

Purchase Order: Create OData Service in the SAP NetWeaver Gateway System for CRUDQ Operations



Applies to:

Duet Enterprise 2.0 SP01

Summary

This guide describes in detail how to create and test OData service for CRUDQ operations in the SAP NetWeaver Gateway system for Purchase Order scenario. This guide explains the creation of the service using SAP NetWeaver Gateway Service Builder tool and the RFCs for Purchase Order.

Author: Monisha Victor

Company: SAP Labs India

Created on: 12 February 2013

Author Bio



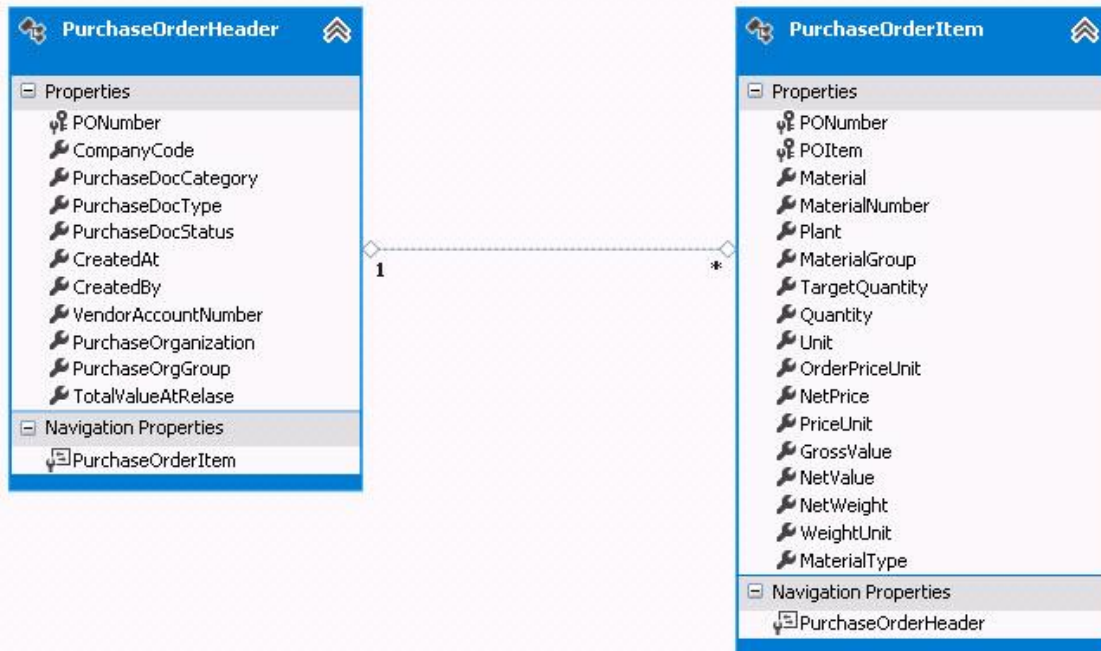
Monisha Victor is working as a Developer with the Duet Enterprise 2.0 team at SAP Labs India.

Table of Contents

1.	Introduction	3
2.	Prerequisites	3
3.	Procedure	4
3.1	Creating SAP NetWeaver Gateway Project	4
3.2	Import Data Model	6
3.3	Map Data Source for the Query, Read, Update and Delete Operations	8
3.3.1	Query Operation	8
3.3.2	Read operation	15
3.3.3	Update operation	19
3.3.4	Delete operation	22
3.4	Generate OData service	24
3.5	Modifying the Metadata Provider Class	27
3.6	Implementing Data Provider Class	29
3.7	Register Service in SAP NetWeaver Gateway system	30
4	Test OData Service	33
4.1	Service Document	33
4.2	Metadata	35
4.3	SharePoint Purchase Order Application	36
4.4	Create (Deep Insert) operation	38
4.4.1	Execute the Create operation	38
4.5	Query operation	45
4.5.1	Read all Purchase Order Header entries	46
4.5.2	Read all Purchase Order Item entries	46
4.6	Read operation	46
4.6.1	Read specific Purchase Order Header entry	46
4.6.2	Read specific Purchase Order Item entry	47
4.7	Update operation	47
4.7.1	Purchase Order Header Update	47
4.7.2	Purchase Order Item update	48
4.8	Delete operation	49
4.8.1	Purchase Order Item Delete	49
4.8.2	Purchase Order Header Delete	50
5.	Glossary	51
	Related Content	52
	Copyright	53

1. Introduction

This guide describes the Purchase Order scenario for CRUDQ calls. The sample scenario used in this guide has two entities: purchase order header and purchase order item. The following diagram shows the Entity data model used in this guide:



The OData service is generated using the SAP NetWeaver Gateway Service Builder in the SAP NetWeaver Gateway system. Using this, the metadata and data provider classes, the model and the service is generated. In the OData Channel based development, the entities of the data model are defined in the metadata provider class. The run time behavior of the OData service is handled by the data provider class that is assigned to the object models. The methods in the metadata and data provider extension classes can be redefined according to the requirements.

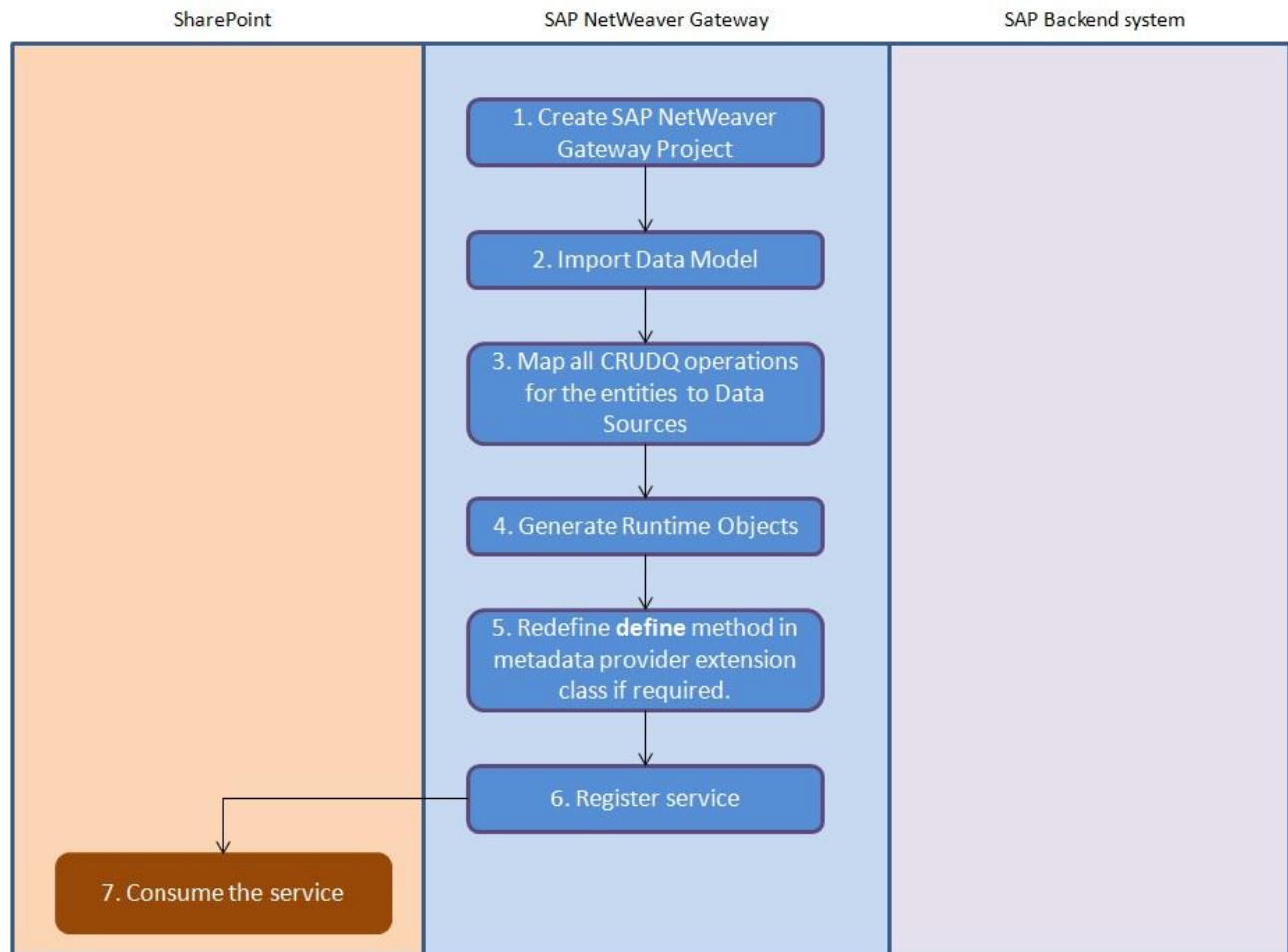
This document describes in detail the procedure to create the OData service and also steps to test it.

2. Prerequisites

1. SAP NetWeaver Gateway 2.0 SP5 (or higher) installed and configured. For more information, refer to [How to Install and Configure Duet Enterprise 2.0](#) guide.
2. IW_BEP software component installed in the SAP NetWeaver Gateway system.
3. SAP backend system is connected to the SAP NetWeaver Gateway system.
4. Purchase Order scenario should be configured in the SAP system and it should be possible to create Purchase Order from the transaction me21n.
5. Minimum knowledge of ABAP Object Oriented programming is required.

3. Procedure

The following diagram gives an overview of the steps to be performed in the creation of the OData service:



3.1 Creating SAP NetWeaver Gateway Project

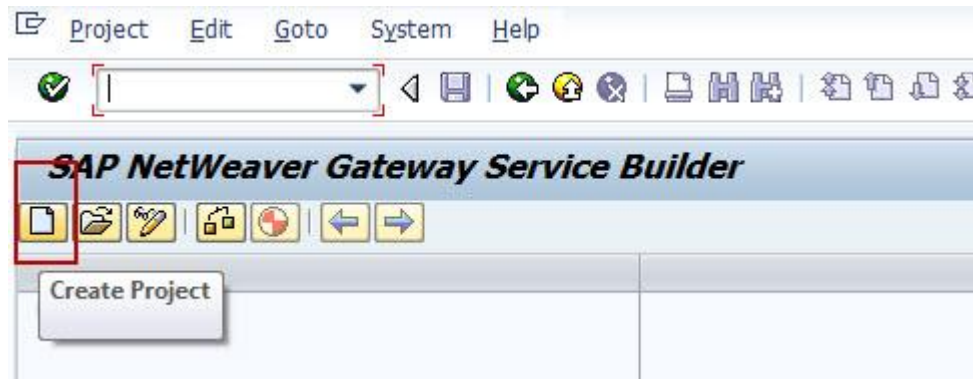
In this step, we will be creating a new SAP NetWeaver Gateway project to build the OData service.

Ensure the following authorization object required to create a -project is assigned to the user in the SAP NetWeaver Gateway system:

Authorization Object	Authorization Field	Value
/IWBEP/SB	/IWBEP/PRJ	*
	ACTVT	01, 02, 03, 07, 60
	DEVCLASS	\$tmp

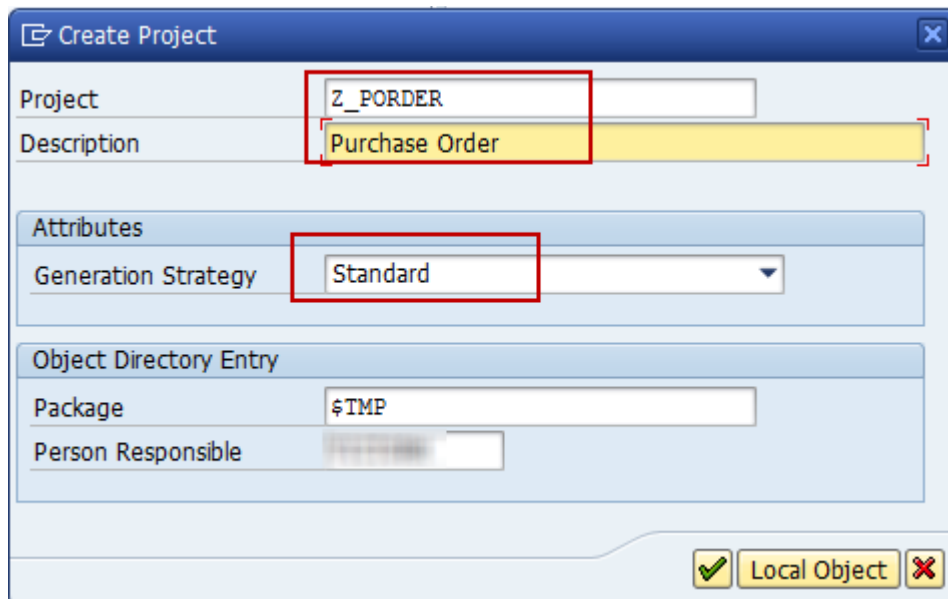
Refer to [Duet Enterprise 2.0 Security Guide](#) for more information on how to create a role and assign to users.

1. In the SAP system where IW_BEP component is installed, go to transaction **segu**.
2. Create a new project by clicking the **Create Project** option.

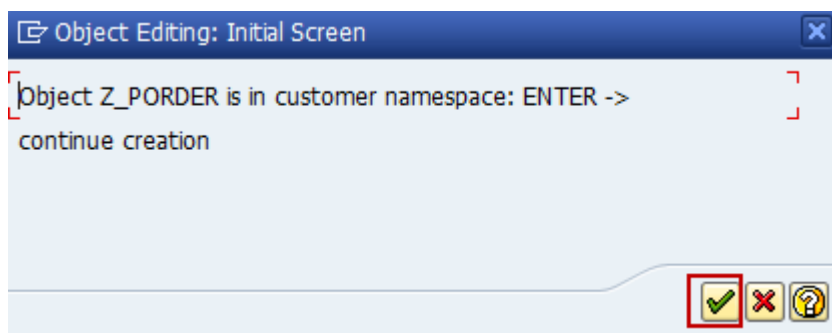


3. Enter the following details and click on **Local Object**:

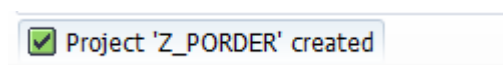
- Project: Z_PORDER
- Description: Purchase Order
- Generation Strategy: Standard



4. Click on the tick mark in the next screen.

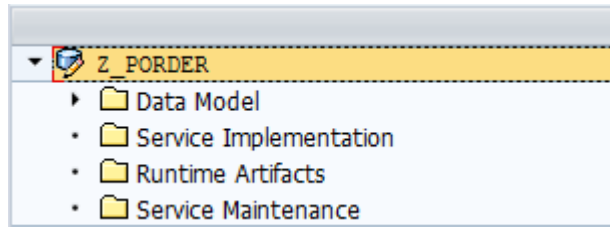


5. The following success message is displayed.



6. Click on Save.

7. The project created will contain the folders as shown below:

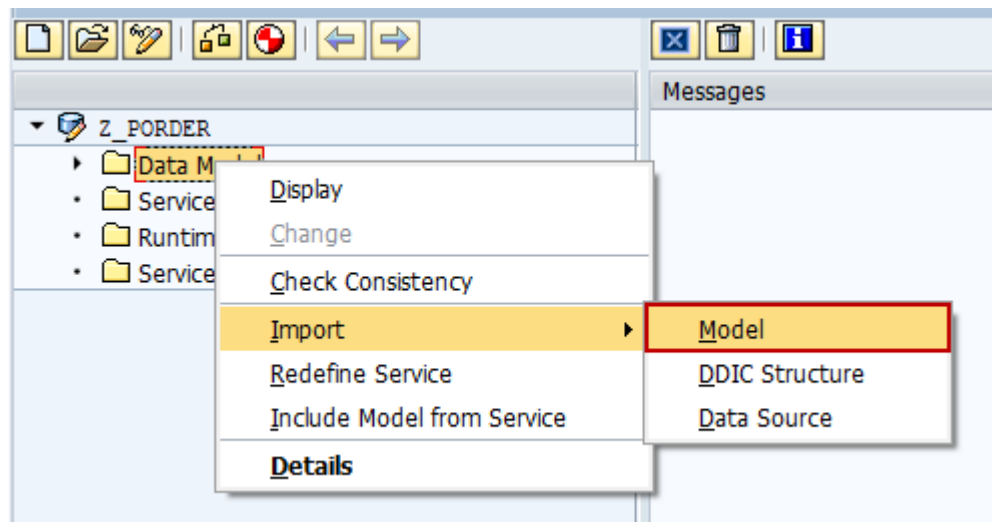


3.2 Import Data Model

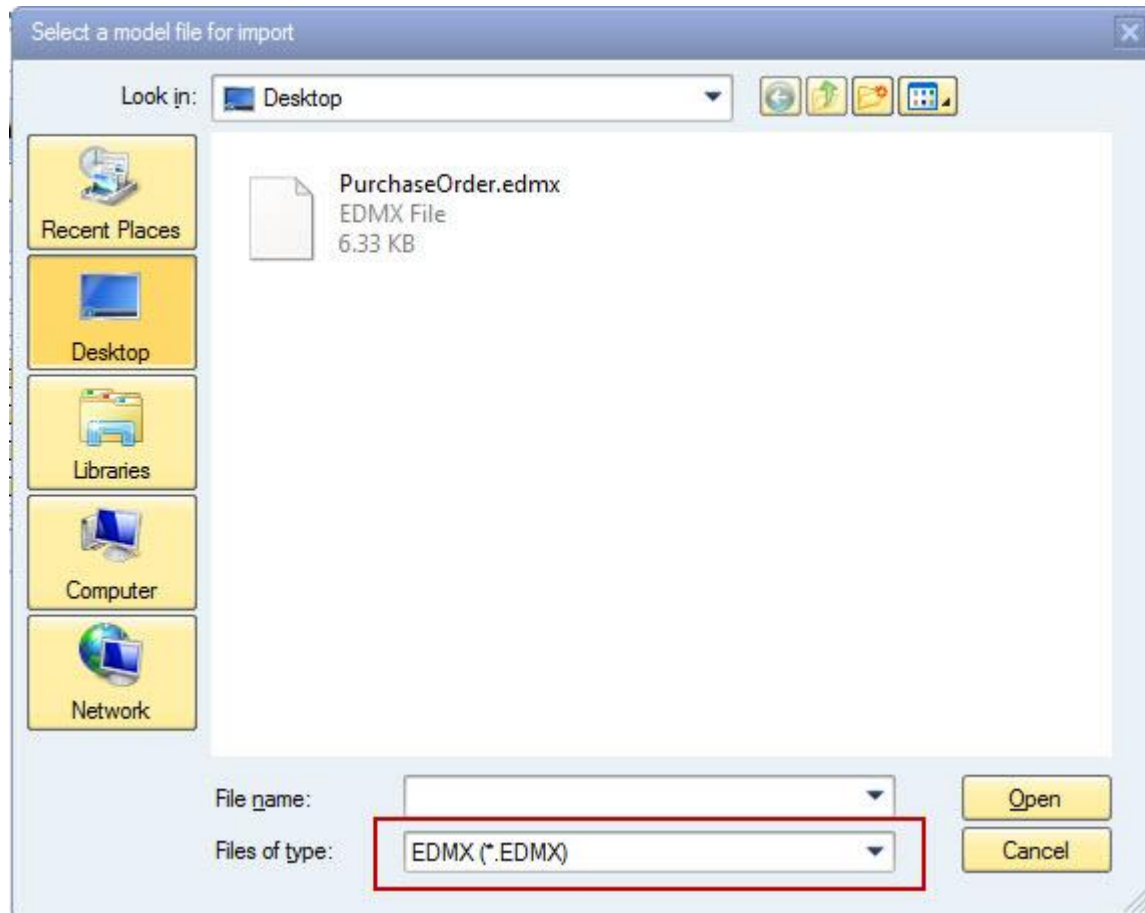
This step involves importing the edmx data model and creating the data model. An edmx file is an xml file that defines a data model and is used as a common interface between the SharePoint developer and the SAP developer. This data model was created using Visual Studio.

Refer to the model file [POrder.txt](#) that is used as a sample in this guide. Copy the contents into a file and save it as a .edmx file.

