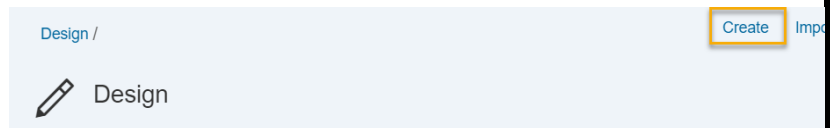
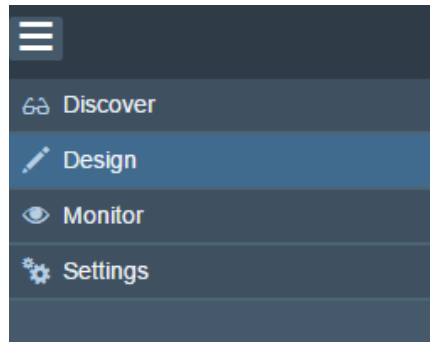
- a) **Name:** OData_Service_XX

NOTE: Replace **XX** with your S/P/I/D User ID

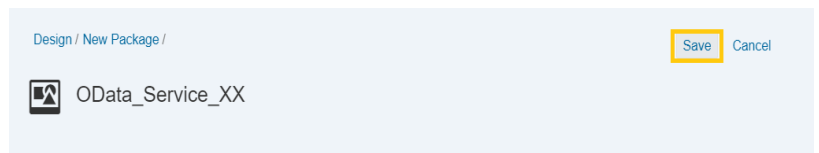
- b) **Short Description:** Expose backend API as OData service

- c) **Version:** 1.0.0

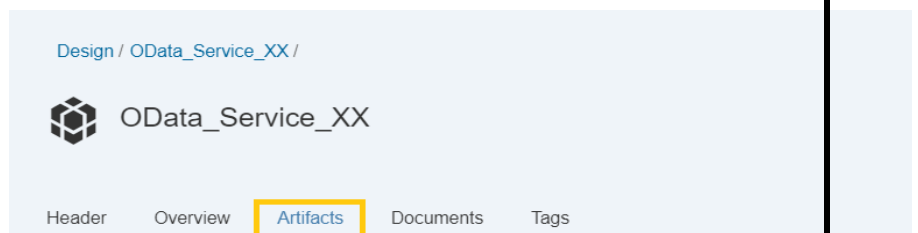
- d) **Vendor:** SAP



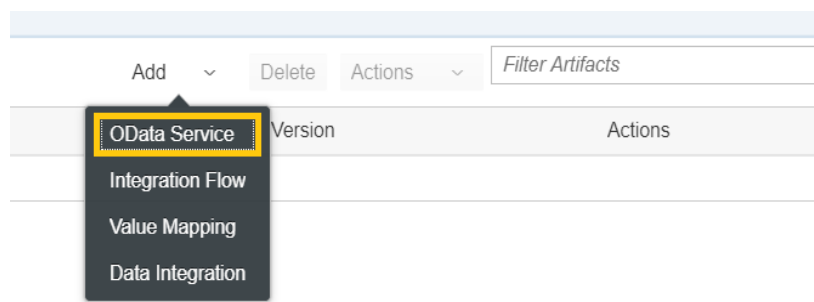
- 5) Click on Save



- 6) Navigate to **Artifacts** Tab



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

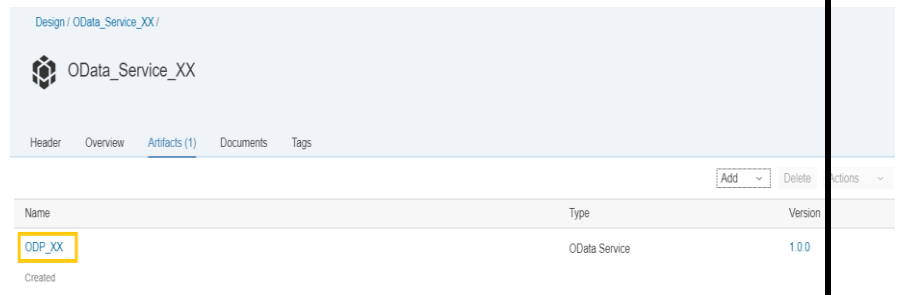


8) Enter as name for your OData service **ODP_XX** and click on **OK**.

Hint: ODP stands for OData provisioning.


NOTE: Replace **XX** with your S/P/I/D User ID

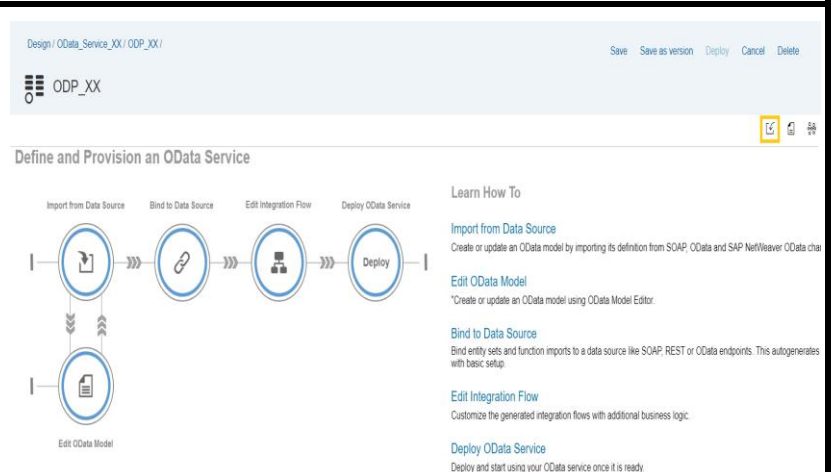
9) Click on the newly created OData Service.



10) The overview page of OData provisioning launches.

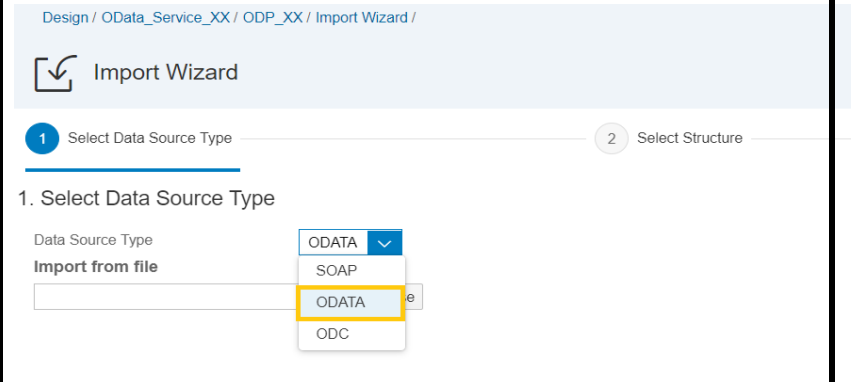
Click on **Edit**,

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



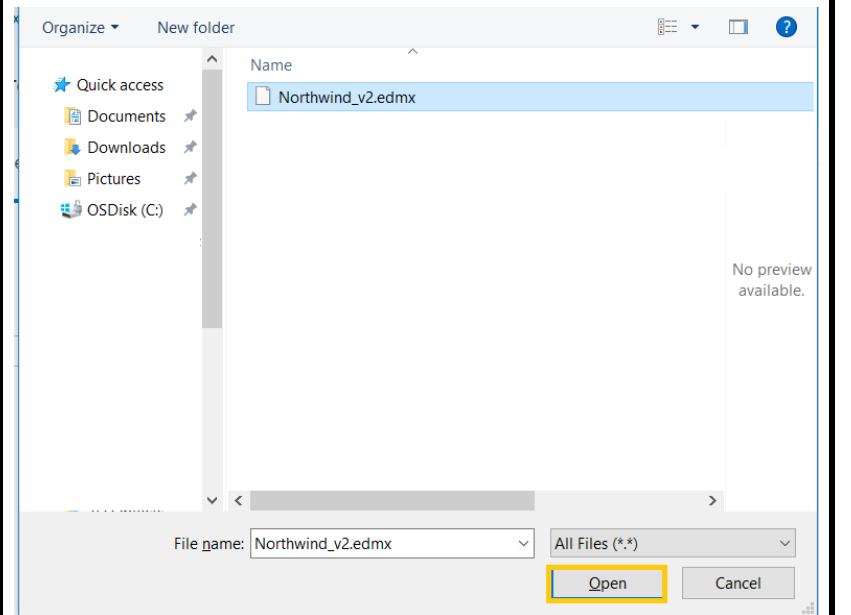
11) First you select a data source and import the respective definition file from your file system.

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

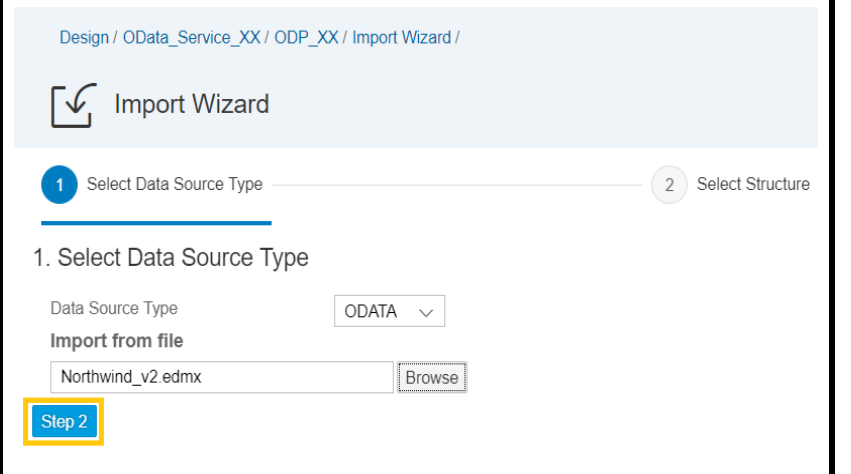


12) Now let's import an OData definition file (.edmx) of a public OData service which includes data structures for Products:

Go to folder where you have downloaded the exercise files, select file **Northwind_v2.edmx** and click on **Open**.



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



14) 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.

Import Wizard

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

2. Select Structure

OData Hierarchy	EDM Type
> <input type="checkbox"/> Category	Entity Type
> <input type="checkbox"/> CustomerDemographic	Entity Type
> <input type="checkbox"/> Customer	Entity Type
> <input type="checkbox"/> Employee	Entity Type
> <input type="checkbox"/> Order_Detail	Entity Type
> <input type="checkbox"/> Order	Entity Type
> <input checked="" type="checkbox"/> Product	Entity Type
> <input type="checkbox"/> Region	Entity Type
> <input type="checkbox"/> Shipper	Entity Type
> <input type="checkbox"/> Supplier	Entity Type

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

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

Click on button **Step 3**.

Import Wizard

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

2. Select 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

16) 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**.

Import Wizard

1 Select Data Source Type ————— 2 Select Structure ————— 3 **Review and Finish EDMX Structure**


3. Review and Finish EDMX Structure

Restore Default Values

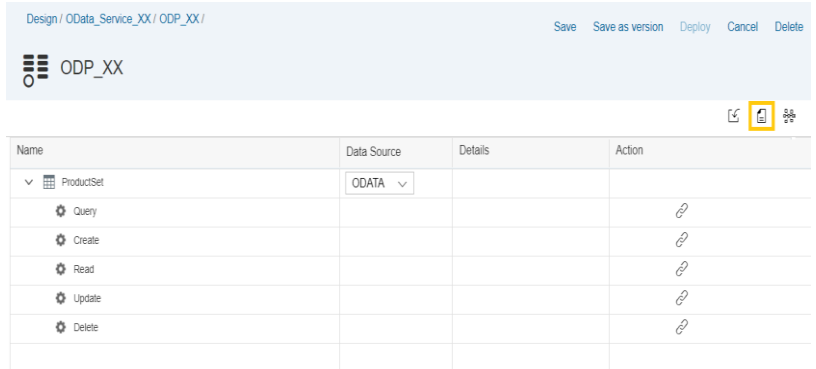
EDM Name	EDM Type	Key	Documentation
Product			
<input type="checkbox"/> ProductID	Int32	<input checked="" type="checkbox"/>	
<input type="checkbox"/> ProductName	String	<input type="checkbox"/>	
<input type="checkbox"/> SupplierID	Int32	<input type="checkbox"/>	
<input type="checkbox"/> CategoryID	Int32	<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

17) 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.