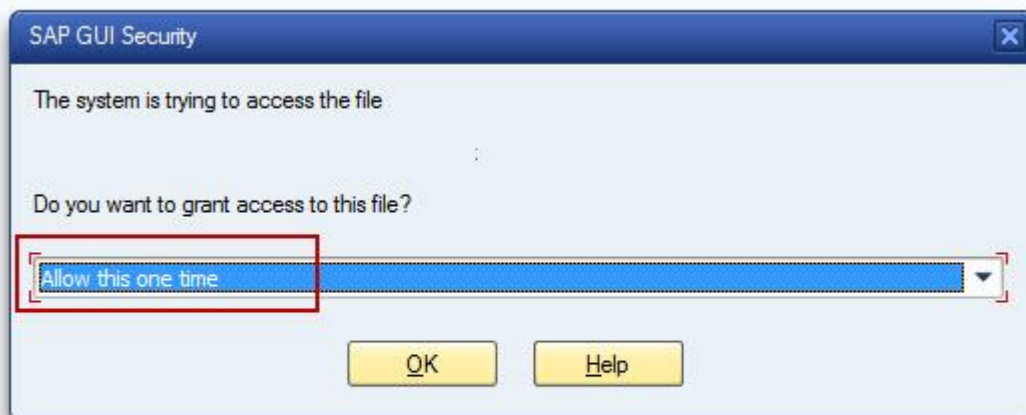
1. From the project folder in the service builder transaction **sebw**, right click on the Data Model folder and click on **Import-> Data model**. Browse for the edmx file and import the file.



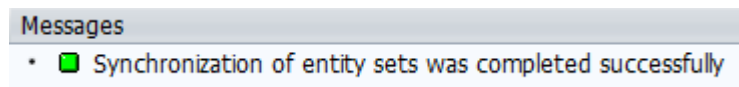
2. In the browse screen, choose EDMX as the files of type and browse for the edmx file and click on Open.



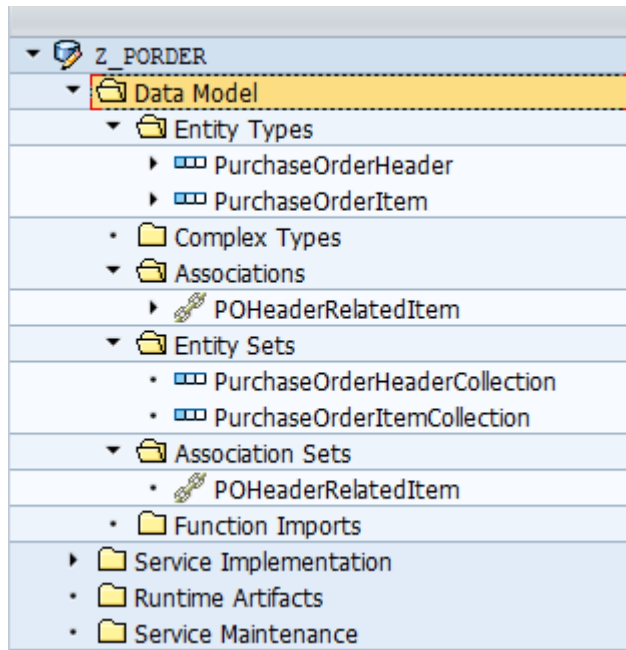
3. In the SAP GUI Security screen, choose **Allow this one time** and click on OK.



4. The following success message is displayed.



5. The entities and their properties can be viewed from the Data Model folder.



Name	K...	Edm Core Type	Pre...	Sc...	Ma...	Unit Property Name
CompanyCode	<input type="checkbox"/>	Edm.String	0	0	4	
CreatedAt	<input type="checkbox"/>	Edm.DateTime	0	0	0	
CreatedBy	<input type="checkbox"/>	Edm.String	0	0	12	
PONumber	<input checked="" type="checkbox"/>	Edm.String	0	0	10	
PurchaseDocCategory	<input type="checkbox"/>	Edm.String	0	0	1	
PurchaseDocStatus	<input type="checkbox"/>	Edm.String	0	0	1	
PurchaseDocType	<input type="checkbox"/>	Edm.String	0	0	4	
PurchaseOrganization	<input type="checkbox"/>	Edm.String	0	0	4	
PurchaseOrgGroup	<input type="checkbox"/>	Edm.String	0	0	3	
TotalValueAtRelease	<input type="checkbox"/>	Edm.Decimal	15	2	0	
VendorAccountNumber	<input type="checkbox"/>	Edm.String	0	0	10	

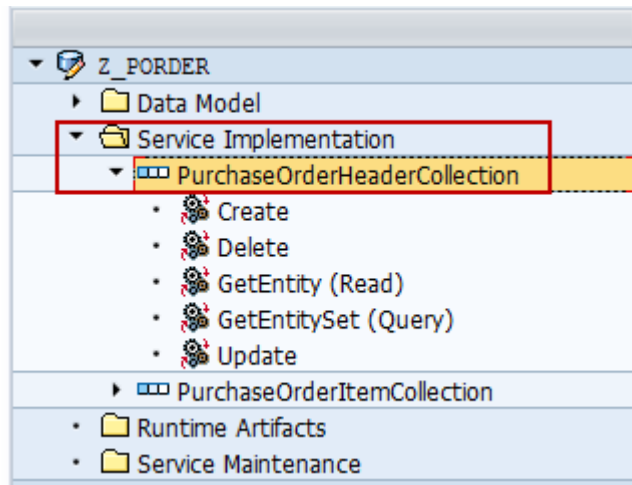
3.3 Map Data Source for the Query, Read, Update and Delete Operations

3.3.1 Query Operation

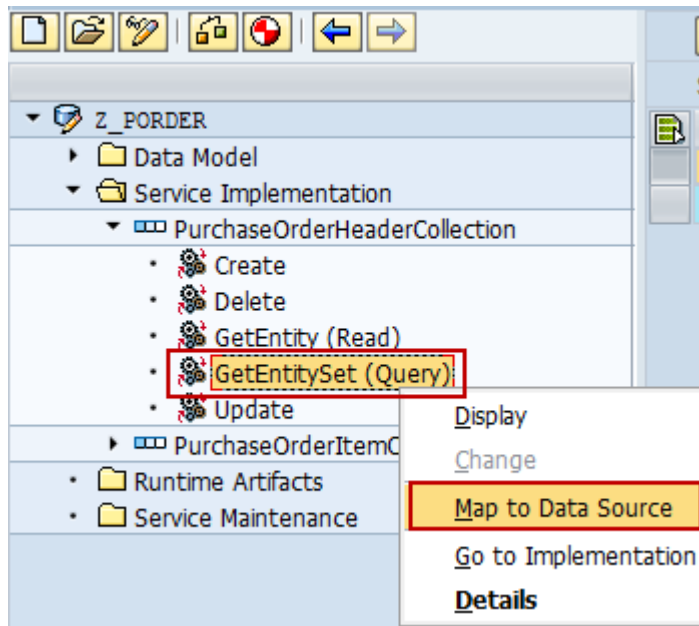
In this section, we shall see how to map the data source to the Query Operation for Purchase Order Header and Purchase Order Item.

Purchase Order Header:

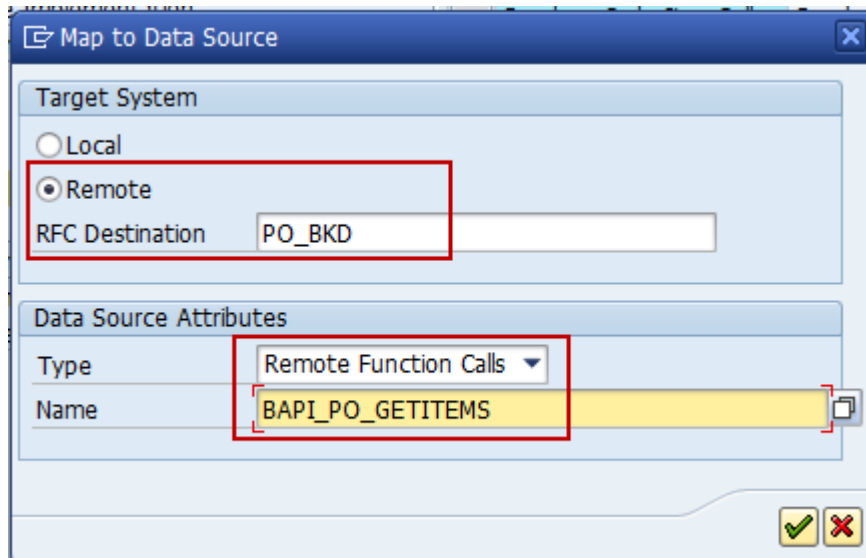
1. Expand the node **Service Implementation-> PurchaseOrderHeaderCollection**



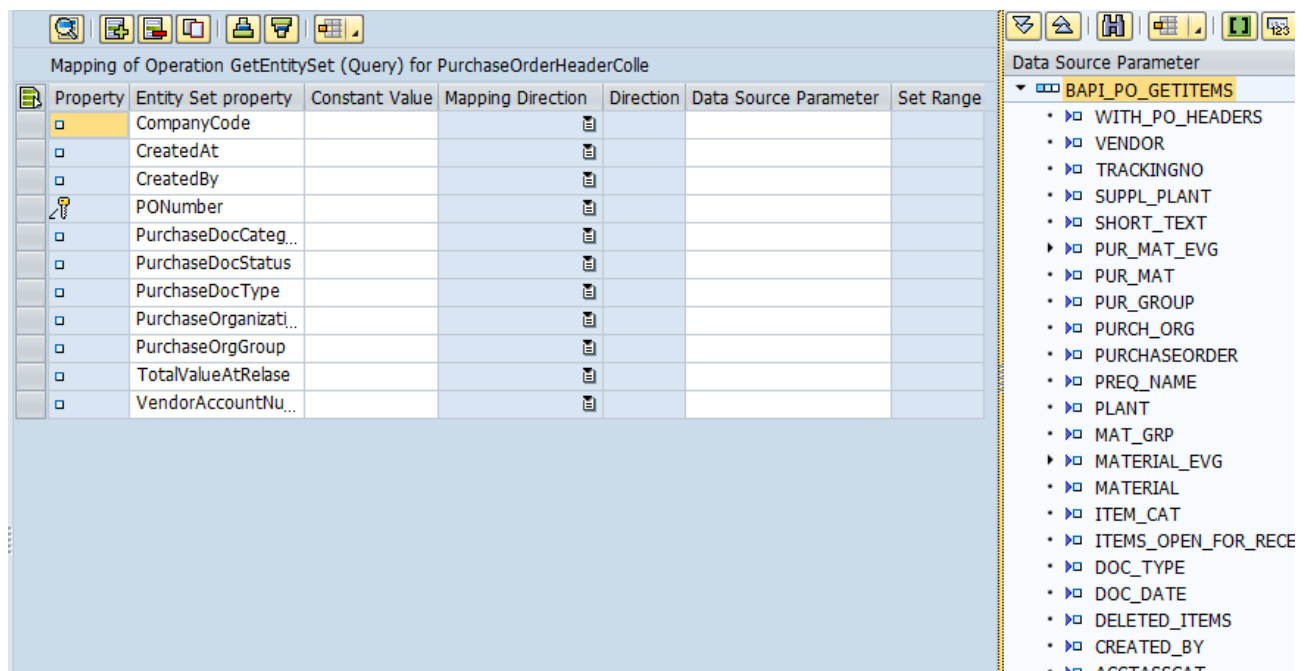
2. Right click on **GetEntitySet (Query)** and click on **Map to Data Source**.



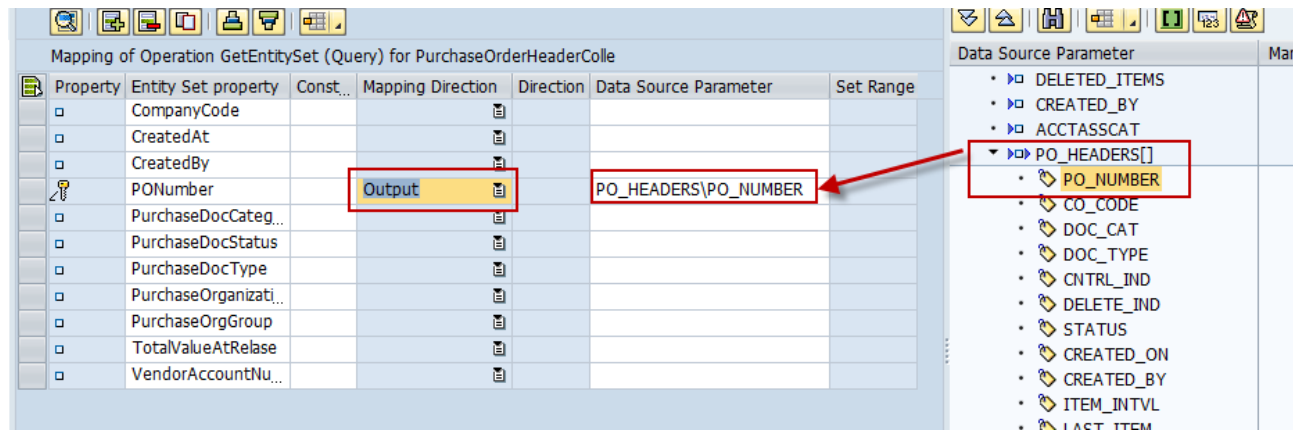
3. Select **Remote** under the Target System if the SAP system where Purchase Order is configured a different system. Enter the rfc destination of the SAP System in the RFC Destination field. Select the type to be **Remote Function Calls** and the **Name** as **BAPI_PO_GETITEMS**.



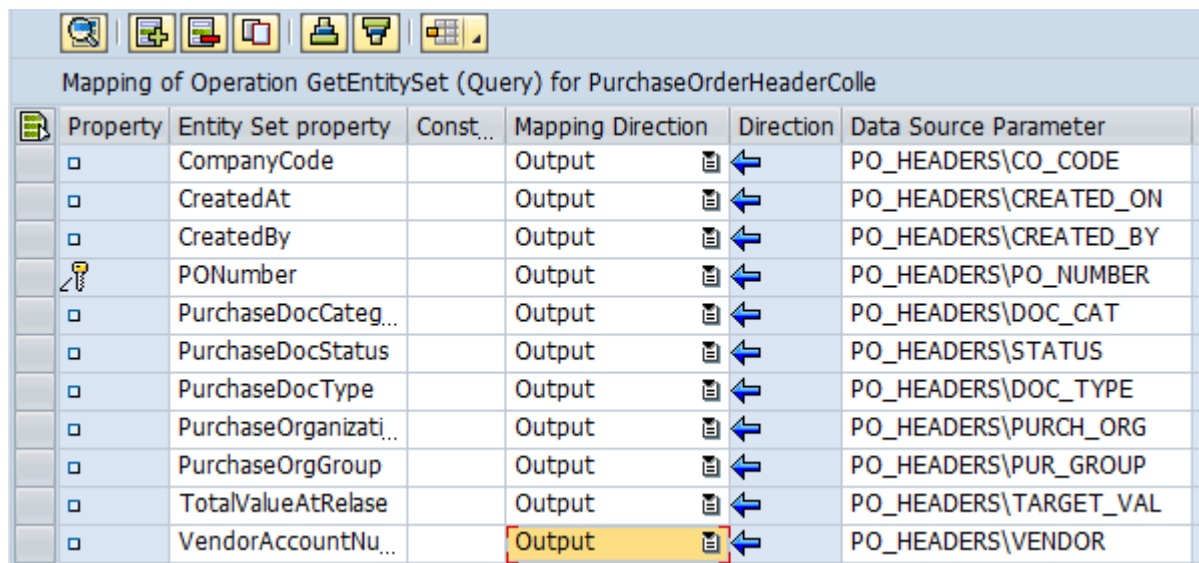
4. The following screen appears after clicking on the tick mark.



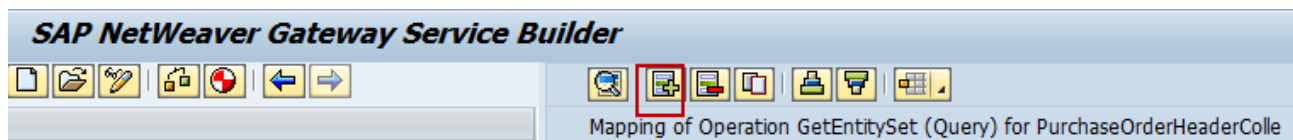
5. After this, from the **Data Source Parameter** tab on the right hand side, drag and drop the output parameters to the **Mapping of Operation** screen on the left hand side. The Data Source Parameter can also be selected using the F4 help. Ensure the **Mapping Direction** is selected as Input or Output for all the mappings based on the import or export parameter of the Function Module.



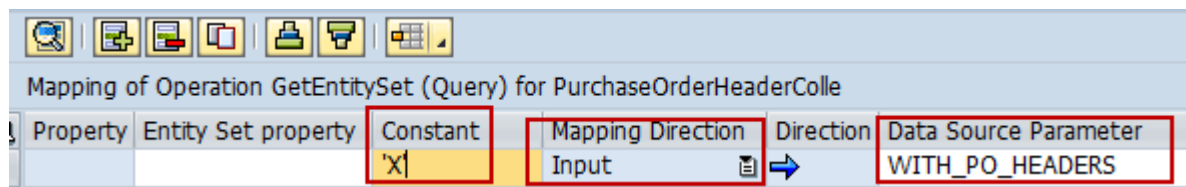
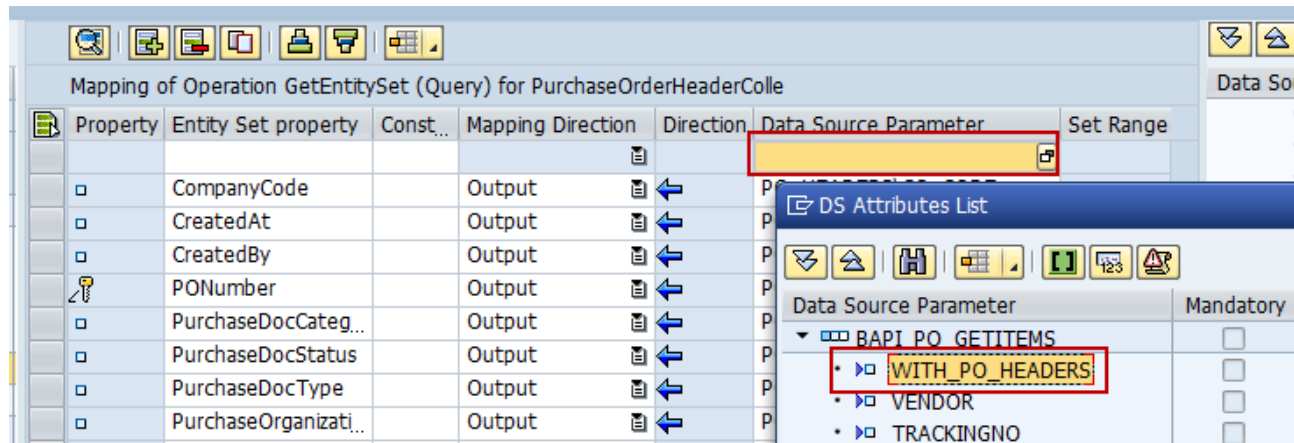
6. This way, we map the other parameters as well from the Data Source Parameter tab to the entity set property. The mapping looks as below.



7. To indicate any Input parameter to the function module, click on the **Insert Row** option.



8. An empty row appears in the **Mapping of operation** screen.
9. Using the **F4 help**, double click on the required parameter for the **Data Source Parameter**. Choose the Mapping Direction as **Input** from the drop down option. And enter the required value in the Constant column as shown below.



10. Click on **Save**.

11. The table below gives the mapping required for the Purchase Order Header Query operation.

Entity Set Property	Constant Value	Mapping Direction	Data Source Parameter
	'NB'	Input	DOC_TYPE
	'X'	Input	WITH_PO_HEADERS
CompanyCode		Output	PO_HEADERS\CO_CODE
CreatedAt		Output	PO_HEADERS\CREATED_ON
CreatedBy		Output	PO_HEADERS\CREATED_BY
PONumber		Output	PO_HEADERS\PO_NUMBER
PurchaseDocCategory		Output	PO_HEADERS\DOC_CAT
PurchaseDocStatus		Output	PO_HEADERS\STATUS
PurchaseDocType		Output	PO_HEADERS\DOC_TYPE
PurchaseOrganization		Output	PO_HEADERS\PURCH_ORG
PurchaseOrgGroup		Output	PO_HEADERS\PUR_GROUP
TotalValueAtRelease		Output	PO_HEADERS\TARGET_VAL
VendorAccountNumber		Output	PO_HEADERS\VENDOR

12. The below screenshot shows the final mapping of Data source parameter to the entity set property for the Purchase Order Header Query operation.

Property	Entity Set property	Constant Value	Mapping Direction	Direction	Data Source Parameter	Set Range
		'NB'	Input	➡	DOC_TYPE	
		'X'	Input	➡	WITH_PO_HEADERS	
□	CompanyCode		Output	⬅	PO_HEADERS\CO_CO...	
□	CreatedAt		Output	⬅	PO_HEADERS\CREATE...	
□	CreatedBy		Output	⬅	PO_HEADERS\CREATE...	
🔑	PONumber		Output	⬅	PO_HEADERS\PO_NU...	
□	PurchaseDocCateg...		Output	⬅	PO_HEADERS\DOC_C...	
□	PurchaseDocStatus		Output	⬅	PO_HEADERS\STATUS	
□	PurchaseDocType		Output	⬅	PO_HEADERS\DOC_T...	
□	PurchaseOrganizati...		Output	⬅	PO_HEADERS\PURCH...	
□	PurchaseOrgGroup		Output	⬅	PO_HEADERS\PUR_GR...	
□	TotalValueAtRelease		Output	⬅	PO_HEADERS\TARGE...	
□	VendorAccountNu...		Output	⬅	PO_HEADERS\VENDOR	

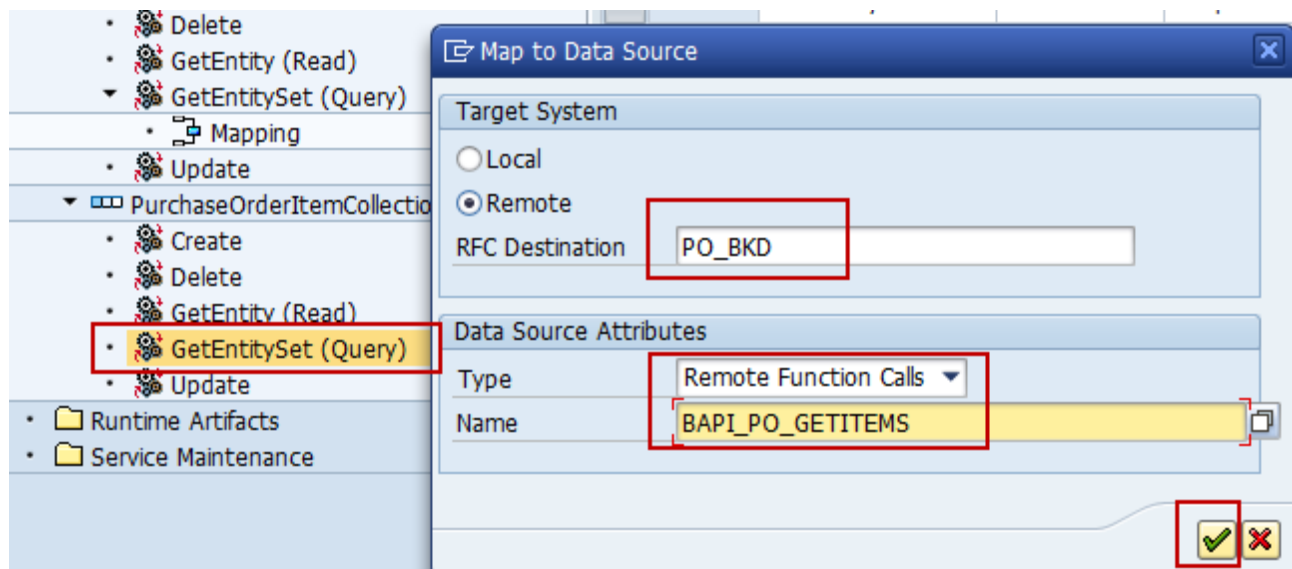
This completes the steps to map the data source parameters and the property of the entity set.

Purchase Order Item:

1. Similarly for Purchase Order Item, expand **Service Implementation -> PurchaseOrderItemCollection**. Right click on **GetEntitySet (Query)** option, select **Map to Data Source** and enter the required fields as below and click on the tick mark.

Type: Remote Function Calls
























Name: BAPI_PO_GETITEMS



2. The table below gives the mapping required for the Purchase Order Item Query operation.

Entity Set Property	Constant Value	Mapping Direction	Data Source Parameter
	'NB'	Input	DOC_TYPE
GrossValue			
Material		Output	PO_ITEMS\MATERIAL
MaterialGroup		Output	PO_ITEMS\MAT_GRP
MaterialNumber		Output	PO_ITEMS\PUR_MAT
MaterialType			
NetPrice		Output	PO_ITEMS\NET_PRICE
NetValue			
NetWeight			
OrderPriceUnit		Output	PO_ITEMS\ORDERPR_UN
Plant		Output	PO_ITEMS\PLANT
POItem		Output	PO_ITEMS\PO_ITEM
PONumber		Output	PO_ITEMS\PO_NUMBER
PONumber		Input	PURCHASEORDER
PriceUnit		Output	PO_ITEMS\PRICE_UNIT
Quantity		Output	PO_ITEMS\DISP_QUAN
TargetQuantity			
Unit		Output	PO_ITEMS\UNIT
WeightUnit			

3. The final mapping for the Purchase Order Item Query operation looks as shown below.

Mapping of Operation GetEntitySet (Query) for PurchaseOrderItemCollect							
	Property	Entity Set property	Constant Value	Mapping Direction	Direction	Data Source Parameter	Set Range
			'NB'	Input		DOC_TYPE	
<input type="checkbox"/>	GrossValue						
<input type="checkbox"/>	Material			Output		PO_ITEMS\MATERIAL	
<input type="checkbox"/>	MaterialGroup			Output		PO_ITEMS\MAT_GRP	
<input type="checkbox"/>	MaterialNumber			Output		PO_ITEMS\PUR_MAT	
<input type="checkbox"/>	MaterialType						
<input type="checkbox"/>	NetPrice			Output		PO_ITEMS\NET_PRICE	
<input type="checkbox"/>	NetValue						
<input type="checkbox"/>	NetWeight						
<input type="checkbox"/>	OrderPriceUnit			Output		PO_ITEMS\ORDERPR...	
<input type="checkbox"/>	Plant			Output		PO_ITEMS\PLANT	
	POItem			Output		PO_ITEMS\PO_ITEM	
	PONumber			Output		PO_ITEMS\PO_NUMBER	
	PONumber			Input		PURCHASEORDER	
<input type="checkbox"/>	PriceUnit			Output		PO_ITEMS\PRICE_UNIT	
<input type="checkbox"/>	Quantity			Output		PO_ITEMS\DISP_QUAN	
<input type="checkbox"/>	TargetQuantity						
<input type="checkbox"/>	Unit			Output		PO_ITEMS\UNIT	
<input type="checkbox"/>	WeightUnit						

4. Click on **Save**.

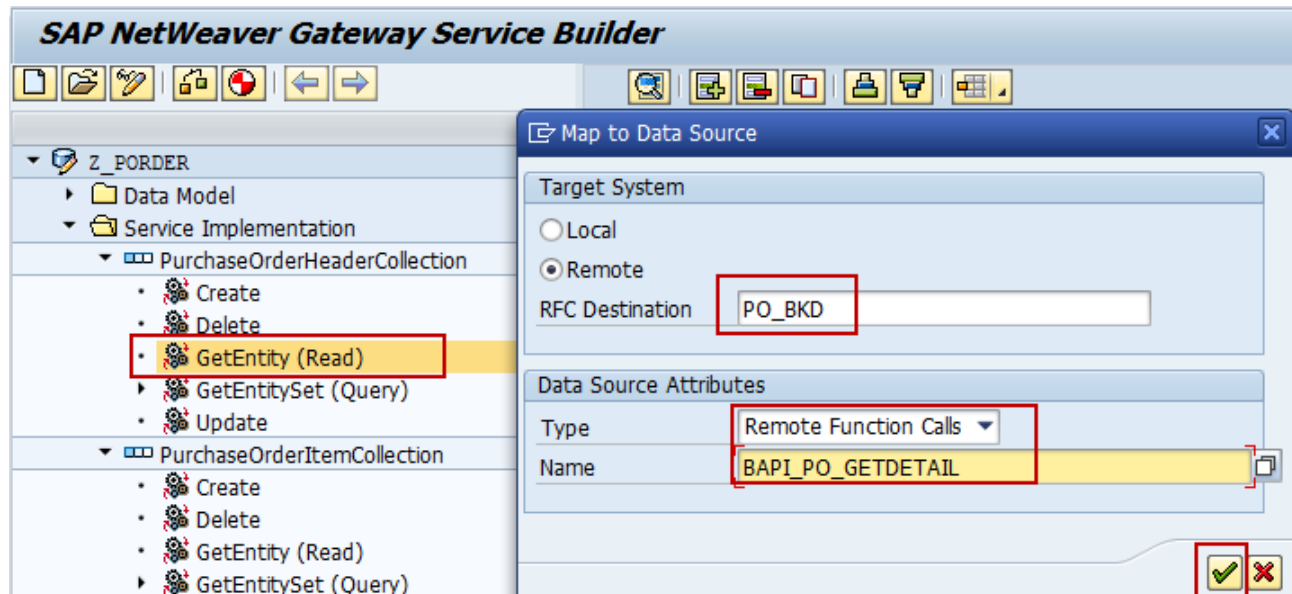
3.3.2 Read operation

Purchase Order Header

1. Expand **Service Implementation -> PurchaseOrderHeaderCollection**. Right click on **GetEntity (Read)** option, select **Map to Data Source** and enter the required fields as below and click on the tick mark.

Type: Remote Function Calls

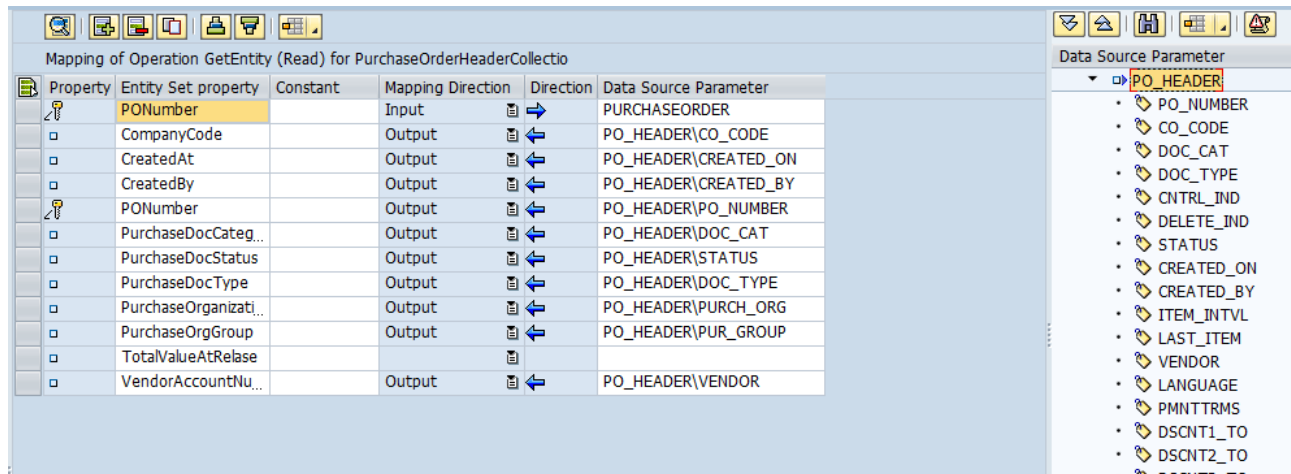
Name: BAPI_PO_GETDETAIL



2. The table below gives the mapping required for the Purchase Order Header Read operation.

Entity Set Property	Constant Value	Mapping Direction	Data Source Parameter
CompanyCode		Output	PO_HEADER\CO_CODE
CreatedAt		Output	PO_HEADER\CREATED_ON
CreatedBy		Output	PO_HEADER\CREATED_BY
PONumber		Output	PO_HEADER\PO_NUMBER
PONumber		Input	PURCHASEORDER
PurchaseDocCategory		Output	PO_HEADER\DOC_CAT
PurchaseDocStatus		Output	PO_HEADER\STATUS
PurchaseDocType		Output	PO_HEADER\DOC_TYPE
PurchaseOrganization		Output	PO_HEADER\PURCH_ORG
PurchaseOrgGroup		Output	PO_HEADER\PUR_GROUP
TotalValueAtRelease			
VendorAccountNumber		Output	PO_HEADER\VENDOR

3. Below is the screenshot showing the mapping of the data source parameter and the entity set property for the Purchase Order Header Read operation.



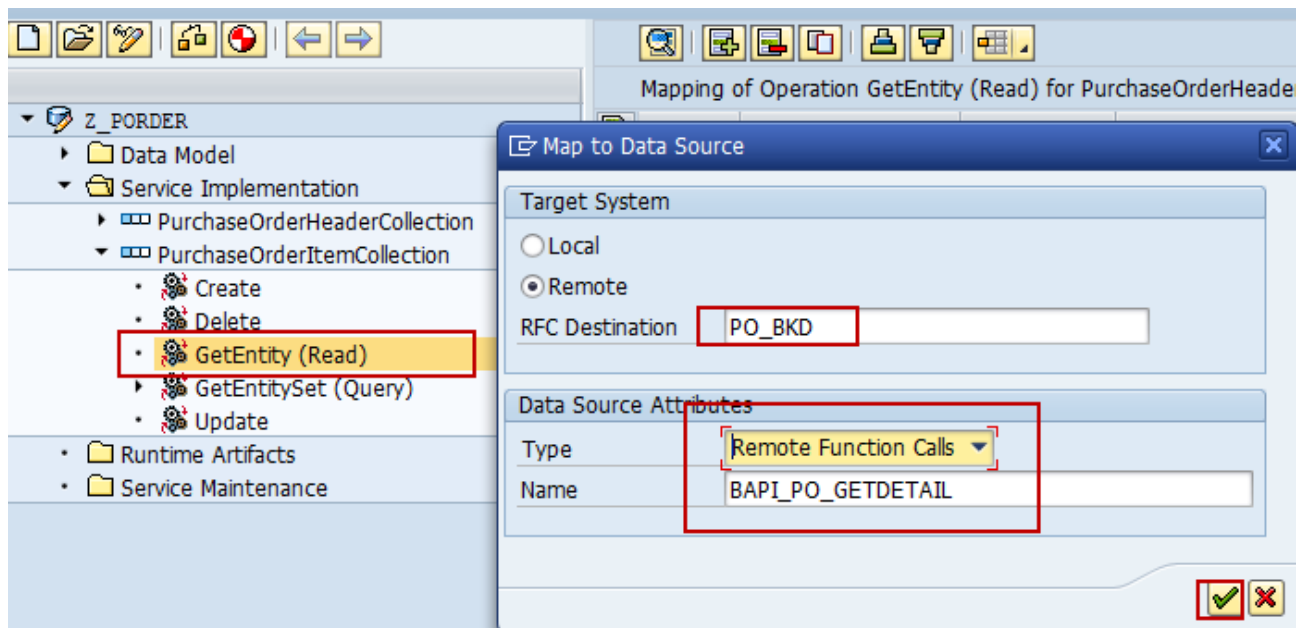
4. Click on **Save**.

Purchase Order Item

1. Expand **Service Implementation -> PurchaseOrderItemCollection**. Right click on **GetEntity (Read)** option, select **Map to Data Source** and enter the required fields as below and click on the tick mark.

Type: Remote Function Calls

Name: BAPI_PO_GETDETAIL




























2. The table below gives the mapping required for the Purchase Order Item Read operation.

Entity Set Property	Constant Value	Mapping Direction	Data Source Parameter
	X'	Input	ITEMS
GrossValue		Output	PO_ITEMS\GROS_VALUE
Material		Output	PO_ITEMS\MATERIAL

MaterialGroup		Output	PO_ITEMS\MAT_GRP
MaterialNumber		Output	PO_ITEMS\PUR_MAT
MaterialType		Output	PO_ITEMS\MAT_TYPE
NetPrice		Output	PO_ITEMS\NET_PRICE
NetValue		Output	PO_ITEMS\NET_VALUE
NetWeight		Output	PO_ITEMS\NET_WEIGHT
OrderPriceUnit		Output	PO_ITEMS\ORDERPR_UN
Plant		Output	PO_ITEMS\PLANT
POItem		Input	PO_ITEMS\PO_ITEM
POItem		Output	PO_ITEMS\PO_ITEM
PONumber		Output	PO_ITEMS\PO_NUMBER
PONumber		Input	PURCHASEORDER
PriceUnit		Output	PO_ITEMS\PRICE_UNIT
Quantity		Output	PO_ITEMS\QUANTITY
TargetQuantity		Output	PO_ITEMS\TARGET_QTY
Unit		Output	PO_ITEMS\UNIT
WeightUnit		Output	PO_ITEMS\WEIGHTUNIT

3. The screenshot below shows the final mapping of the Data Source Parameters to the entity set properties for the Purchase Order Item Read operation.

Mapping of Operation GetEntity (Read) for PurchaseOrderItemCollection						
	Property	Entity Set property	Constant	Mapping Direction	Direction	Data Source Parameter
			'X'	Input		ITEMS
<input type="checkbox"/>		GrossValue		Output		PO_ITEMS\GROS_VALUE
<input type="checkbox"/>		Material		Output		PO_ITEMS\MATERIAL
<input type="checkbox"/>		MaterialGroup		Output		PO_ITEMS\MAT_GRP
<input type="checkbox"/>		MaterialNumber		Output		PO_ITEMS\PUR_MAT
<input type="checkbox"/>		MaterialType		Output		PO_ITEMS\MAT_TYPE
<input type="checkbox"/>		NetPrice		Output		PO_ITEMS\NET_PRICE
<input type="checkbox"/>		NetValue		Output		PO_ITEMS\NET_VALUE
<input type="checkbox"/>		NetWeight		Output		PO_ITEMS\NET_WEIGHT
<input type="checkbox"/>		OrderPriceUnit		Output		PO_ITEMS\ORDERPR_UN
<input type="checkbox"/>		Plant		Output		PO_ITEMS\PLANT
	POItem			Input		PO_ITEMS\PO_ITEM
	POItem			Output		PO_ITEMS\PO_ITEM
	PONumber			Output		PO_ITEMS\PO_NUMBER
	PONumber			Input		PURCHASEORDER
<input type="checkbox"/>		PriceUnit		Output		PO_ITEMS\PRICE_UNIT
<input type="checkbox"/>		Quantity		Output		PO_ITEMS\QUANTITY
<input type="checkbox"/>		TargetQuantity		Output		PO_ITEMS\TARGET_QTY
<input type="checkbox"/>		Unit		Output		PO_ITEMS\UNIT
<input type="checkbox"/>		WeightUnit		Output		PO_ITEMS\WEIGHTUNIT

4. Click on **Save**.

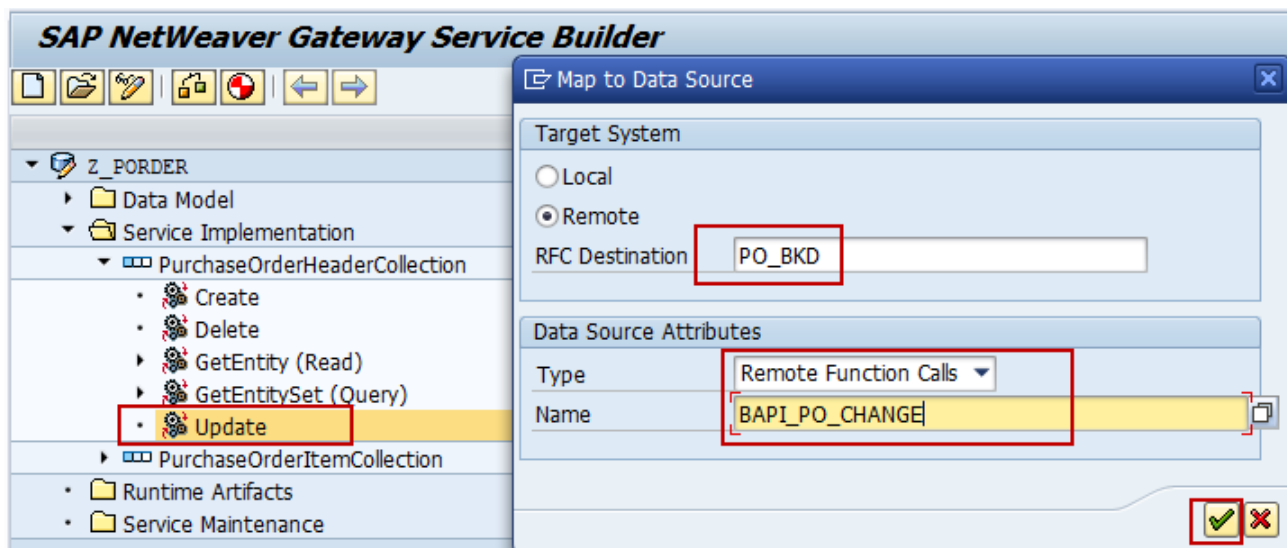
3.3.3 Update operation

Purchase Order Header

1. Expand **Service Implementation -> PurchaseOrderHeaderCollection**. Right click on **Update** option, select **Map to Data Source** and enter the required fields as below and click on the tick mark.