18) In this exercise, we will add two new properties to the OData structure.

Place your cursor after the last property end tag (`</Property>`) and press **ENTER**. This will add an empty row.

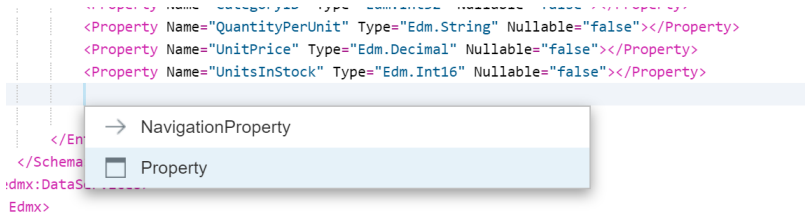
```

1 <!-- Edmx -->
2 xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx"
3 xmlns:sap="http://www.sap.com/Protocols/SAPData" Version="1.0"
4 <!-- DataServices -->
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="true"></Property>
18   <Property Name="SupplierID" Type="Edm.Int32" Nullable="true"></Property>
19   <Property Name="CategoryID" Type="Edm.Int32" Nullable="true"></Property>
20   <Property Name="QuantityPerUnit" Type="Edm.String" Nullable="true"></Property>
21   <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="true"></Property>
22   <Property Name="UnitsInStock" Type="Edm.Int16" Nullable="true"></Property>
23 </EntityType>
24 </Schema>
25 </edmx:DataServices>
26 </edmx:Edmx>

```

19) Press keys **CTRL + SPACEBAR** on your keyboard for launching the *Schema-Based Code Assist* of the *OData Model Editor*.

Select **Property** from the context menu.



20) The parameters for defining a property are inserted automatically.

```

20 <Property Name="QuantityPerUnit" Type="Edm.String" Nullable="true"></Property>
21 <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="true"></Property>
22 <Property Name="UnitsInStock" Type="Edm.Int16" Nullable="true"></Property>
23 <Property Name="" Type="" Nullable="true"></Property>
24 </EntityType>
25 </Schema>
26 </edmx:DataServices>
27 </edmx:Edmx>

```

21) Enter the following values:

- **Name:** OrderPrice
- **Type:** Edm.Decimal
- **Nullable:** false

```

21 <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="true"></Property>
22 <Property Name="UnitsInStock" Type="Edm.Int16" Nullable="true"></Property>
23 <Property Name="OrderPrice" Type="Edm.Decimal" Nullable="false"></Property>
24 </EntityType>
25 </Schema>
26 </edmx:DataServices>
27 </edmx:Edmx>

```

22) Let's add another new property to the EDMX file.

Again, place the cursor after the last property end tag (`</Property>`) and press **ENTER** for adding a new row.

```

20 <Property Name="QuantityPerUnit" Type="Edm.String" Nullable="true"></Property>
21 <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="true"></Property>
22 <Property Name="UnitsInStock" Type="Edm.Int16" Nullable="true"></Property>
23 <Property Name="OrderPrice" Type="Edm.Decimal" Nullable="false"></Property>
24 </EntityType>
25 </Schema>
26 </edmx:DataServices>
27 </edmx:Edmx>

```


23) Press keys CTRL + SPACEBAR on your keyboard for launching the *Schema-Based Code Assist* of the *OData Model Editor* once more.

Select **Property** from the context menu.

```

23      <Property Name="UnitsOnOrder" Type="Edm.Int16" Nullable="false"></Property>
24      <Property Name="OrderPrice" Type="Edm.Decimal" Nullable="false"></Property>
25
26    </EntityType>
27  </Schema>
28 </edmx:DataServices>
29 </edmx:Edmx>

```

24) Enter the following values:

- **Name:** TargetStock
- **Type:** Edm.Int16
- **Nullable:** false

```

21      <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="true"></Property>
22      <Property Name="UnitsInStock" Type="Edm.Int16" Nullable="true"></Property>
23      <Property Name="OrderPrice" Type="Edm.Decimal" Nullable="false"></Property>
24      <Property Name="TargetStock" Type="Edm.Int16" Nullable="false"></Property>
25
26    </EntityType>
27  </Schema>
28 </edmx:DataServices>
29 </edmx:Edmx>

```






25) 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.

26) Click on Navigation Path “ODP_XX” to go back to main integration scenario screen

27) 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			

28) 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

Entity Sets: Products

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

[OK] [Cancel]

29) 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.

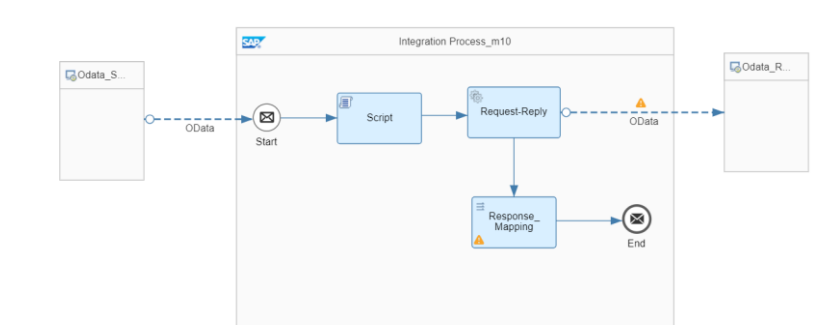
Click on the integration flow icon (🔌).