Type: Remote Function Calls

Name: BAPI_PO_CHANGE



2. The table below gives the mapping required for the Purchase Order Header Update operation.

Entity Set Property	Constant Value	Mapping Direction	Data Source Parameter
	X'	Input	POHEADERX\PO_NUMBER
	X'	Input	POHEADERX\PUR_GROUP
CompanyCode			
CreatedAt			
CreatedBy			
PONumber		Input	PURCHASEORDER
PurchaseDocCategory			
PurchaseDocStatus			
PurchaseDocType			
PurchaseOrganization			
PurchaseOrgGroup		Input	POHEADER\PUR_GROUP
TotalValueAtRelease			
VendorAccountNumber			

3. The screenshot below shows the final mapping of the Data Source Parameters to the entity set properties for the Purchase Order Header Update operation.

Mapping of Operation Update for PurchaseOrderHeaderCollection						
Property	Entity Set property	Constant	Mapping Direction	Direction	Data Source Parameter	
		'X'	Input	⇒	POHEADERX\PO_NUMBER	
		'X'	Input	⇒	POHEADERX\PUR_GROUP	
□	CompanyCode					
□	CreatedAt					
□	CreatedBy					
🔑	PONumber		Input	⇒	PURCHASEORDER	
□	PurchaseDocCateg...					
□	PurchaseDocStatus					
□	PurchaseDocType					
□	PurchaseOrganizati...					
□	PurchaseOrgGroup		Input	⇒	POHEADER\PUR_GROUP	
□	TotalValueAtRelase					
□	VendorAccountNu...					

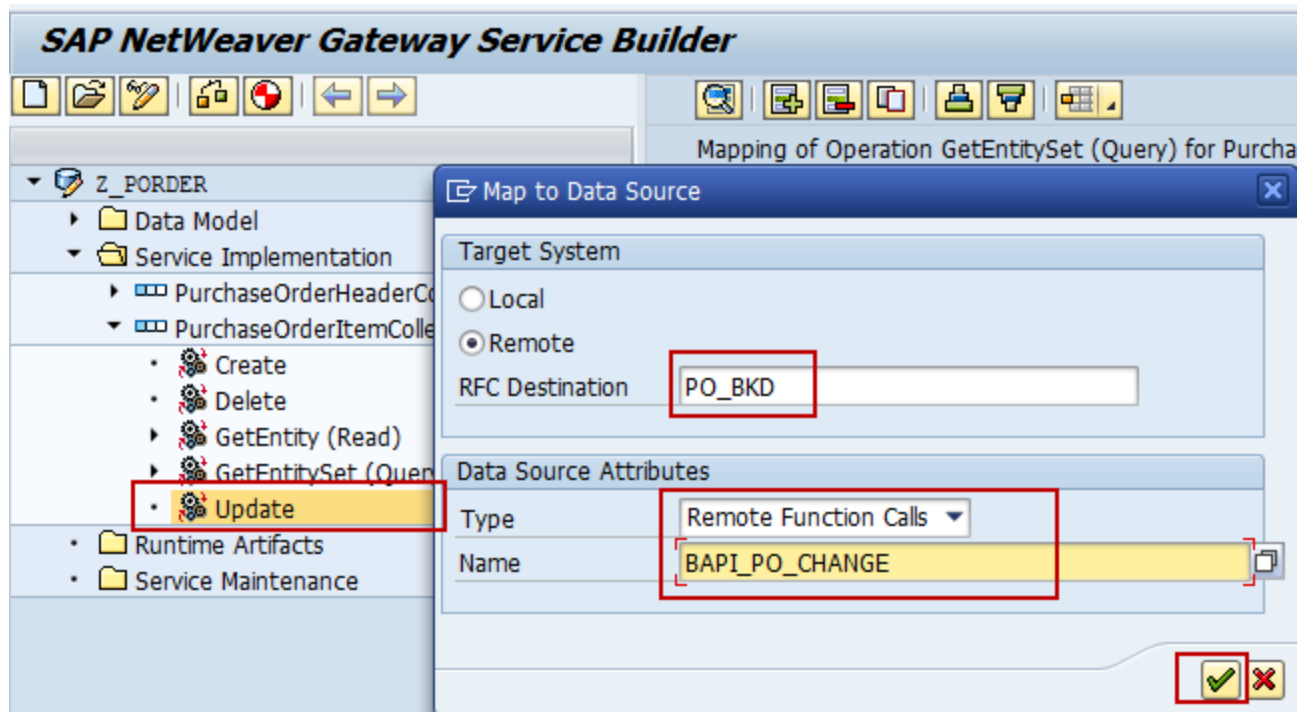
4. Click on **Save**.

Purchase Order Item

1. Expand **Service Implementation -> PurchaseOrderItemCollection**. Right click on **Update** option, select **Map to Data Source** and enter the required fields as below and click on the tick mark.

Type: Remote Function Calls





























Name: BAPI_PO_CHANGE



2. The table below gives the mapping required for the Purchase Order Item Update operation.

Entity Set Property	Constant Value	Mapping Direction	Data Source Parameter
	X'	Input	POITEMX\QUANTITY
GrossValue			
Material			
MaterialGroup			
MaterialNumber			
MaterialType			
NetPrice			
NetValue			
NetWeight			
OrderPriceUnit			
Plant			
POItem		Input	POITEM\PO_ITEM
POItem		Input	POITEMX\PO_ITEM
PONumber		Input	PURCHASEORDER
PriceUnit			
Quantity		Input	POITEM\QUANTITY
TargetQuantity			
Unit			
WeightUnit			

3. The screenshot below shows the final mapping of the Data Source Parameters to the entity set properties for the Purchase Order Item Update operation.

Mapping of Operation Update for PurchaseOrderItemCollection					
Property	Entity Set property	Constant	Mapping Direction	Direction	Data Source Parameter
	POItem		Input	 	POITEMX\PO_ITEM
		'X'	Input	 	POITEMX\QUANTITY
<input type="checkbox"/>	GrossValue				
<input type="checkbox"/>	Material				
<input type="checkbox"/>	MaterialGroup				
<input type="checkbox"/>	MaterialNumber				
<input type="checkbox"/>	MaterialType				
<input type="checkbox"/>	NetPrice				
<input type="checkbox"/>	NetValue				
<input type="checkbox"/>	NetWeight				
<input type="checkbox"/>	OrderPriceUnit				
<input type="checkbox"/>	Plant				
	POItem		Input	 	POITEM\PO_ITEM
	PONumber		Input	 	PURCHASEORDER
<input type="checkbox"/>	PriceUnit				
<input type="checkbox"/>	Quantity		Input	 	POITEM\QUANTITY
<input type="checkbox"/>	TargetQuantity				
<input type="checkbox"/>	Unit				
<input type="checkbox"/>	WeightUnit				

4. Click on **Save**.

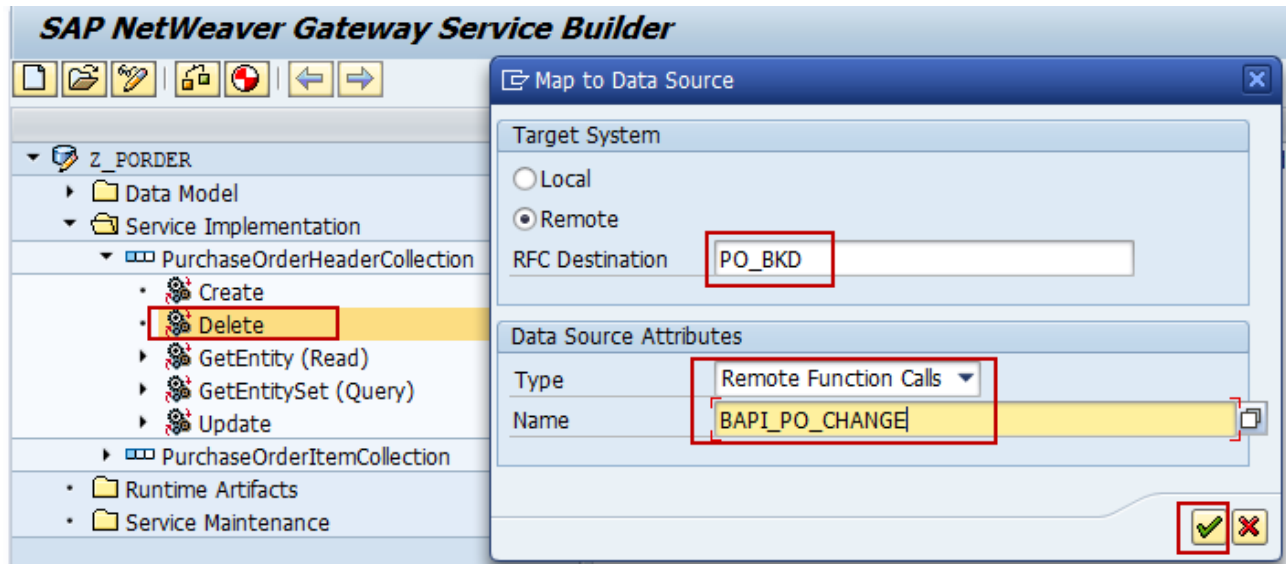
3.3.4 Delete operation

Purchase Order Header

1. Expand **Service Implementation -> PurchaseOrderHeaderCollection**. Right click on **Delete** option, select **Map to Data Source** and enter the required fields as below and click on the tick mark.

Type: Remote Function Calls

Name: BAPI_PO_CHANGE



2. The table below gives the mapping required for the Purchase Order Header Delete operation.

Entity Set Property	Constant Value	Mapping Direction	Data Source Parameter
	'X'	Input	POHEADER\DELETE_IND
	'X'	Input	POHEADER\PO_NUMBER
	'X'	Input	POHEADERX\DELETE_IND
PONumber		Input	PURCHASEORDER

3. The screenshot below shows the final mapping of the Data Source Parameters to the entity set properties for the Purchase Order Header Delete operation.

Property	Entity Set property	Constant	Mapping Direction	Direction	Data Source Parameter
		'X'	Input	→	POHEADER\PO_NUMBER
		'X'	Input	→	POHEADER\DELETE_IND
		'X'	Input	→	POHEADERX\DELETE_IND
🔑	PONumber		Input	→	PURCHASEORDER

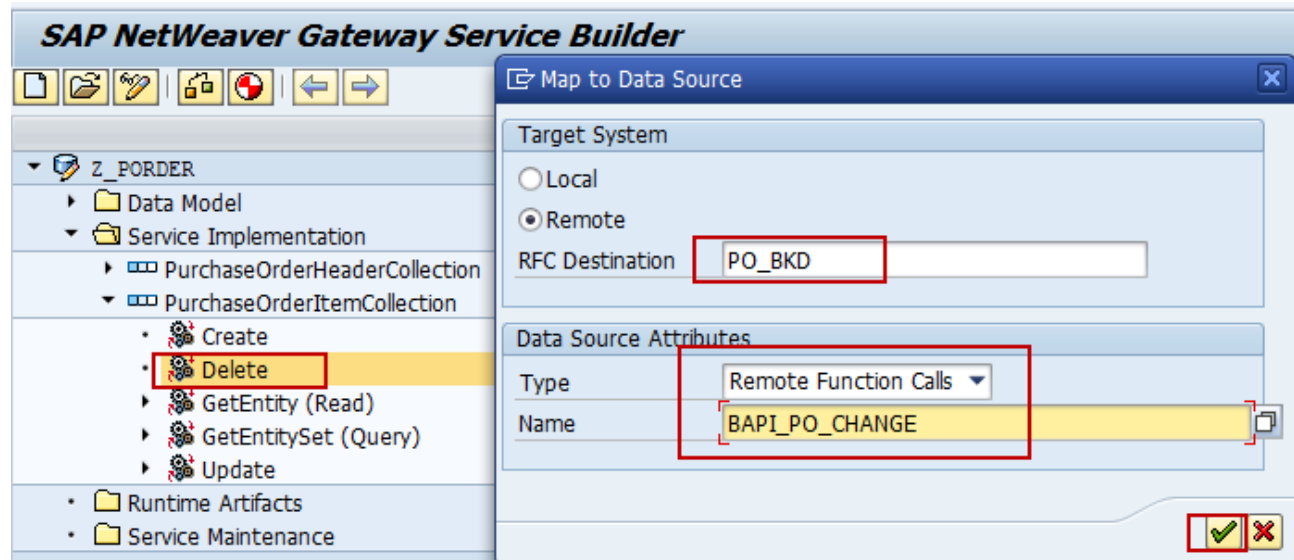
4. Click on **Save**.

Purchase Order Item

1. Expand **Service Implementation -> PurchaseOrderItemCollection**. Right click on **Delete**, select **Map to Data Source** option and enter the required fields as below and click on the tick mark.

Type: Remote Function Calls

Name: BAPI_PO_CHANGE



2. The table below gives the mapping required for the Purchase Order Item Delete operation.

Entity Set Property	Constant Value	Mapping Direction	Data Source Parameter
	'L'	Input	POITEM\DELETE_IND
	'L'	Input	POITEMX\DELETE_IND
	'X'	Input	POITEMX\PO_ITEMX
POItem		Input	POITEM\PO_ITEM
POItem		Input	POITEMX\PO_ITEM
PONumber		Input	PURCHASEORDER

3. The screenshot below shows the final mapping of the Data Source Parameters to the entity set properties for the Purchase Order Item Delete operation.

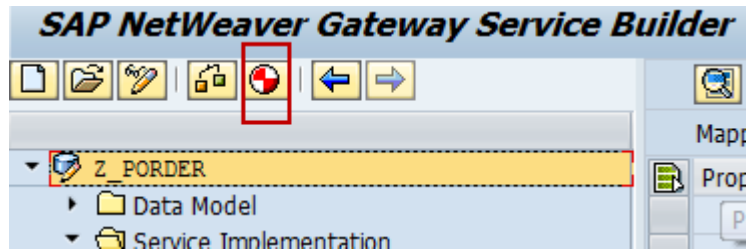
Mapping of Operation Delete for PurchaseOrderItemCollection						
Property	Entity Set property	Constant	Mapping Direction	Direction	Data Source Parameter	
		'L'	Input	→	POITEM\DELETE_IND	
		'L'	Input	→	POITEMX\DELETE_IND	
		'X'	Input	→	POITEMX\PO_ITEMX	
POItem	POItem		Input	→	POITEM\PO_ITEM	
POItem	POItem		Input	→	POITEMX\PO_ITEM	
PONumber	PONumber		Input	→	PURCHASEORDER	

4. Click on **Save**.

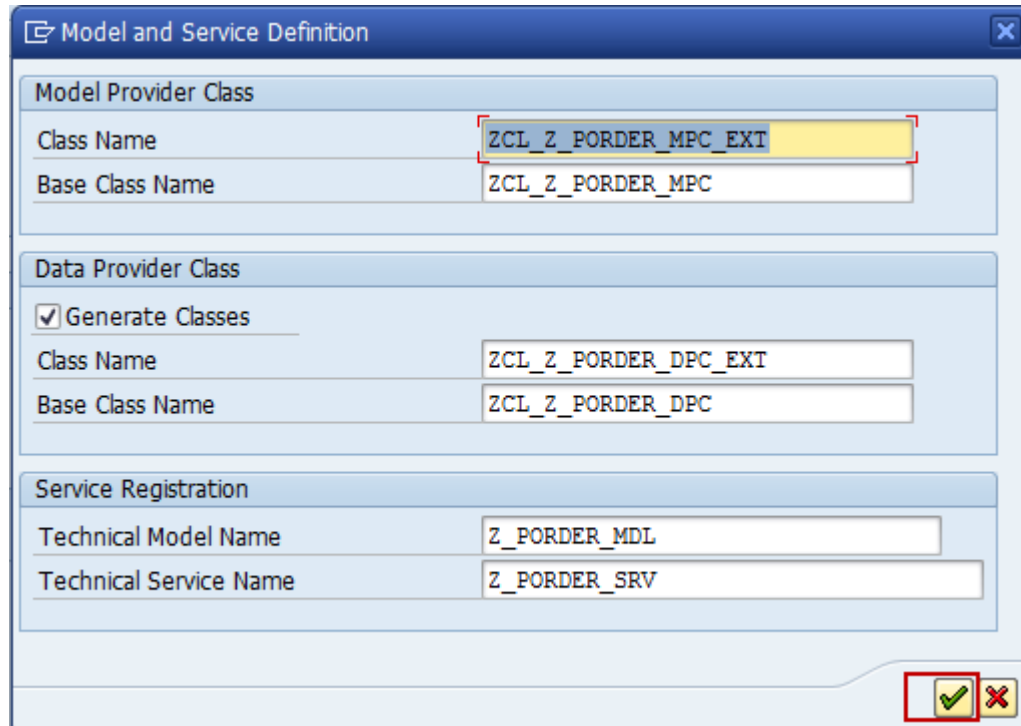
3.4 Generate OData service

The run time objects – model, service, metadata provider class and the data provider class are generated at the end of this step. The service maintenance is also done at the end of this step.

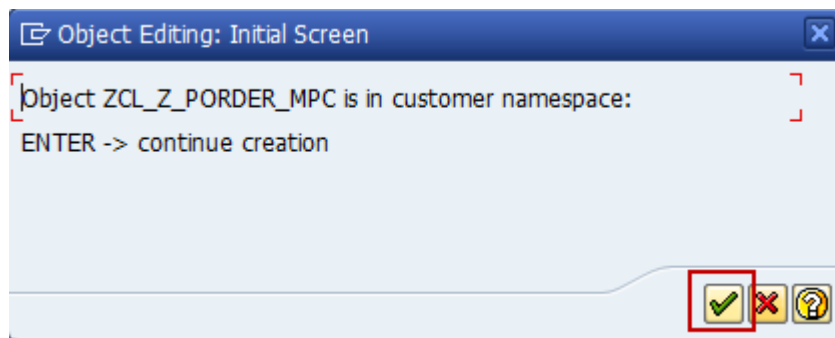
1. The OData service is generated by clicking on the **Generate Runtime Objects** option.



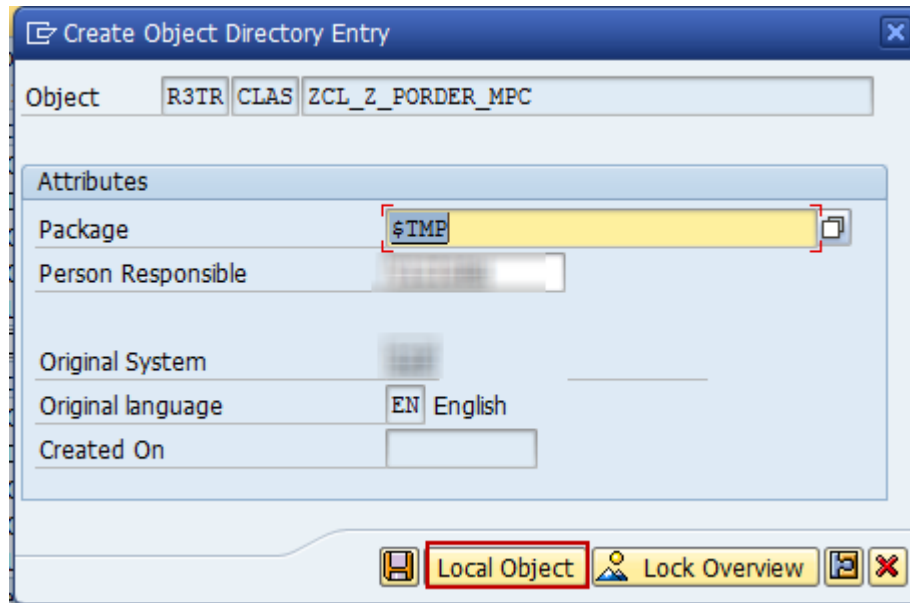
2. In the Model and Service Definition screen, press enter.