Name	Data Source	Details	Action
ProductSet	ODATA	Products	[Integration Flow Icon]
Query			[Integration Flow Icon]
Create			[Integration Flow Icon]
Read			[Integration Flow Icon]
Update			[Integration Flow Icon]
Delete			[Integration Flow Icon]

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

You only need to implement the following:

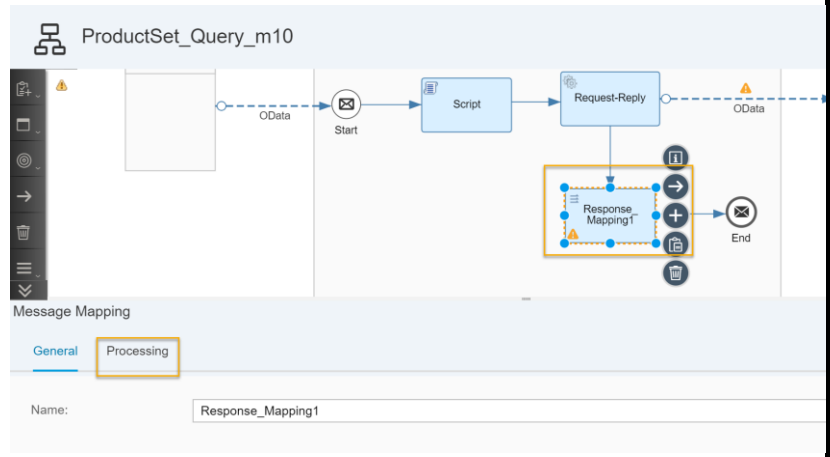
- Response Mapping
- Configure the OData channel to connect with the OData source



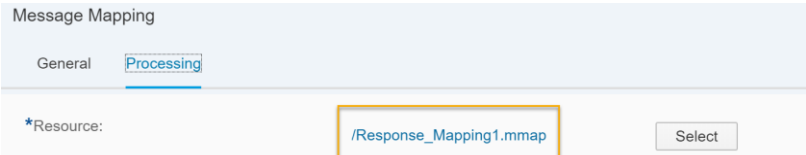
31) Let's implement the response mapping to map the message (here: Products) from the OData source to the new OData service (here: ProductSet).

Select **Response Mapping** step in the design canvas.

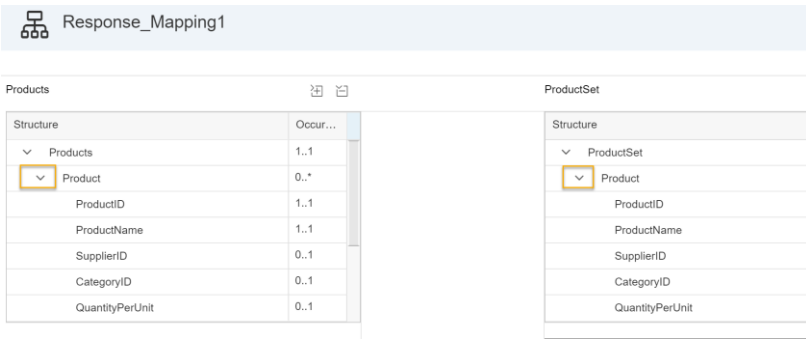
Navigate to **Processing** tab.



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

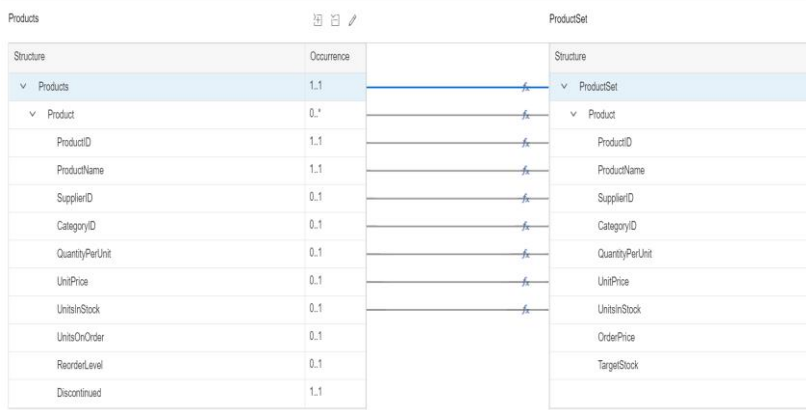


33) On both sides, expand the nodes by clicking on the expand icons (➤) next to **Products** on the left and **ProductSet** on the right-hand side which would display the attributes.



34) Implement the mapping program:

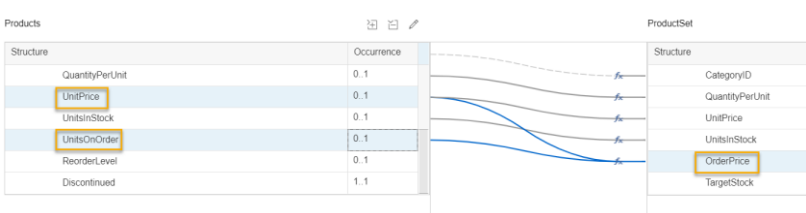
- Click **Products** on the left and drag & drop it to the corresponding field of **ProductSet** on the right.
- Repeat the mapping of following elements:
Product
ProductID,
ProductName,
SupplierID,
CategoryID,
QuantityPerUnit,
UnitPrice,
UnitsInStock

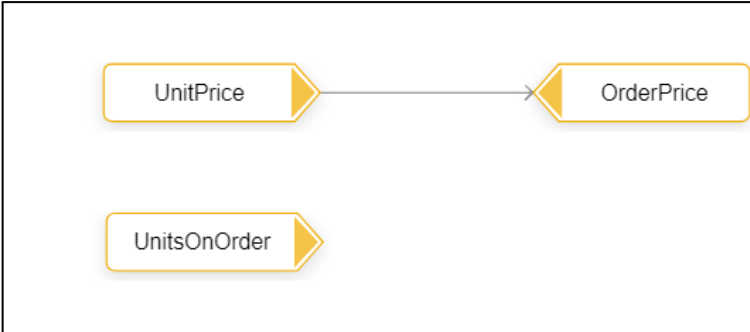
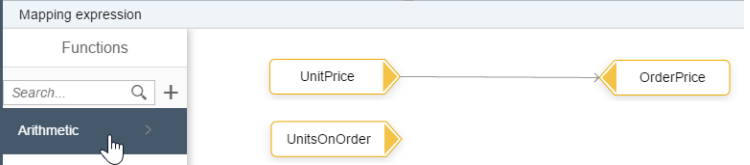
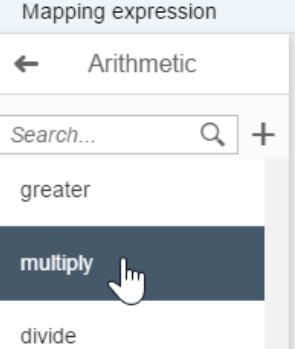
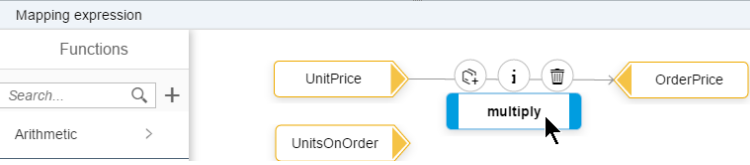
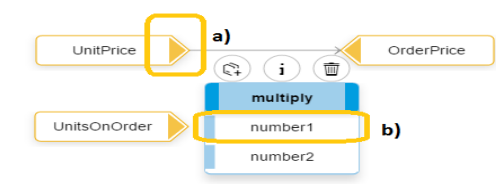
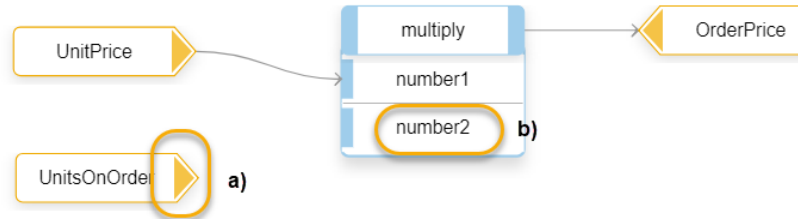


35) For the two new elements that we added to the EDMX file using the OData Model Editor, we need to implement a mapping using a predefined mapping function.

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**.

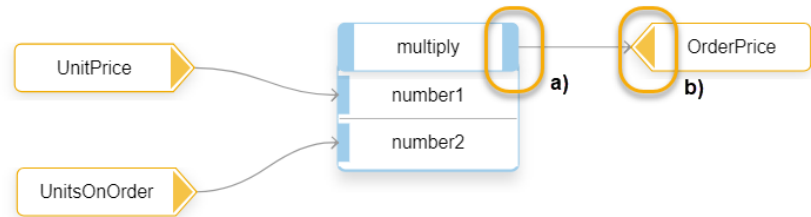
First, drag & drop **UnitPrice** of the source message to **OrderPrice**.



<p>Second, drag & drop UnitOnOrder of the source structure to OrderPrice of the target structure</p> <p>In the Mapping Expression Editor UnitPrice and UnitsOnOrder are shown on the left and OrderPrice on the right.</p> <p>Hint: Move OrderPrice a bit to the right for adding more space to add the mapping function.</p>	
<p>36) From the list of <i>Functions</i> click on Arithmetic.</p>	
<p>37) The list of arithmetical mapping functions appears.</p> <p>Click on multiply.</p>	
<p>38) The multiply function is added to the canvas of the <i>Mapping Expression Editor</i>.</p> <p>Select the same and move it in the middle between all three elements.</p>	
<p>39) 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>40) 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>	

- 41) Finally select **multiply** function (a) and click on the yellow triangle in front of **OrderPrice** (b).

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



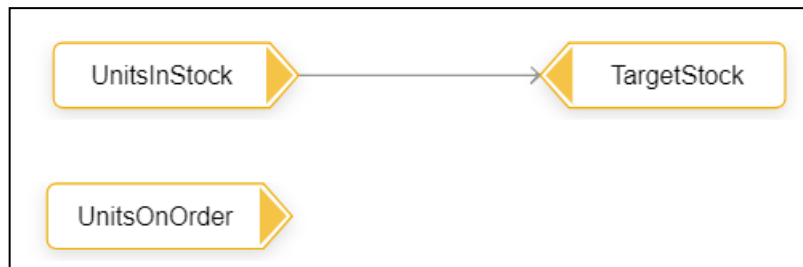
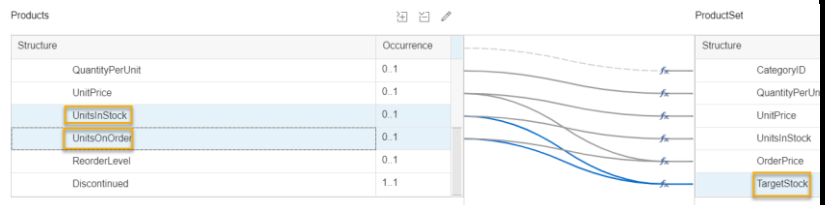
- 42) Now let's calculate the value for **TargetStock** using **UnitsInStock** and **UnitsOnOrder** elements of the source structure.

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

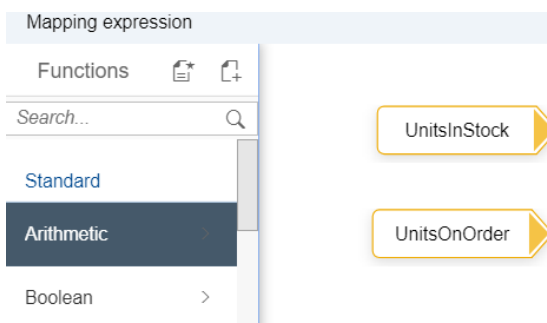
Second, drag & drop **UnitsOnOrder** of the source message to **TargetStock**.

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.

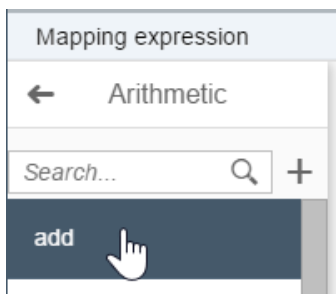


- 43) From the list of **Functions** click on **Arithmetic**.




- 44) The list of arithmetical mapping functions appears.