3. Click on the tick mark in the next screen.

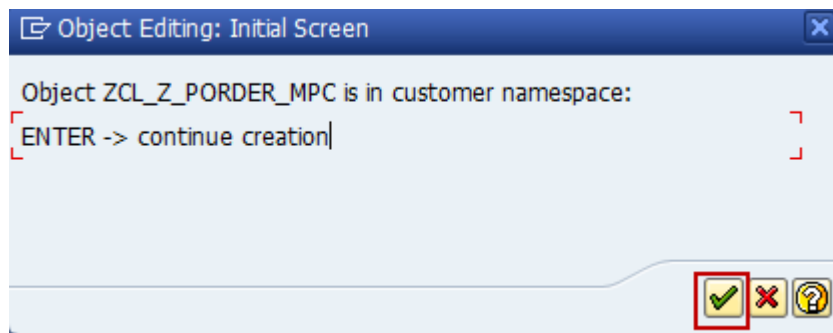


4. Choose **Local Object** in the Create Object Directory Entry screen.

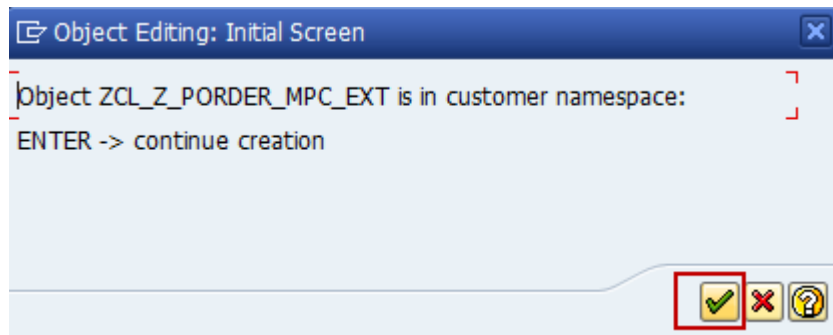


The 'Create Object Directory Entry' dialog box shows the 'Object' field with 'R3TR' selected, 'CLAS' selected, and 'ZCL_Z_PORDER_MPC' entered. The 'Attributes' section includes 'Package' (set to '\$TMP'), 'Person Responsible', 'Original System', 'Original language' (set to 'EN English'), and 'Created On'. At the bottom, the 'Local Object' button is highlighted with a red box.

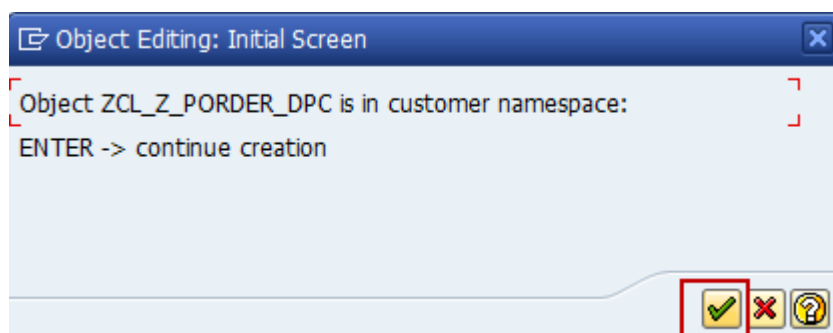
5. Click on the tick mark in the further screens.



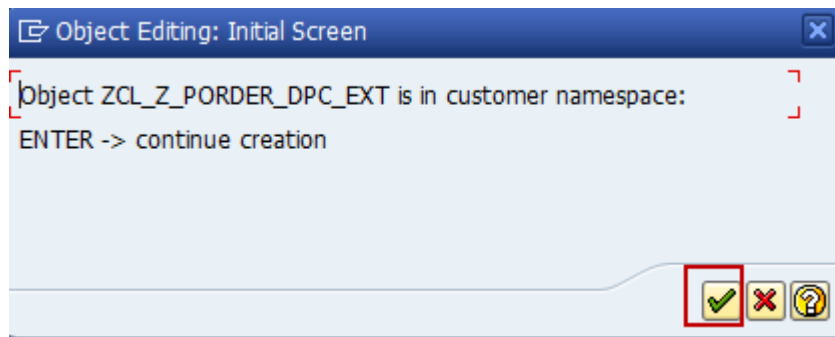
The 'Object Editing: Initial Screen' dialog box displays the message: 'Object ZCL_Z_PORDER_MPC is in customer namespace: ENTER -> continue creation'. At the bottom right, the 'Yes' button (represented by a green checkmark icon) is highlighted with a red box.



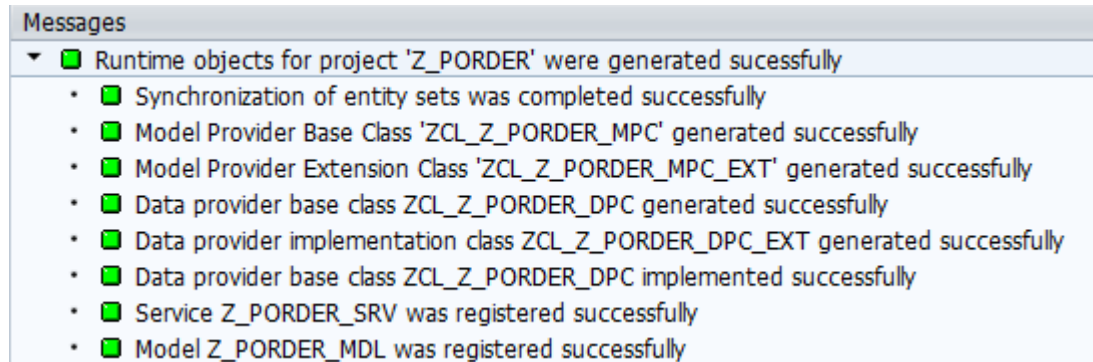
The 'Object Editing: Initial Screen' dialog box displays the message: 'Object ZCL_Z_PORDER_MPC_EXT is in customer namespace: ENTER -> continue creation'. At the bottom right, the 'Yes' button (represented by a green checkmark icon) is highlighted with a red box.



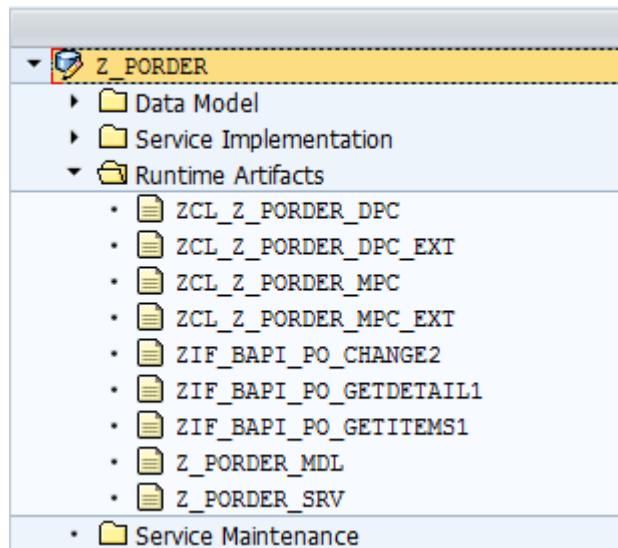
The 'Object Editing: Initial Screen' dialog box displays the message: 'Object ZCL_Z_PORDER_DPC is in customer namespace: ENTER -> continue creation'. At the bottom right, the 'Yes' button (represented by a green checkmark icon) is highlighted with a red box.



6. The success message of all the runtime artifacts generated is shown as below.



7. The runtime artifacts can also be seen in the project folder by expanding the **Runtime Artifacts** folder.



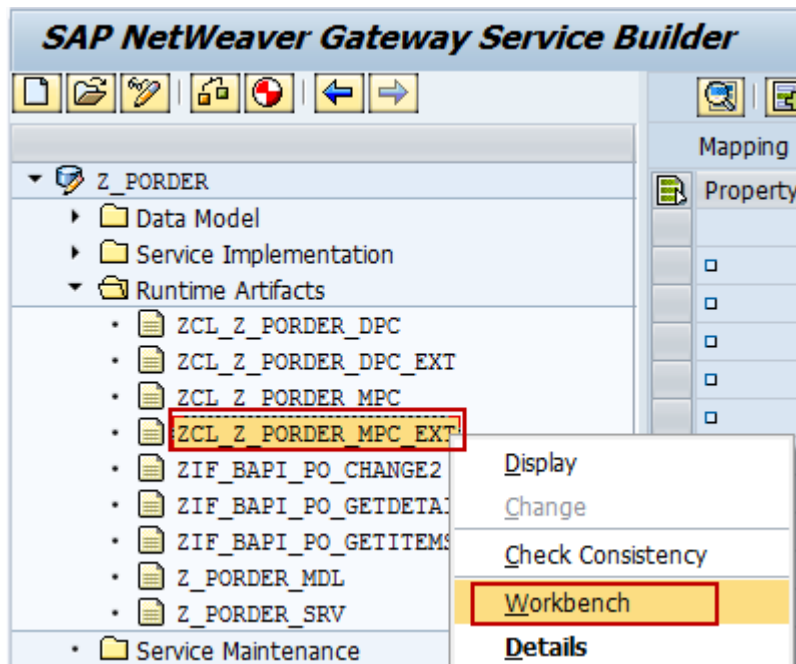
8. The code generated can be viewed in the classes shown in the Runtime Artifacts folder. The MPC class `zcl_z_porder_mpc` has the model definition and the DPC class `zcl_z_porder_dpc` has the run time implementation (mapping of the existing RFC). In this case any modifications or customization of the code for runtime can be done in the DPC class.

3.5 Modifying the Metadata Provider Class

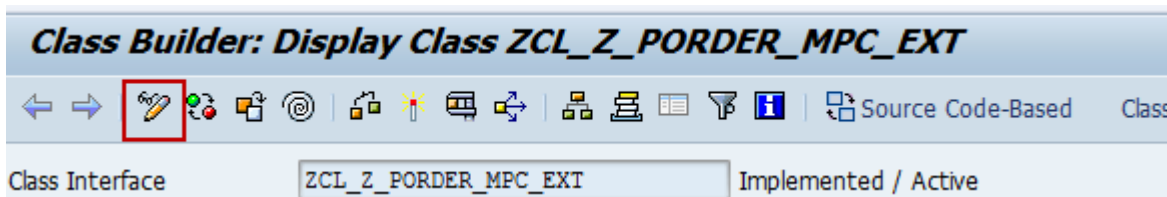
This section describes customizing the metadata provider class.

The runtime artifacts generated contains two MPC (metadata provider class) classes `zcl_z_porder_mpc` and `zcl_z_porder_mpc_ext`. To make changes to the model, the DEFINE method in the extension class `zcl_z_porder_mpc_ext` has to be re-defined.

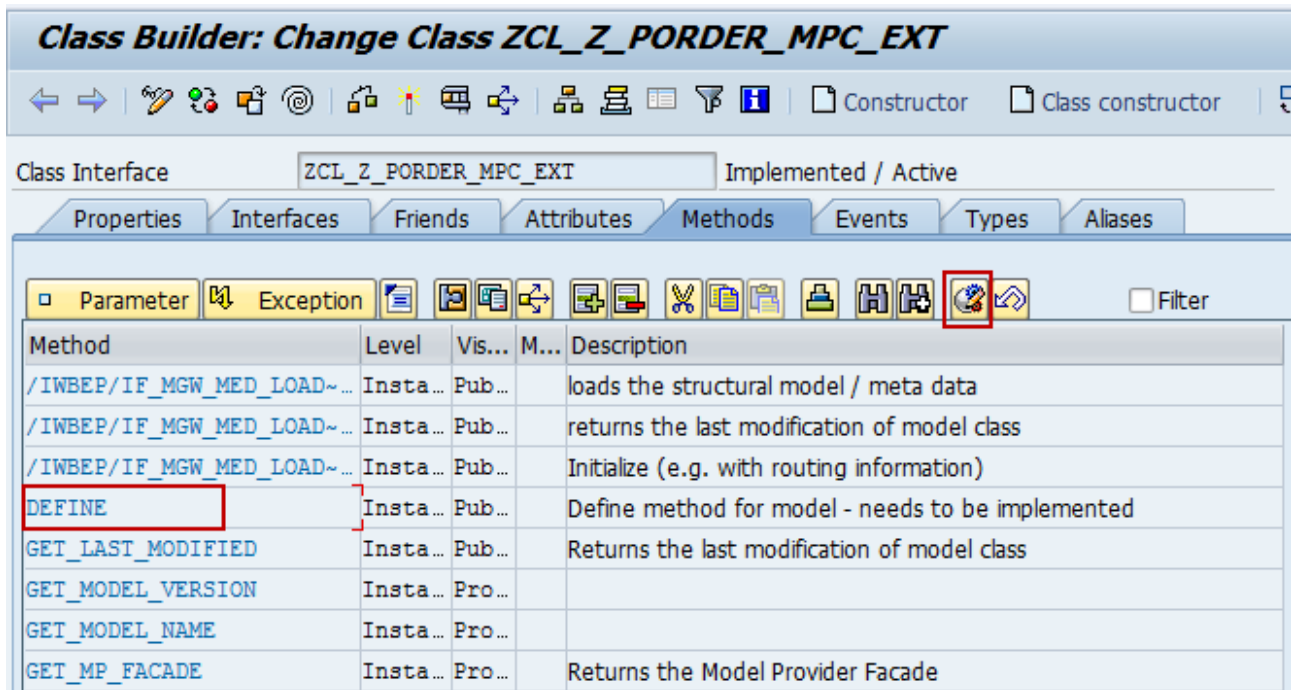
1. Go to the runtime artifacts folder in the project folder.
2. Right click on the MPC_EXT class and click on **Workbench** from the options.



3. Click on **Edit** option for the class.



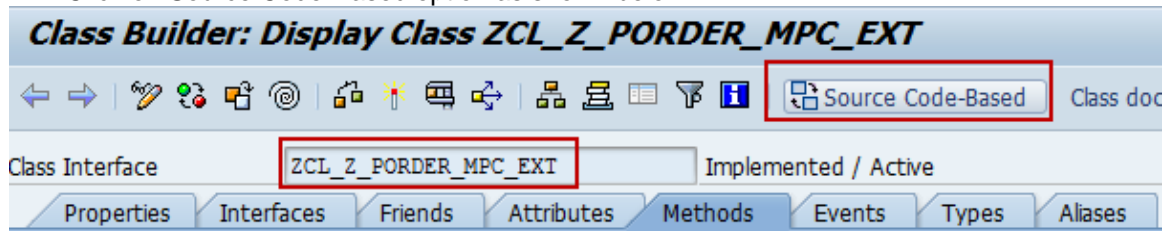
4. Select the DEFINE method and click on **Redefine** option.



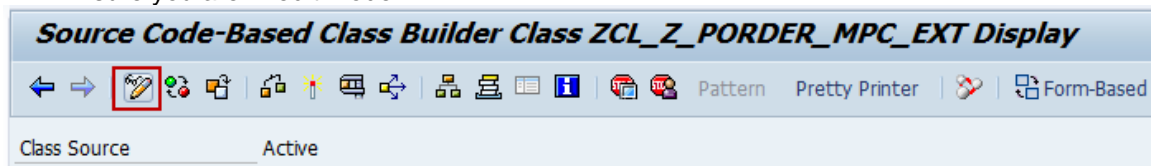
5. Write suitable code here to include filterable properties for the properties or setting the creatable, updatable, deletable flags for the entities. The creatable, updatable, deletable flags on the entities have to be set to true to enable the Create, Update and Delete operations on these entities. Sample code can be found from [define_method.txt](#)
6. Activate the MPC_EXT class.
7. Select all the inactive objects in the MPC_EXT class and click enter.
8. Click on back and navigate back to Service Builder screen.

Follow the steps below to create proxies required for implementing deep create functionality.

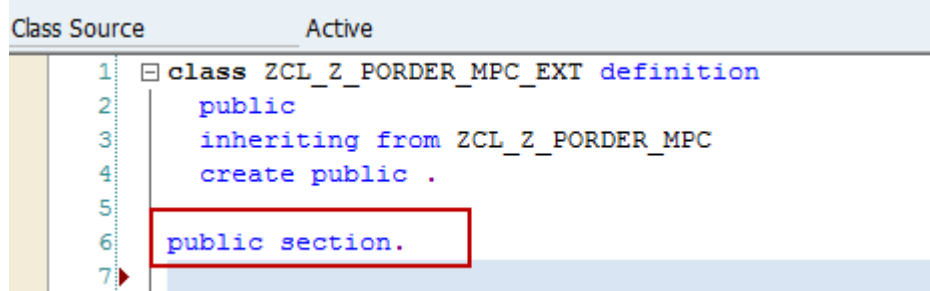
1. Click on Source Code-Based option as shown below.



2. Ensure you are in edit mode.



3. Add the code from [proxy_for_deep_create.txt](#) under the public section of the class shown below.



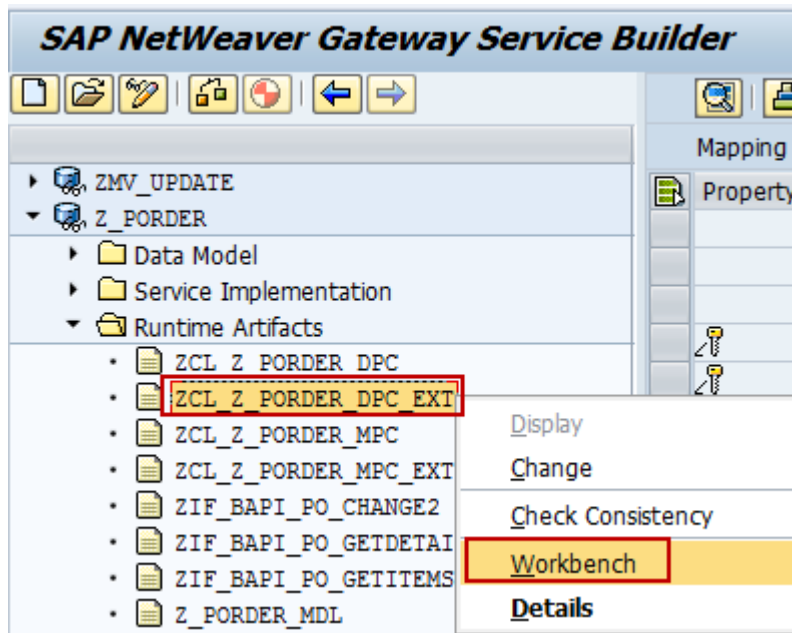
4. Click on Save and Activate.

3.6 Implementing Data Provider Class

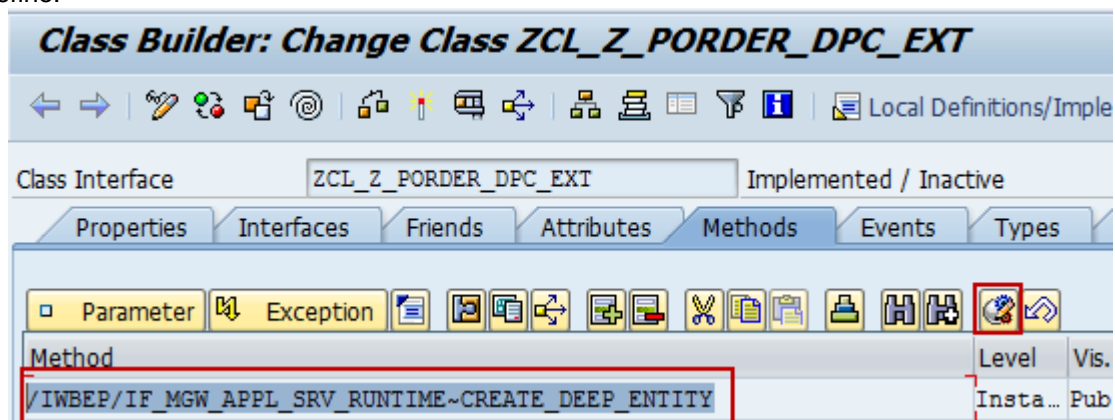
Deep Create functionality is used to create an entity with deep or nested data provided in an inline format. SAP NetWeaver Gateway Service Builder tool does not support Deep Create implementation. Hence this has to be implemented after the runtime artifacts are generated. The method that needs to be redefined in the Data Provider extension class to implement the deep create functionality is

/IWBEPIF_MGW_APPL_SRV_RUNTIME~CREATE_DEEP_ENTITY in ZCL_Z_PORDER_DPC_EXT class.