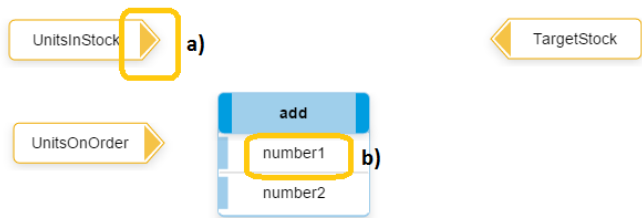
Click on **add**.



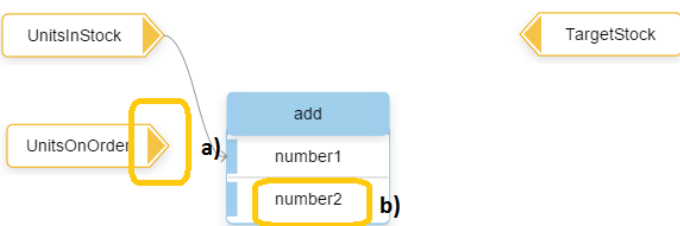
45) The **add** function is added to the canvas of the *Mapping Expression Editor*. Select the same and move it in the middle between all three elements.



46) 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).




47) 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).



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

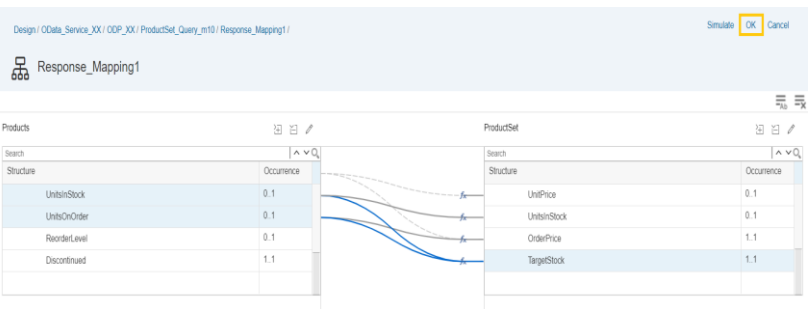
With that the mapping for calculating **TargetStock** out of the source message is implemented.

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



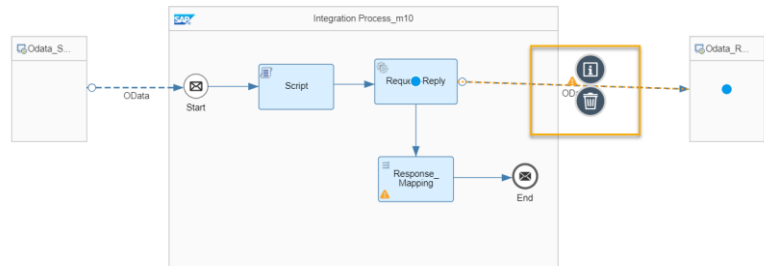
49) Click on **Ok** in the top right corner for saving your mapping.

You will be directed to the preconfigured integration flow.

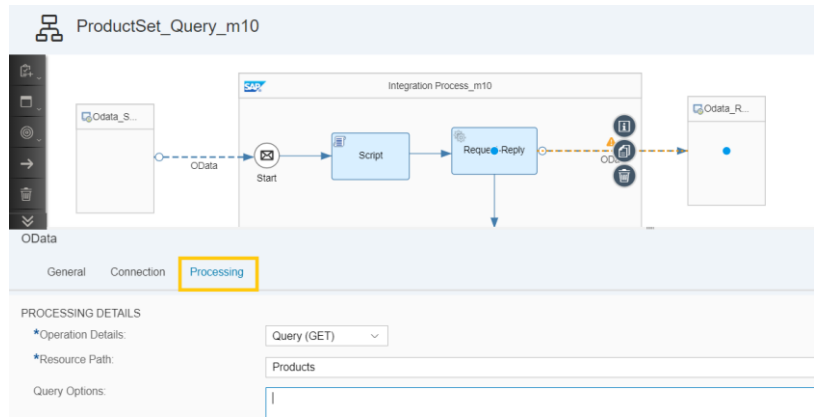


50) Next, you configure the OData channel for pulling the needed order data out of the existing OData source.

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.



51) Switch to **Processing** tab.



52) Modify the query options of the OData channel for defining which data to query from the OData source:

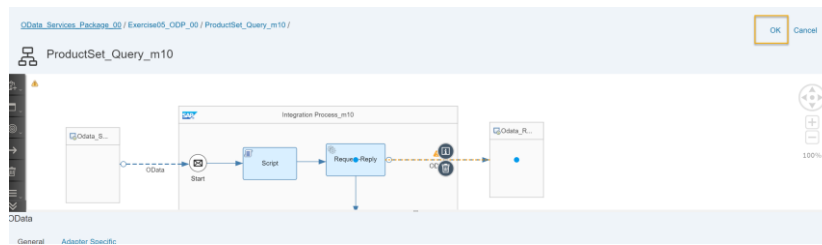
Specify the value of **Query Options** as follows:

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

Leave the other values as it is.

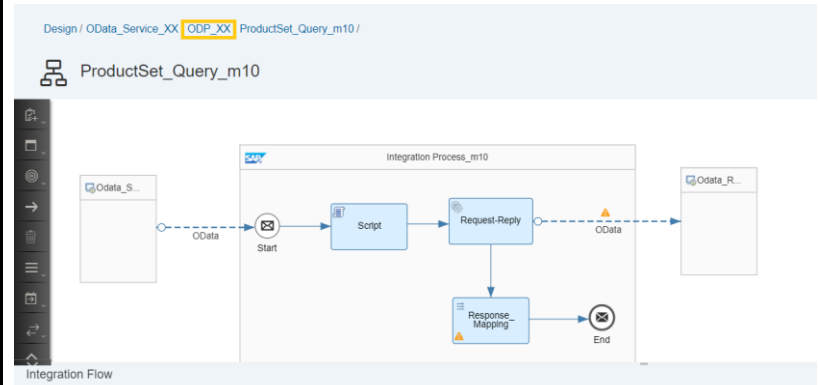
The screenshot shows the 'ProductSet_Query_m10' configuration window with the 'Query Options' field populated with the following value: **Products?\$select=ProductID,ProductName,SupplierID,CategoryID,QuantityPerUnit,UnitPrice,UnitsInStock,UnitsOnOrder**. The other fields remain the same as in the previous screenshot.

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



54) Click on Navigation Path
“**ODP_XX**” to go back to main
integration scenario screen

Note: Here **XX** is your S/P/I/D
User ID



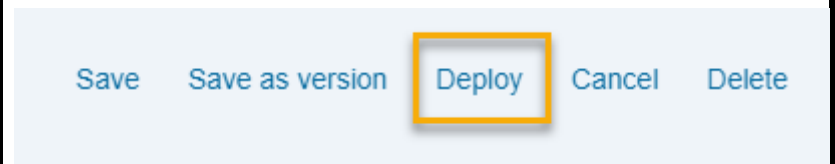
55) Click on **Save** to save the
changes.

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

**Deploy Integration Project on
tenant:**

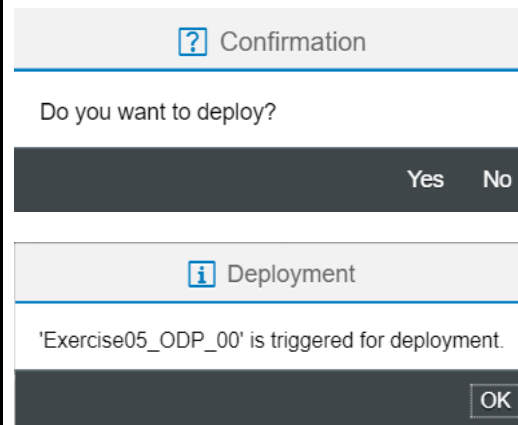
Follow this step to deploy Integration Project on tenant.

56) Press **Deploy** in the OData
Service Task Bar.




57) You will receive a confirmation.
Click on **Yes**.

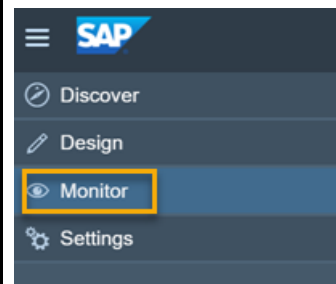
Once the deployment is
completed successfully you will
receive a 2nd notification.

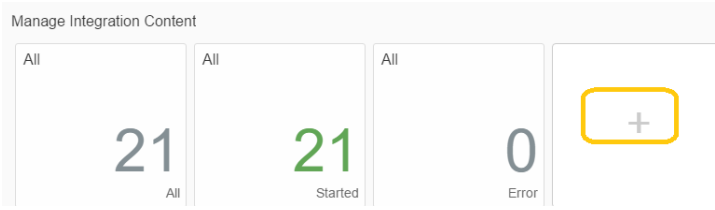
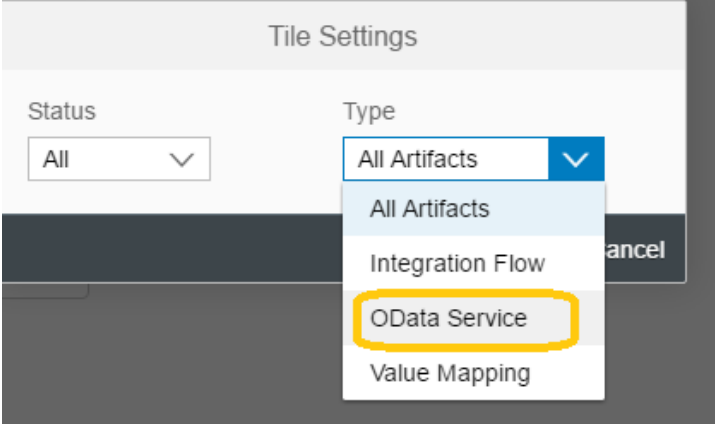
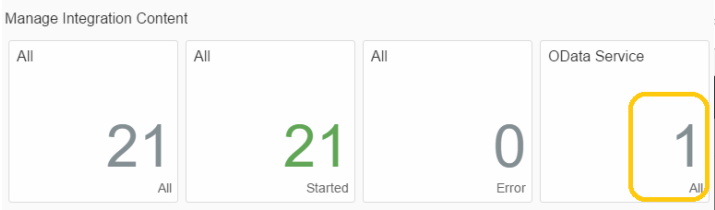
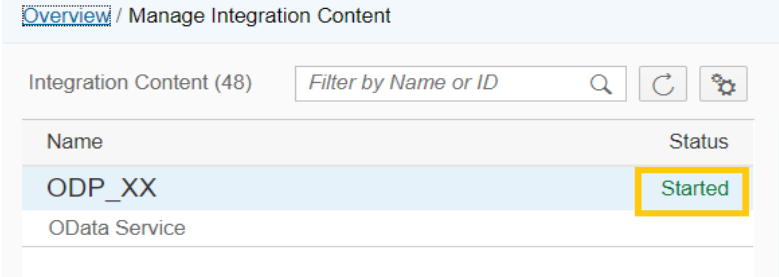
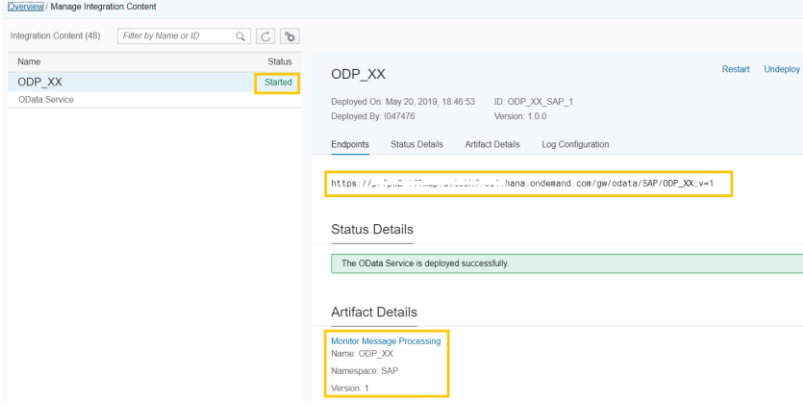