1. From the project folder, expand **Runtime Artifacts**.
2. Right click on ZCL_Z_PORDER_DPC_EXT and choose **Workbench**.



3. Ensure you are in edit mode.
4. Select the method /IWBEF/IF_MGW_APPL_SRV_RUNTIME~CREATE_DEEP_ENTITY and choose redefine.



5. Replace the method with the code from the file [deep_create.txt](#)

3.7 Register Service in SAP NetWeaver Gateway system

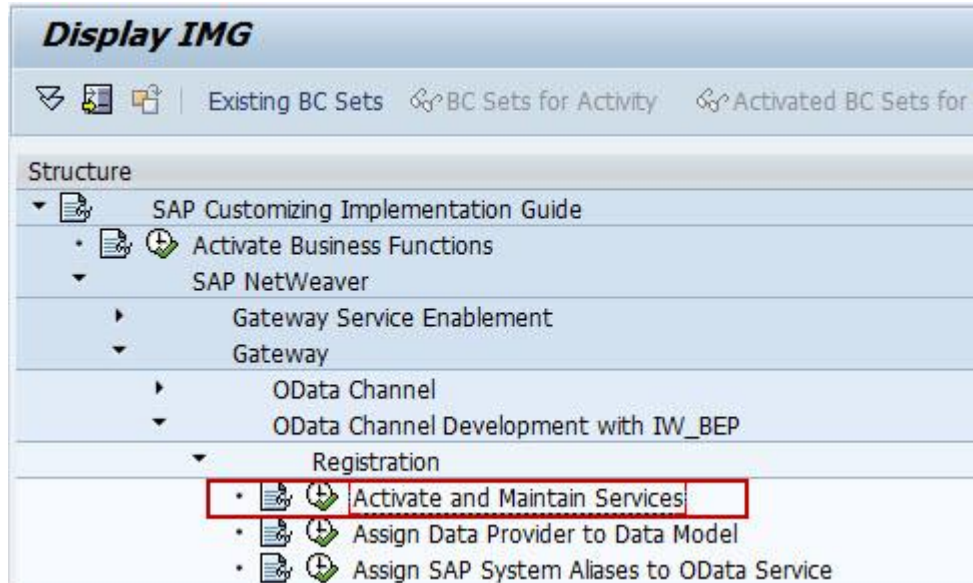
Before registering the service in the SAP NetWeaver Gateway system, ensure you create system alias to the SAP system with the **For Local App** checked as shown below. This can be created by the following navigation **SAP NetWeaver -> Gateway -> Former Development -> Generic Channel -> Configuration -> Connection Settings -> SAP NetWeaver Gateway to SAP System -> Manage SAP System Aliases** in the SAP NetWeaver Gateway system.

Change View "Manage SAP System Aliases": Overview					
BC Set: Change Field Values					
Manage SAP System Aliases					
SAP System Alias	Description	Local GW	For Local App	RFC Destination	Software Version
PO_BKD	System Alias to the PO backend	<input type="checkbox"/>	<input checked="" type="checkbox"/>	PO_BKD	DEFAULT

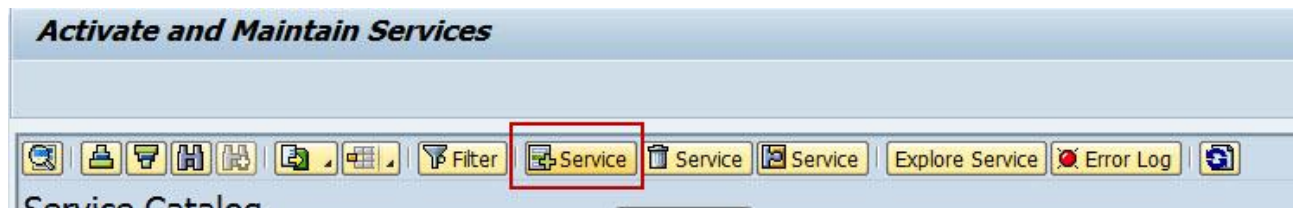
This section describes the steps in registering the service in the SAP NetWeaver Gateway system.

1. Go to transaction **SPRO** in the SAP NetWeaver Gateway system.

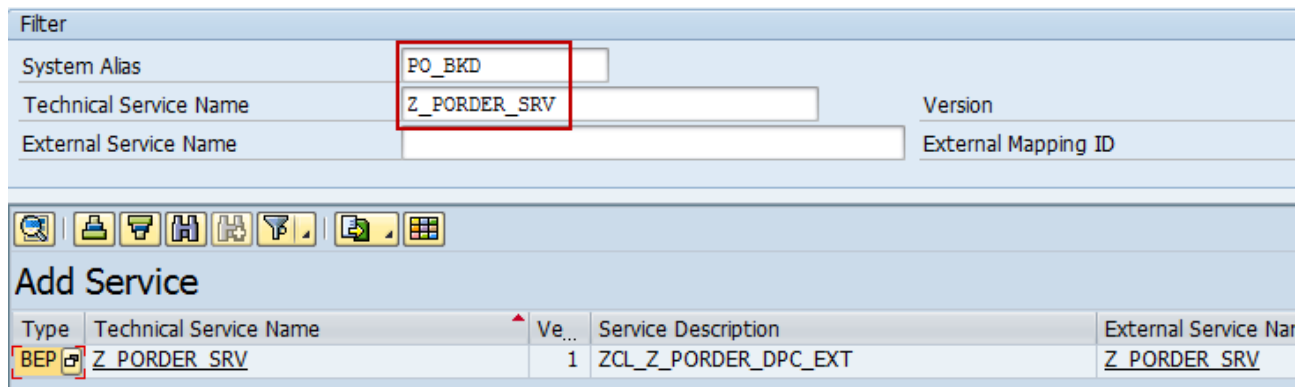
2. Navigate to the path **SAP NetWeaver → Gateway → OData Channel Development with IW_BEP → Registration → Activate and Maintain Services.**



3. Click on **Add service.**



4. Use the System Alias created for the SAP system and the Technical Service Name and add the service.
Technical Service Name: Z_PORDER_SRV.



5. Click on service name.
6. Enter \$tmp in the package and click on enter.

Add Service

Service	
Technical Service Name	Z_PORDER_SRV
Service Version	1
Description	ZCL_Z_PORDER_DPC_EXT
External Service Name	Z_PORDER_SRV
Namespace	
External Mapping ID	
External Data Source Type	C
Model	
Technical Model Name	Z_PORDER_MDL
Model Version	1
Creation Information	
Package	\$TMP
<input checked="" type="checkbox"/> Use current client if 'sap-client' not specified in URL	
ICF Node	
<input checked="" type="radio"/> Standard Mode <input type="radio"/> None <input type="radio"/> Compatibility Mode for SP 02	

☒ ☐ ☐

7. Click on the tick mark in the following screens.

Object Editing: Initial Screen

Object Z_PORDER_MDL_0001_BE is in customer namespace:
ENTER -> continue creation

☒ ☐ ☐

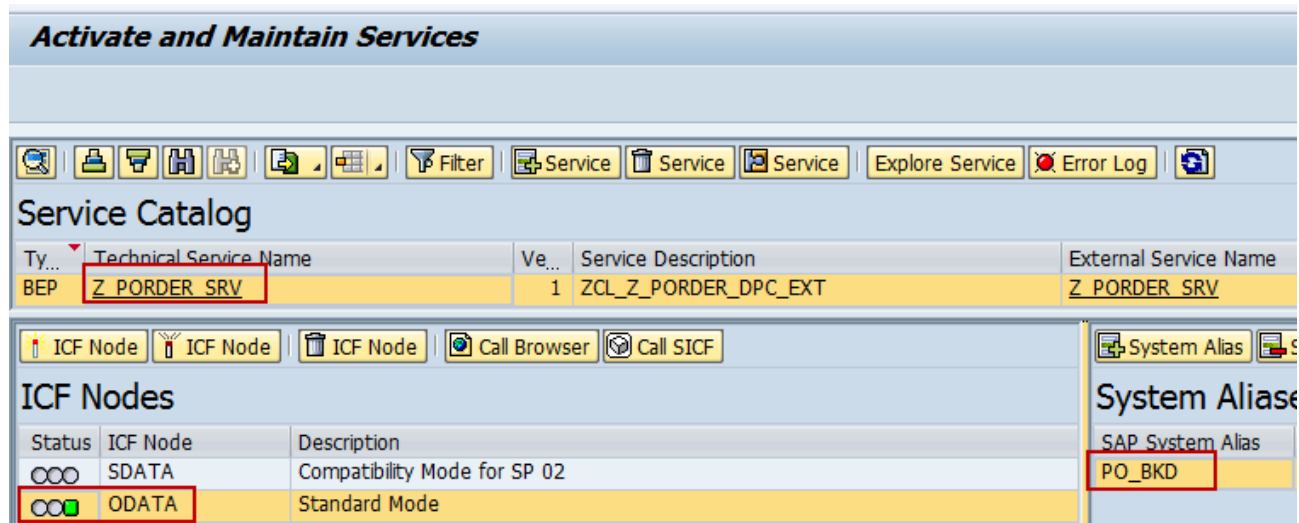
Object Editing: Initial Screen

Object Z_PORDER_MDL_0001_BE is in customer namespace:
ENTER -> continue creation

☒ ☐ ☐



8. Click on back. The OData Service created can be found in the service catalogue list.



This completes the creating, implementing and registering the OData service for Purchase Order CRUDQ calls.

4 Test OData Service

This section describes the steps to test the various calls in the service. In this document, we will consume the OData service using a SharePoint application. Customers can also build similar applications to use the service according to their requirements.

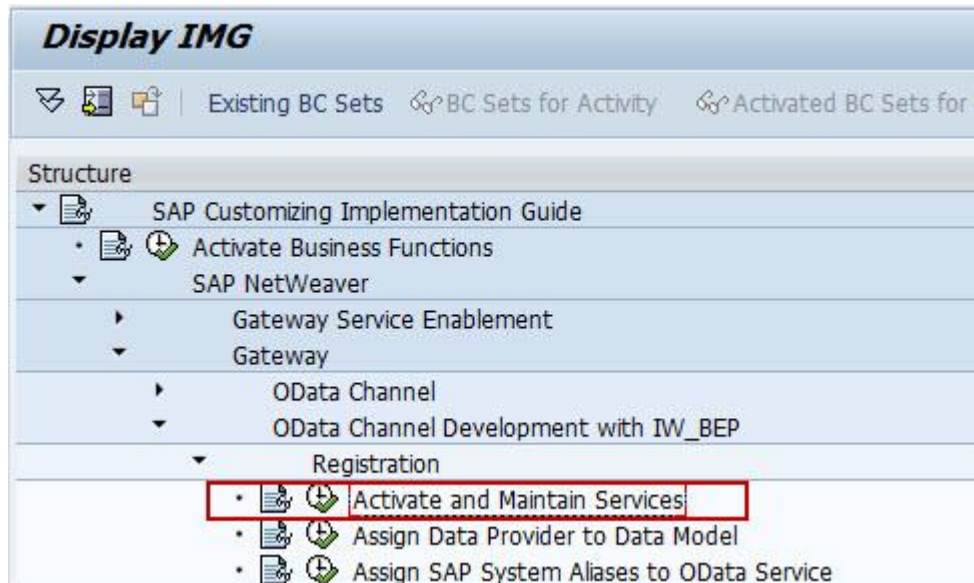
Note: 1. Replace <host> and <port> with the actual values in the request URLs.

4.1 Service Document

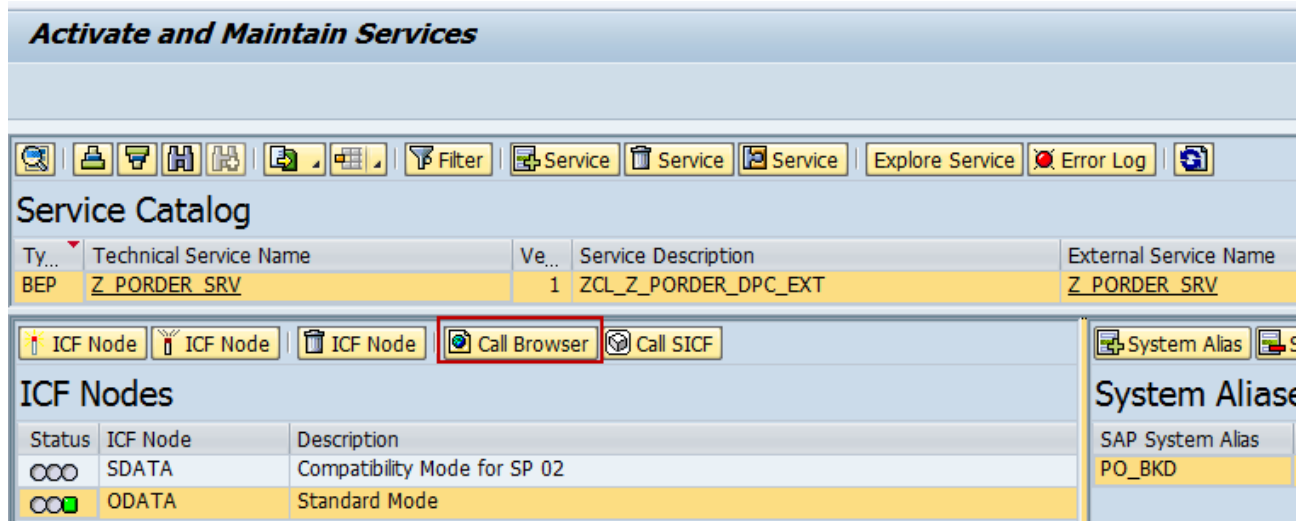
Service document lists all the available entities and collections. It describes the underlying data model.

Steps to view the Service document:

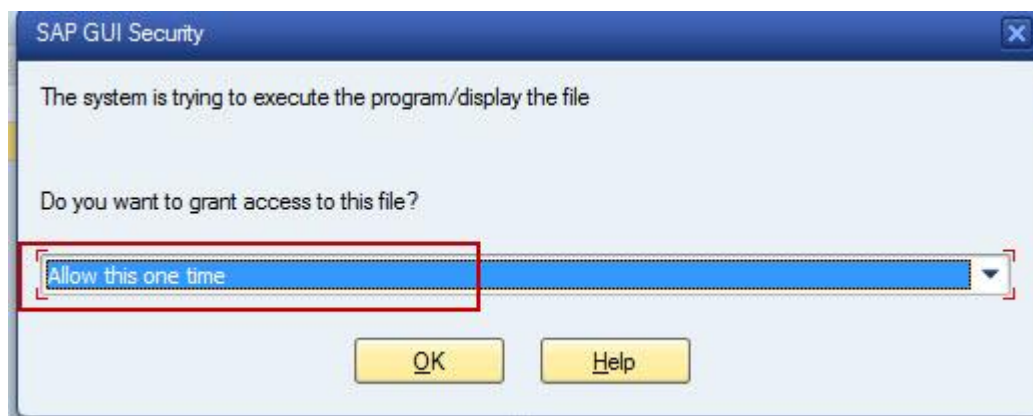
1. Go to transaction **spro** in the SAP NetWeaver Gateway system.
2. Navigate to the path **SAP NetWeaver → Gateway → OData Channel Development with IW_BEP → Registration → Activate and Maintain Services**



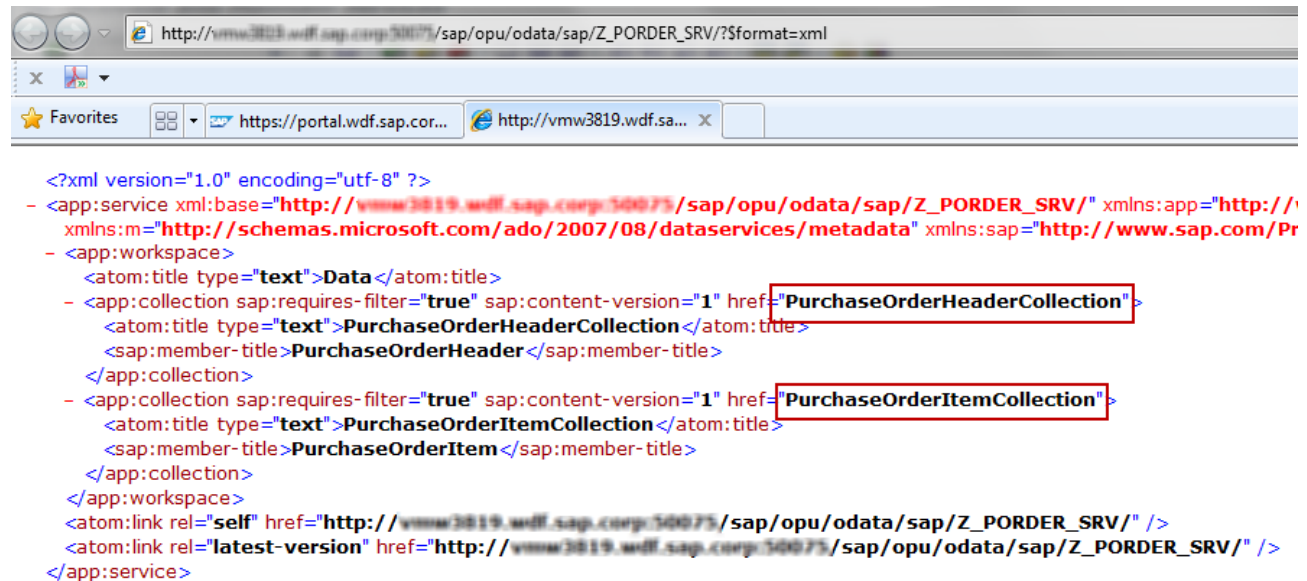
3. Select the row with the Technical Service Name Z_PORDER_SRV.
4. Click on the Call Browser option in the ICF node area for the service.



5. Select **Allow this one time** in the SAP GUI Security screen and click on OK.



6. The service Document is now displayed in the browser. Enter the user credentials of the SAP NetWeaver Gateway system when prompted.



The Service document can be accessed through the URL
[http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/?\\$format=xml](http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/?$format=xml)

4.2 Metadata

The format of the URL is [http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/\\$metadata](http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/$metadata).