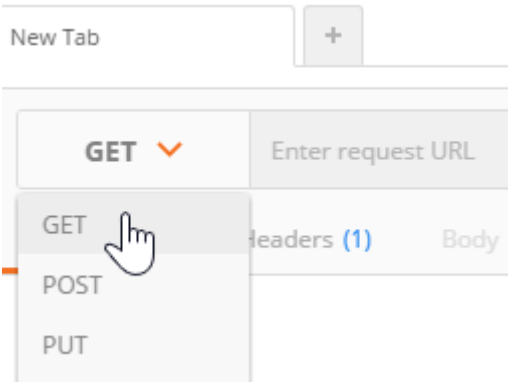

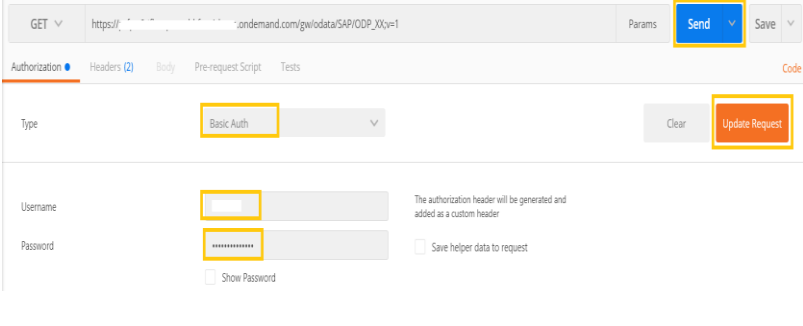

58) Verify if the deployment is
successful:

Go to the Monitoring of CPI to
check the status of the deployed

OData service: Click on  to
navigate to the **Monitor** section.



<p>59) 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.</p>	
<p>60) The Tile Settings configuration will show up. Expand the Type list and choose OData Service from the dropdown menu. Click on OK.</p> <p>With that you created an extra tile for monitoring OData services.</p>	
<p>61) Next, click on the tile OData Service (All).</p>	
<p>62) Your OData service should be listed there.</p>	
<p>63) Make sure that the status of the service is Started, you can see the Endpoint and that Artifact Details such as Name, Namespace and Version 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 refresh (🔄) icon at the top to check if the status changed in the meantime.</p> <p>Copy this Endpoint URL</p>	

Execute end to end scenario:	Follow steps to execute end to end scenario.
<p>64) Finally test the service using a test tool for OData services. For this hands-on we will use Postman. You will have to add this Postman app to your Chrome browser.</p>	
<p>65) First let's make sure that the OData service was created successfully.</p> <p>Select GET as operation option in Postman</p>	
<p>66) Paste the Endpoint URL here that you got from the Manage Integration Content, pointing to your OData Service.</p>	
<p>67) For Authorization choose Type as Basic Auth from the dropdown list.</p> <p>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</p>	
<p>68) If you get a response which shows that a collection ProductSet is available, then the OData service is registered on the CPI tenant.</p>	

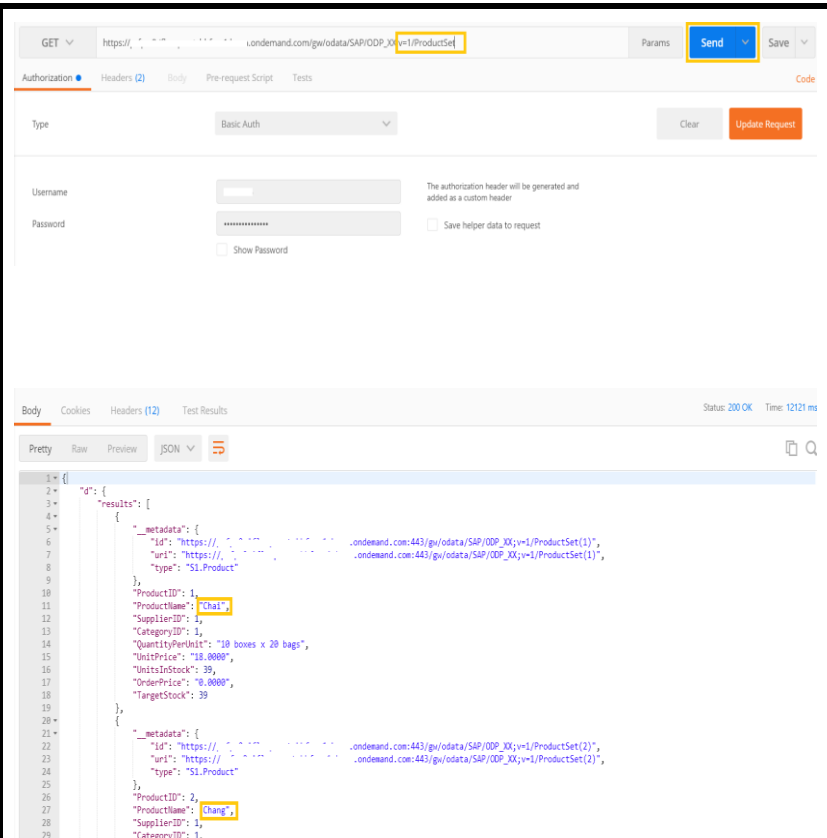
69) Next, call GET operation of this OData service for retrieving the list of orders out of the OData source.

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 exercise successfully to expose an OData API from an OData data source via SAP Cloud Platform Integration !!



**NOTE: THE FOLLOWING PAGE MAY NOT BE DELETED!
THIS IS A LEGAL STATEMENT.**

Coding Samples

Any software coding or code lines/strings ("Code") provided in this documentation are only examples and are not intended for use in a production system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules for certain SAP coding. SAP does not warrant the correctness or completeness of the Code provided herein and SAP shall not be liable for errors or damages caused by use of the Code, except where such damages were caused by SAP with intent or with gross negligence.

www.sap.com/contactsap

© 2019 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 or an SAP affiliate company.

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 or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company 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.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/copyright for additional trademark information and notices.