The screenshot below shows the Purchase order header and Purchase order item entities in the metadata along with the properties.

```

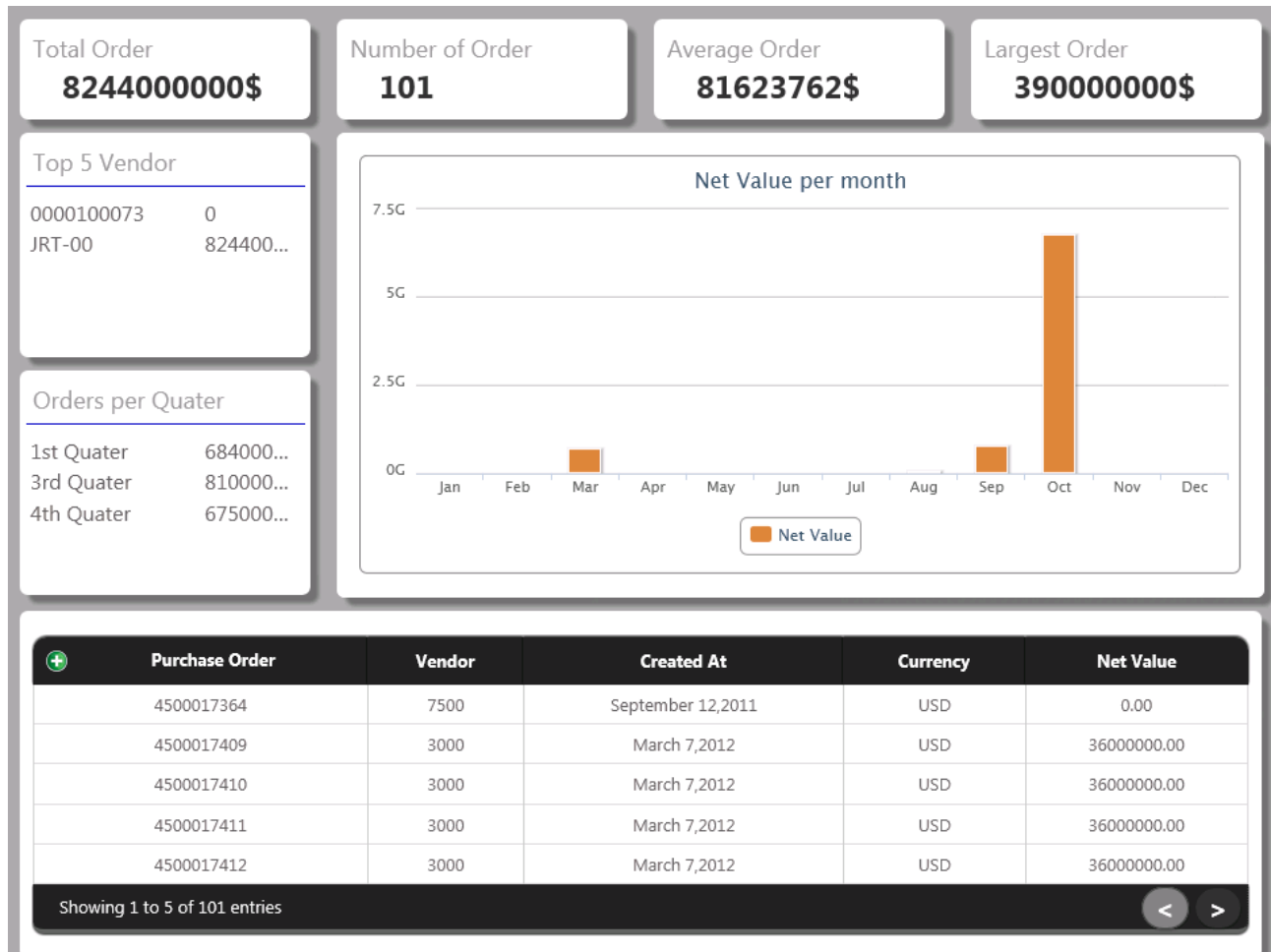
- <EntityType Name="PurchaseOrderHeader" sap:content-version="1">
+ <Key>
  <Property Name="TotalValueAtRelease" Type="Edm.Decimal" Precision="15" Scale="2" />
  <Property Name="PurchaseOrgGroup" Type="Edm.String" MaxLength="3" />
  <Property Name="PurchaseOrganization" Type="Edm.String" MaxLength="4" />
  <Property Name="VendorAccountNumber" Type="Edm.String" MaxLength="10" />
  <Property Name="CreatedBy" Type="Edm.String" MaxLength="12" />
  <Property Name="CreatedAt" Type="Edm.DateTime" />
  <Property Name="PurchaseDocStatus" Type="Edm.String" MaxLength="1" />
  <Property Name="PurchaseDocType" Type="Edm.String" MaxLength="4" />
  <Property Name="PurchaseDocCategory" Type="Edm.String" MaxLength="1" />
  <Property Name="CompanyCode" Type="Edm.String" MaxLength="4" />
  <Property Name="PONumber" Type="Edm.String" Nullable="false" MaxLength="10" />
  <NavigationProperty Name="PurchaseOrderItem" Relationship="Z_PORDER_SRV.POHeaderRelatedItem"
    FromRole="FromRole_POHeaderRelatedItem" ToRole="ToRole_POHeaderRelatedItem" />
</EntityType>
- <EntityType Name="PurchaseOrderItem" sap:content-version="1">
+ <Key>
  <Property Name="MaterialType" Type="Edm.String" MaxLength="4" />
  <Property Name="WeightUnit" Type="Edm.String" MaxLength="3" />
  <Property Name="NetWeight" Type="Edm.Decimal" Precision="13" Scale="3" />
  <Property Name="NetValue" Type="Edm.Decimal" Precision="23" Scale="4" />
  <Property Name="GrossValue" Type="Edm.Decimal" Precision="23" Scale="4" />
  <Property Name="PriceUnit" Type="Edm.Decimal" Precision="5" Scale="0" />
  <Property Name="NetPrice" Type="Edm.Decimal" Precision="23" Scale="4" />
  <Property Name="OrderPriceUnit" Type="Edm.String" MaxLength="3" />
  <Property Name="Unit" Type="Edm.String" MaxLength="3" />
  <Property Name="Quantity" Type="Edm.Decimal" Precision="13" Scale="3" />
  <Property Name="TargetQuantity" Type="Edm.Decimal" Precision="13" Scale="3" />
  <Property Name="MaterialGroup" Type="Edm.String" MaxLength="4" />

```

4.3 SharePoint Purchase Order Application

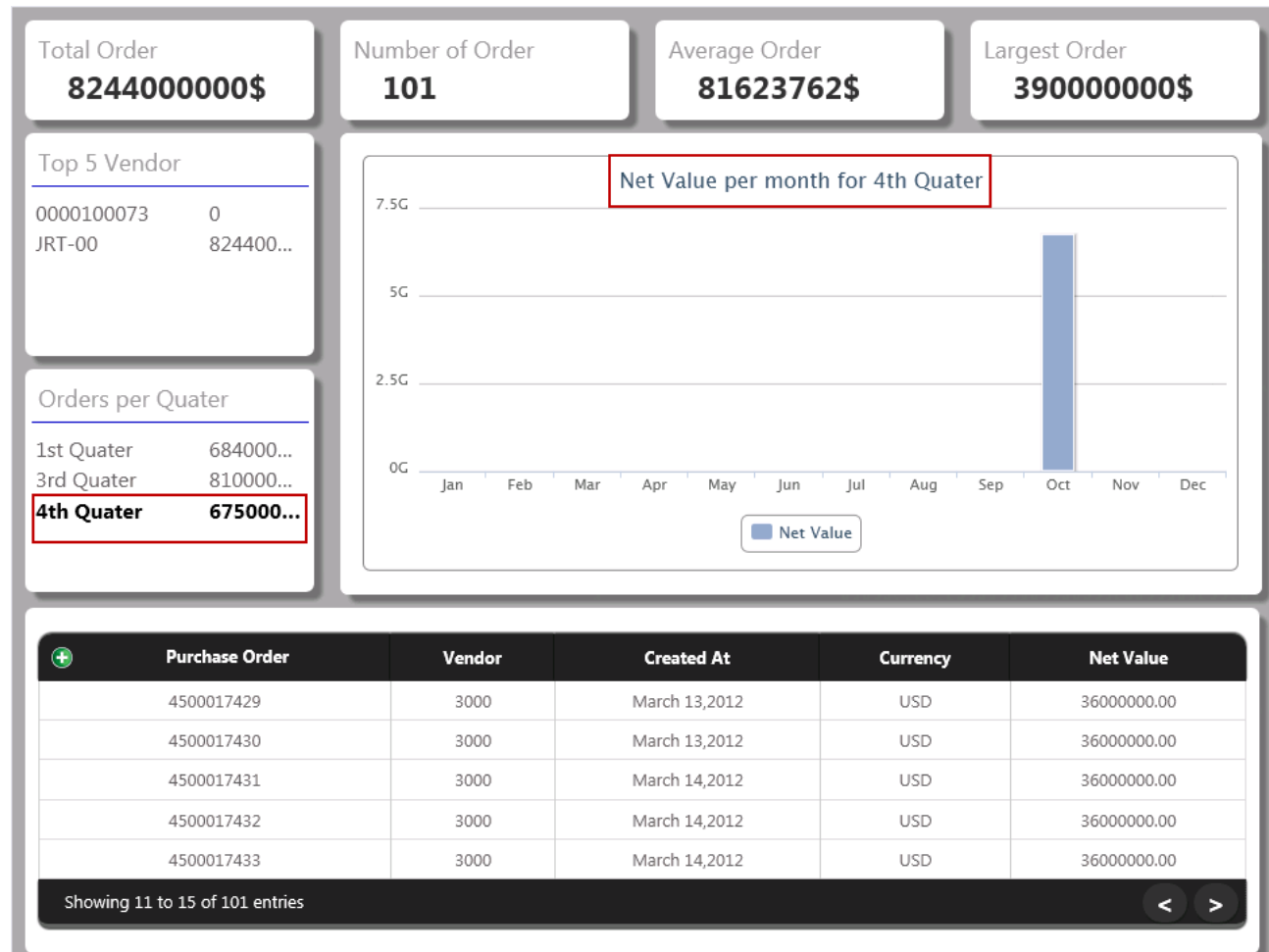
Note: The SharePoint application mentioned in this guide is just for showcasing how we can consume the Purchase Order service. However, this document doesn't contain steps to build a SharePoint 2013 application.

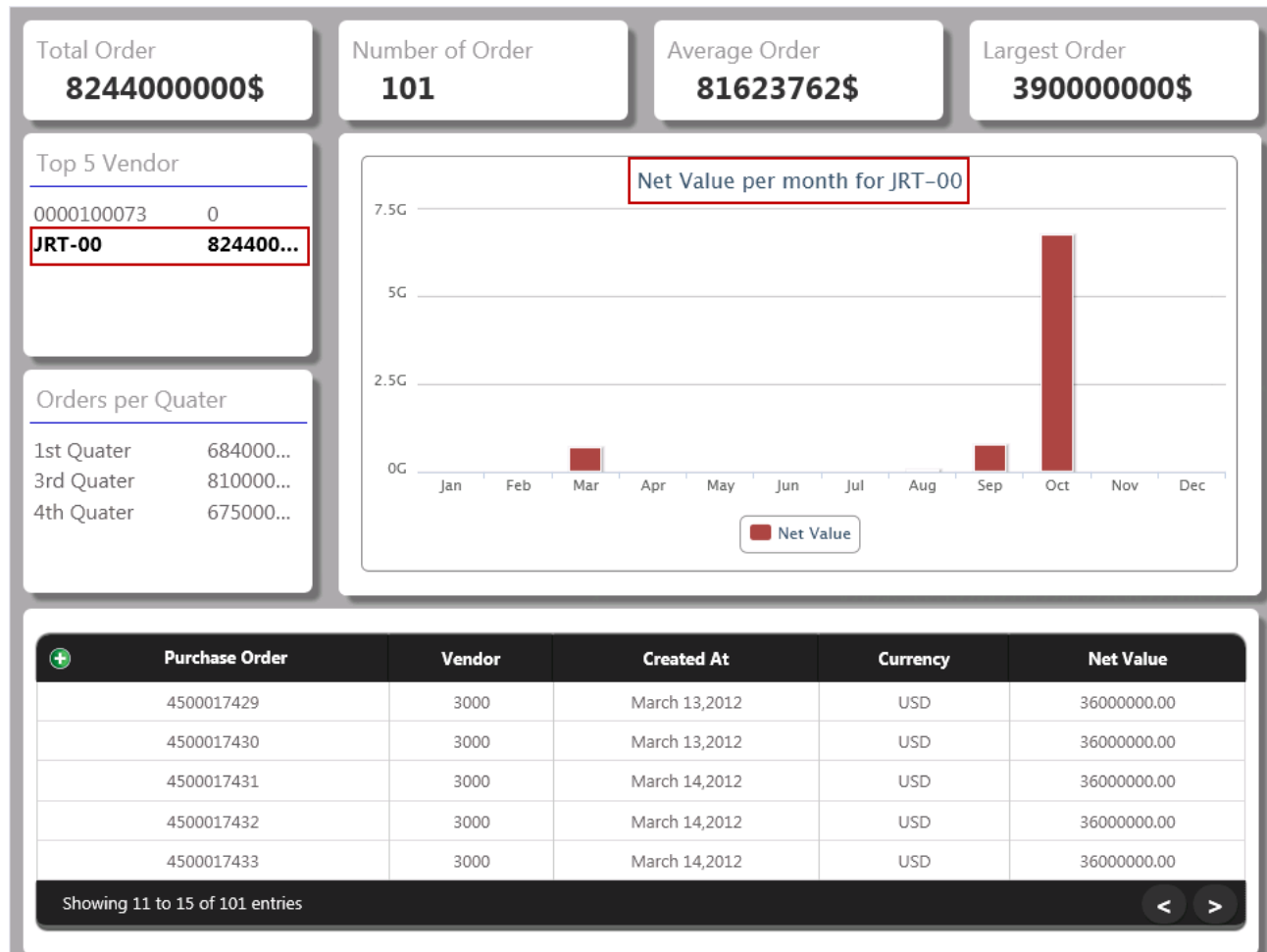
Following is the screen-shot of the Purchase Order SharePoint Application using which we will be testing the service. The application has been built to consume the Purchase Order service created in this guide using Duet Enterprise 2.0 as a platform



The application showcases the various purchase orders in the SAP system along with the details for each purchase order including Vendor, Created At, Currency etc.

The application also lists the Total Order, Number of Order, Average Order, Largest Order, etc. The top 5 Vendors as well as the Orders per Quarter are also retrieved and showcased in the application both in a list as well as in a graphical format.

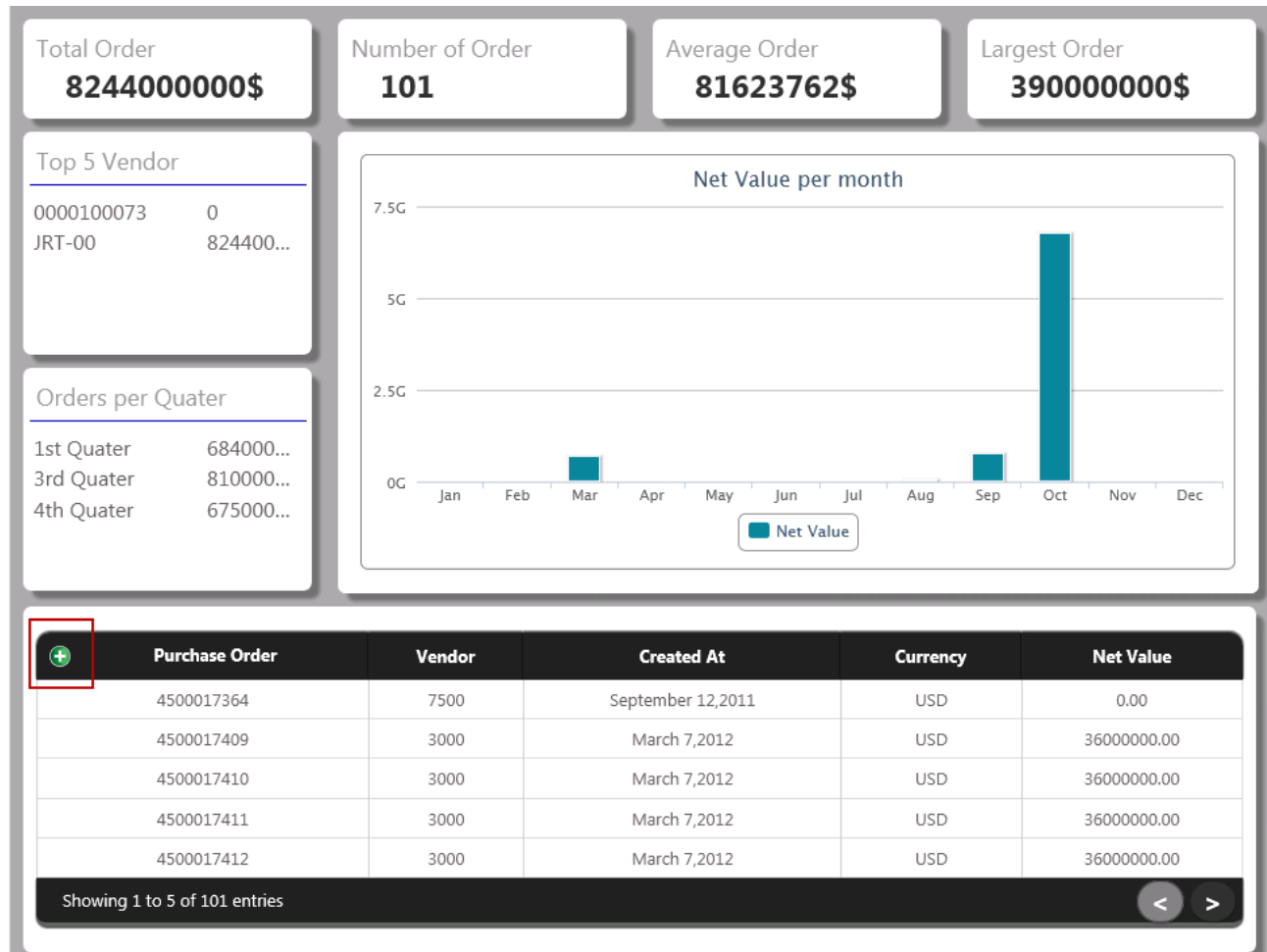




4.4 Create (Deep Insert) operation

4.4.1 Execute the Create operation

1. In order to create a Purchase Order, click on the "Add" button.



- This opens up a form in the application in which the details of the Purchase Order to be created has to be entered.

Item	Description	Quantity	Net Value

3. In order to enter the Billing Address of the Purchase Order, click on the Billing Address which opens up a list of addresses from which the Billing Address can be selected. Similarly the Dispatch Address can be chosen.

Billing Address

Exotic Liquids
49 Gilbert St.
London EC1 4SD
(171) 555-2222

Item **Description** **Net Value**

ID	Description
1	Exotic Liquids
2	New Orleans Cajun Delights
3	Grandma Kelly's Homestead
4	Tokyo Traders
5	Cooperativa de Quesos 'Las Cabras'
6	Mayumi's
7	Pavlova, Ltd.
8	Specialty Biscuits, Ltd.
9	PB Knäckebröd AB
10	Refrescos Americanas LTDA

Showing 1 to 10 of 29 entries

4. In order to enter the Item, Description, Quantity and Net Value details, click on the Item column. This displays a form from which you can choose the Item ID and the description.

Dispatch Address

[illegible]

The screenshot displays a web-based form for creating a Purchase Order. On the left, a 'Billing Address' section contains the text: 'Exotic Liquids', '49 Gilbert St.', 'London EC1 4SD', and '(171) 555-2222'. Below this is a table with columns 'Item' and 'Net Value'. A modal dialog box is open in the center, featuring a search bar and a table with 'ID' and 'Description' columns. The table lists three items: ID 1 with description '2222', ID 2 with description 'Chang', and ID 3 with description 'Aniseed Syrup'. The dialog also shows 'Showing 1 to 3 of 3 entries' and navigation arrows.

ID	Description
1	2222
2	Chang
3	Aniseed Syrup

Showing 1 to 3 of 3 entries

Similarly, you can also enter the Quantity and the Net Value.

Billing Address


Exotic Liquids
49 Gilbert St.
London EC1 4SD
(171) 555-2222


Dispatch Address


Exotic Liquids
49 Gilbert St.
London EC1 4SD
(171) 555-2222

Item	Description	Quantity	Net Value
1	2222	4	2.82

5. Now we have to select the “SAVE” button at the top which will create the Purchase Order in the SAP system using the Purchase Order service.


Save


Back



Billing Address

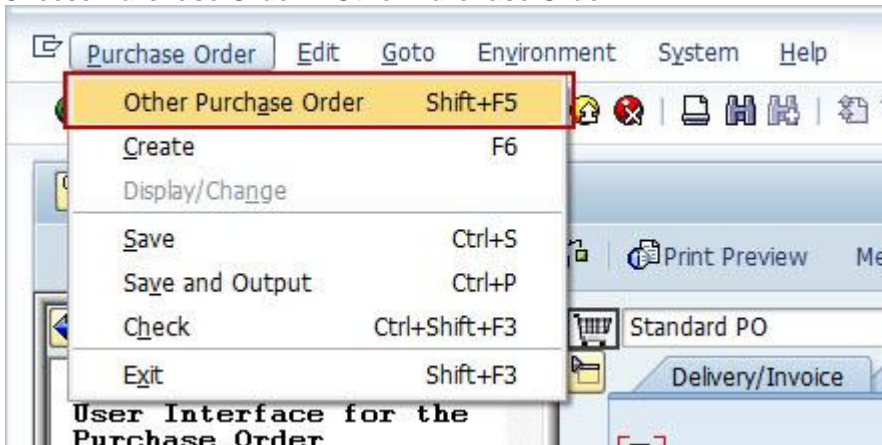
Exotic Liquids
49 Gilbert St.
London EC1 4SD
(171) 555-2222

Dispatch Address

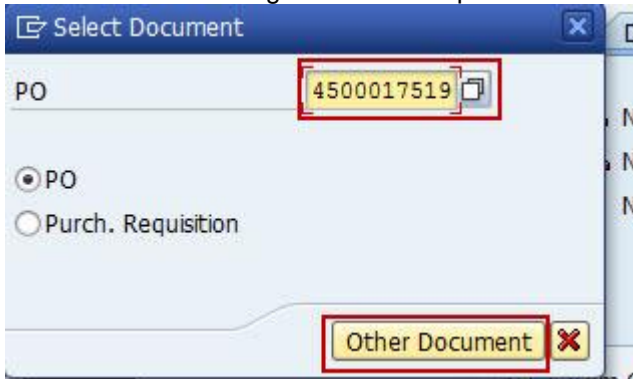
Exotic Liquids
49 Gilbert St.
London LA EC1 4SD
(171) 555-2222
#CAJUN.HTM#

Item	Description	Quantity	Net Value
1	Chai	1	18





6. The SharePoint application should use the following URL and make a HTTP POST call in this case http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderHeaderCollection
7. The details of the Purchase Order to be created should be passed as xml value to the SAP system. The file [create.txt](#) gives a sample deep create request body which can be used to create a purchase order.
8. To check the Purchase Order created in the SAP backend system, go to transaction **me21n** in the SAP backend system.
9. Choose **Purchase Order-> Other Purchase Order**.





10. Enter the PO Number generated in step 5 and click on **Other Document**.



11. This displays the Purchase Order created in this section.

Standard PO		4500017519	Vendor	100073 Proveedor 1	Doc. date	09.10.2012
<div>Delivery/Invoice</div> <div>Conditions</div> <div>Texts</div> <div>Address</div> <div>Communication</div> <div>Partners</div> <div>Additional Data</div> <div>Org. Data</div> <div>Status</div>						
	Active	Ordered	4	PC	11,28	ARS
	Not Yet Sent	Delivered	0	PC	0,00	ARS
	Not Delivered	Still to deliv.	4	PC	11,28	ARS
	Not Invoiced	Invoiced	0	PC	0,00	ARS
		Down paymts			0,00	ARS

		S..	Itm	A	I	Material	Short Text	PO Quantity	O...	C	Deliv. Date	Net Price	Curr...	Per	O...
			1			1108	2222	4	PC	D	15.10.2012	2,82	ARS	1	PC

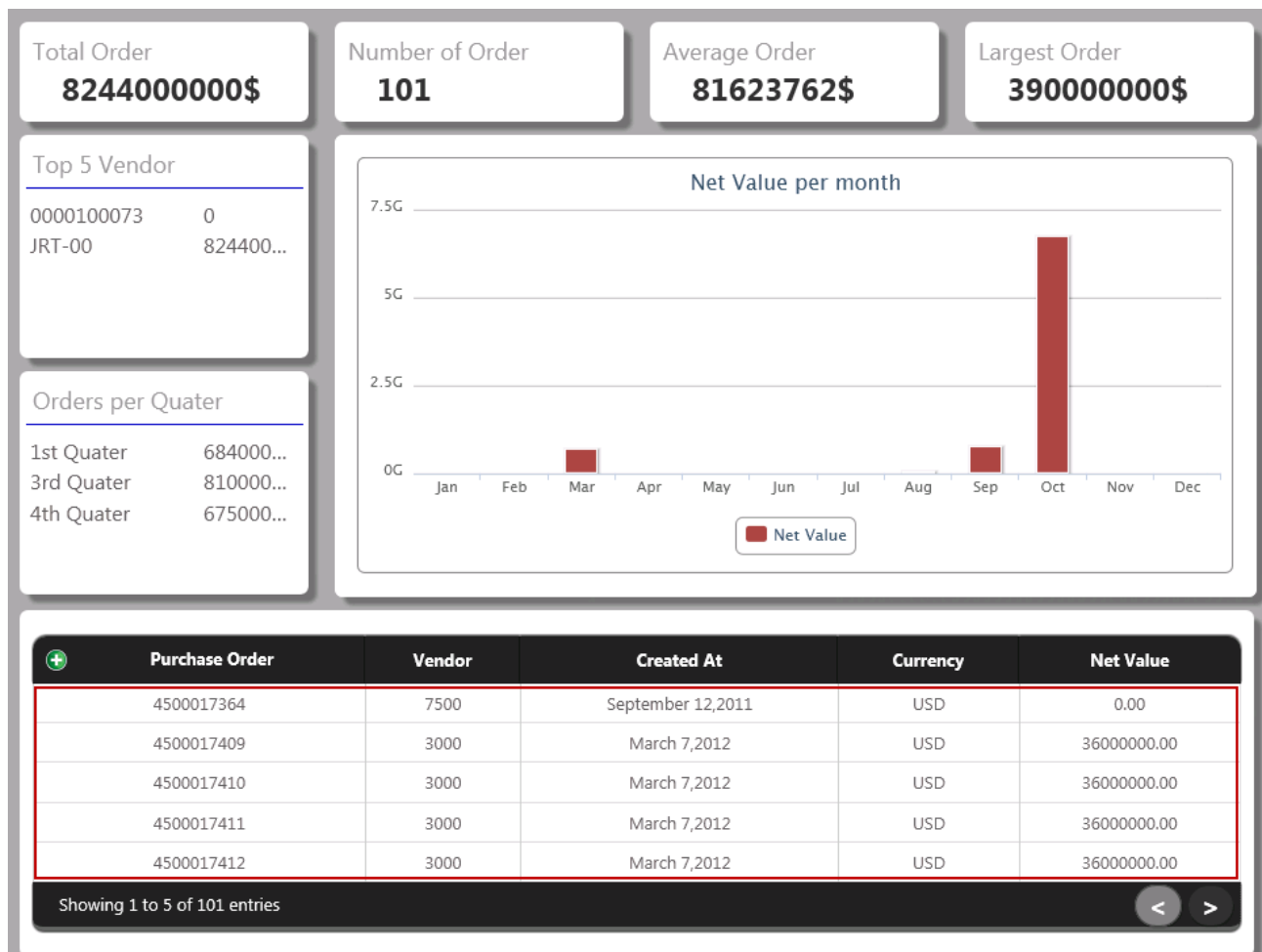
4.5 Query operation

Note: Ensure you change the host and port values in the Request URL.

4.5.1 Read all Purchase Order Header entries

1. In case of the Query operation, the SharePoint application uses the URL http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderHeaderCollection

To fetch the list of all the Purchase Orders (for the particular user in whose context the call is fired) from the SAP system through the SAP NetWeaver Gateway system.



4.5.2 Read all Purchase Order Item entries

1. Similarly, in order to fetch all the Purchase Order items from the SAP system, the application should use the URL http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderItemCollection

This will fetch the list of the Purchase Order items from the SAP system.

4.6 Read operation

4.6.1 Read specific Purchase Order Header entry

1. In order to read a specific Purchase Order Header entry the SharePoint application should use the following URL [http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderHeaderCollection\('4500017519'\)](http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderHeaderCollection('4500017519')) In this case, the Header details of a specific Purchase Order (4500017519 in this case) is fetched.
2. In order to view a specific Purchase Order entry from the SharePoint application, click on any of the Purchase Orders from the list which will show the details of the order in a new window.

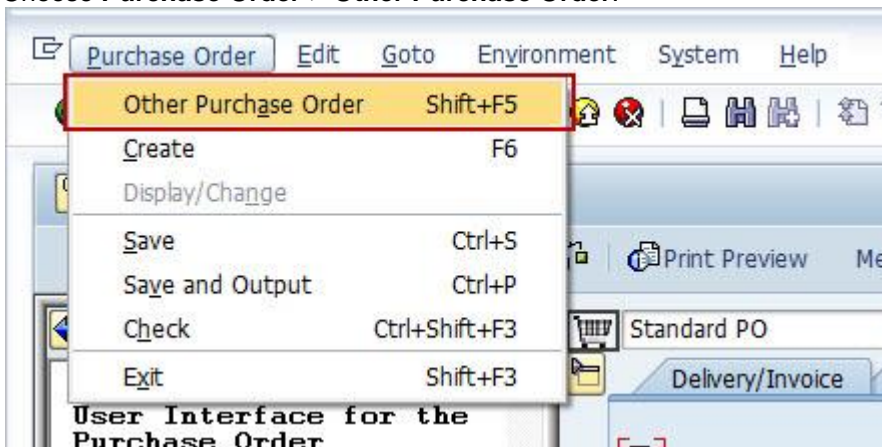
4.6.2 Read specific Purchase Order Item entry

1. In order to read a specific Purchase Order Item entry the SharePoint application should use the following URL
[http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderItemCollection\(POItem='00001',PONumber='4500017519'\)](http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderItemCollection(POItem='00001',PONumber='4500017519')) In this case, the details of Item '0001' of the Purchase Order 4500017519 is fetched.

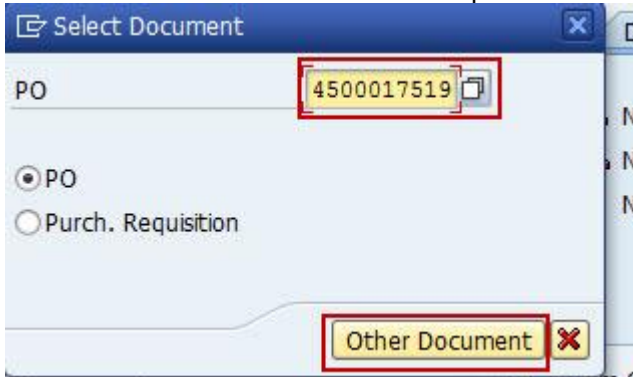
4.7 Update operation

4.7.1 Purchase Order Header Update

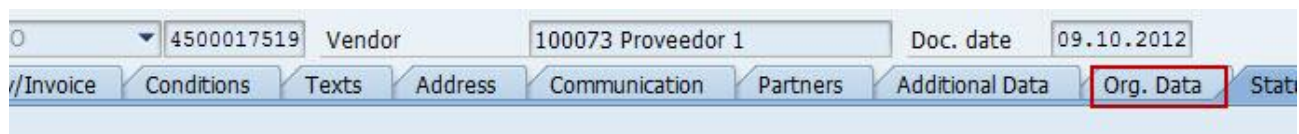
1. The SharePoint application can also be used to update an existing Purchase Order Header in the SAP system. In this case, the application should use the following URL
[http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderHeaderCollection\('4500017519'\)](http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderHeaderCollection('4500017519')) as a HTTP PUT call (Here the Purchase Order 4500017519 is being updated).
2. A sample request body for the Purchase Order Header update call is at [POHeaderUpdate.txt](#).
3. To check the updated Purchase Order go to transaction me21n in the SAP backend system.
4. Choose **Purchase Order-> Other Purchase Order**.



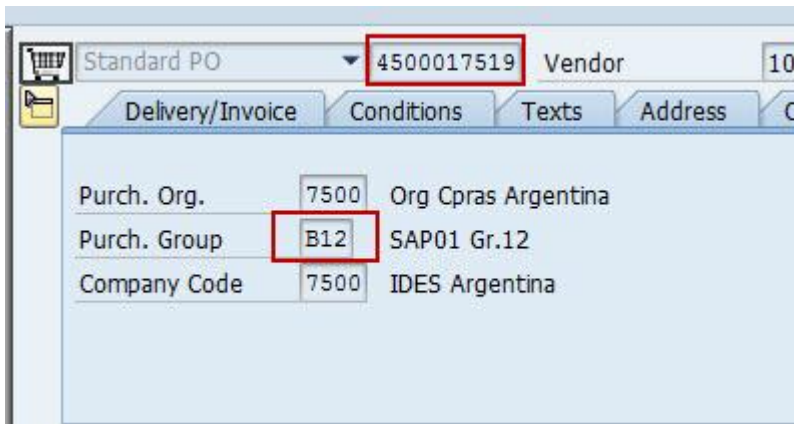
5. Enter the PO Number obtained from step 6 and click on Other Document.



6. This displays the Purchase Order.
7. Choose the **Org.Data** tab in the screen.

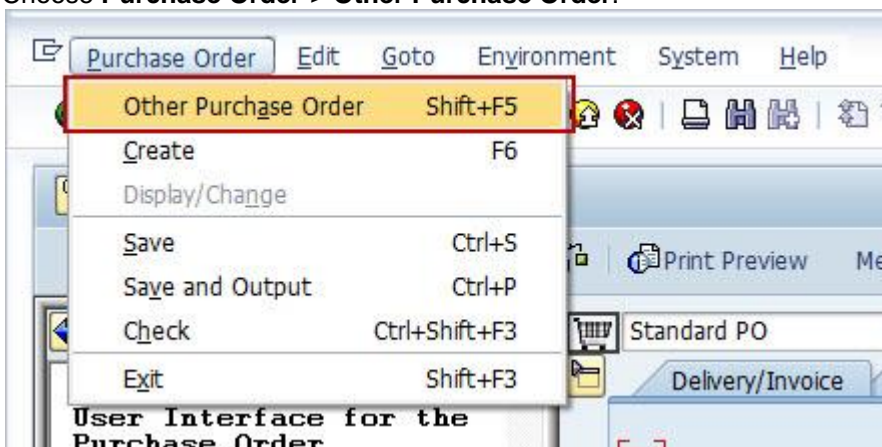


8. The updated Purchase Organization Group can be found here if we tried to update the organisation group for the Purchase Order from the SharePoint application.

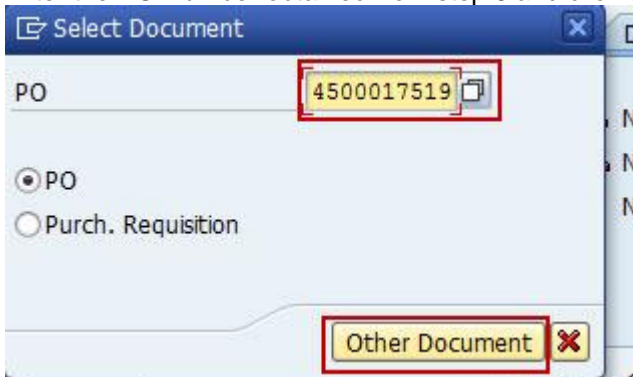


4.7.2 Purchase Order Item update

1. Similarly, the SharePoint application can also be used to update an existing Purchase Order Item in the SAP Business Suite system. In this case, the application should use the following URL [http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderItemCollection\(POItem='00001',PONumber='4500017519'\)](http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderItemCollection(POItem='00001',PONumber='4500017519')) as a HTTP PUT call (Here the item 00001 of the Purchase Order 4500017519 is being updated).
2. A sample request body for the Purchase Order Item update is at [POItemUpdate.txt](#).
3. To check the updated Purchase Order go to transaction **me21n** in the SAP backend system.
4. Choose **Purchase Order-> Other Purchase Order**.



5. Enter the PO Number obtained from step 6 and click on **Other Document**.



6. This displays the Purchase Order.
7. Choose the **Status** tab in the screen.

4500017519 Vendor 100073 Proveedor 1 Doc. date 09.10.2012

Conditions Texts Address Communication Partners Additional Data Org. Data **Status**

Ordered 10 PC 28,20 ARS

8. The updated Purchase Order Quantity can be found here.

Standard PO 4500017519 Vendor 100073 Proveedor 1

Delivery/Invoice Conditions Texts Address Communication

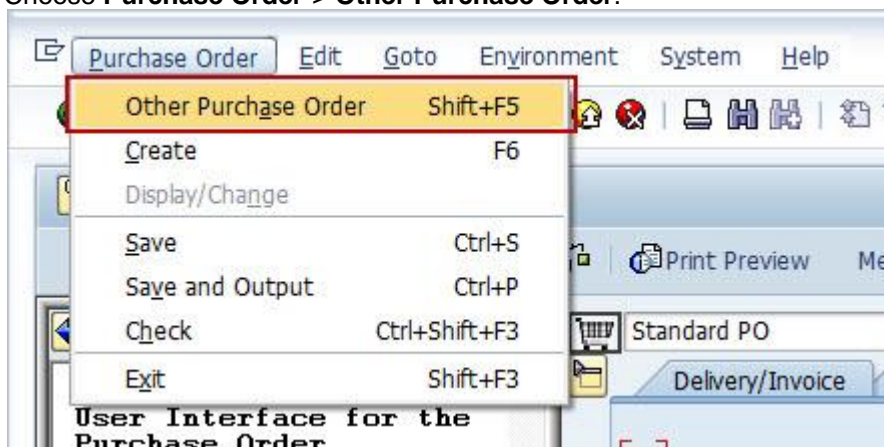
Active Ordered 10 E
 Not Yet Sent Delivered 0 E
 Not Delivered Still to deliv. 10 E
 Not Invoiced Invoiced 0 E
 Down paymts

S.	Itm	A	I	Material	Short Text	PO Quantity
	1			1108	2222	10

4.8 Delete operation

4.8.1 Purchase Order Item Delete

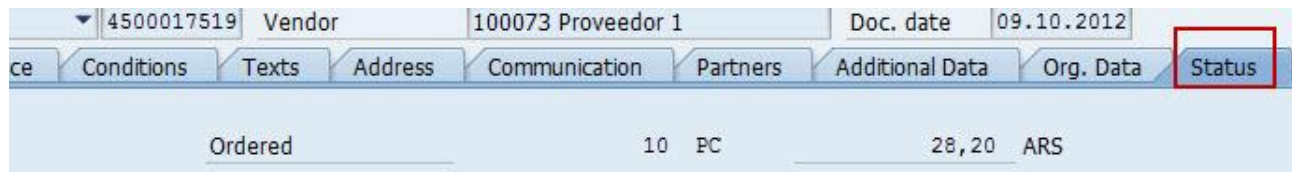
- Similarly, the application can be used to delete an already existing Purchase Order entry in the SAP Business Suite system. A DELETE HTTP call should be used in this case :
[http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderItemCollection\(POItem='00001',PONumber='4500017519'\)](http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderItemCollection(POItem='00001',PONumber='4500017519')) (The call will delete the Item 00001 of the Purchase Order 4500017519 in this case).
- To check the deleted Purchase Order Item, go to transaction **me21n** in the SAP backend system.
- Choose **Purchase Order-> Other Purchase Order**.



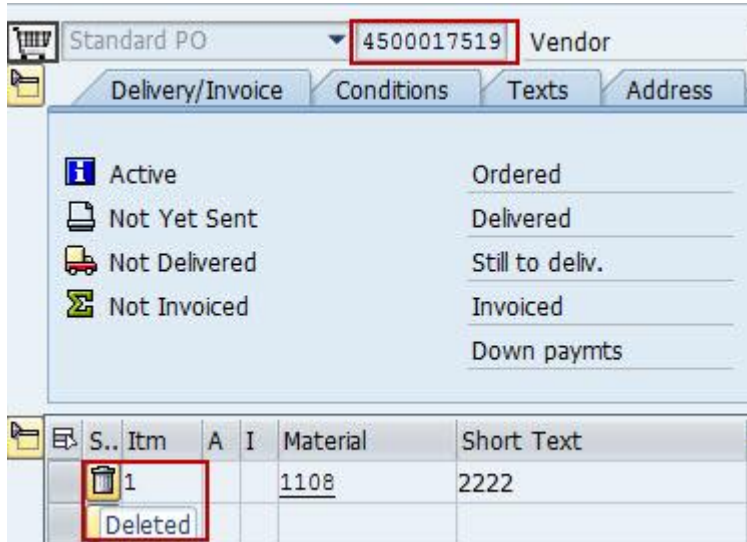
- Enter the PO Number obtained from step 6 and click on **Other Document**.



5. This displays the Purchase Order.
6. Choose the **Status** tab in the screen.



7. The information of the deleted Purchase Order item can be found here.



4.8.2 Purchase Order Header Delete

1. In order to delete a Purchase Order Header entry, the URL call to the SAP system should be similar to the following
[http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderHeaderCollection\('4500017519'\)](http://<host>:<port>/sap/opu/odata/sap/Z_PORDER_SRV/PurchaseOrderHeaderCollection('4500017519'))
 (This call will delete the Purchase Order Header 4500017519 in this case)

5. Glossary

Metadata Provider Class: A runtime object that is created when the OData service is generated. This class contains the generated metadata definition of the OData service.

Metadata Provider Extension Class: A runtime object that is created when the OData service is generated. This is an extension class of the Model Provider Class and additional code to refine the model has to be written in the metadata provider extension class.

Data Provider Class: A runtime object that is created when the OData service is generated. This is called during the runtime when the OData service is being executed.

Data Provider Extension Class: A runtime object that is created when the OData service is generated. This is an extension class for the Data Provider Class. For each of the entities, the methods corresponding to CRUD operations can be implemented in the data provider extension class.

Related Content

[How to Install and Configure Duet Enterprise 2.0](#)

[OData service in the SAP backend system for CRUDQ operations in Purchase Order scenario](#)

Video: [Developing a Purchase Order](#)

Copyright

© Copyright 2013 SAP AG. 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 AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG 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.