5 Ways of Creating an OData Service in SAP

By Sachin Artani

Sachin Artani

ODATA: The Open Data Protocol (OData) is REST based service protocol for simplified access to SAP data.

REST: **RE**presentational **S**tate **T**ransfer (REST) is not a kind of protocol, but an architectural style builds on top of HTTP defining below set of constraints –

- Client Server architecture
- Statelessness
- Cache-ability
- Layered system
- Uniform Interface
- Code on demand

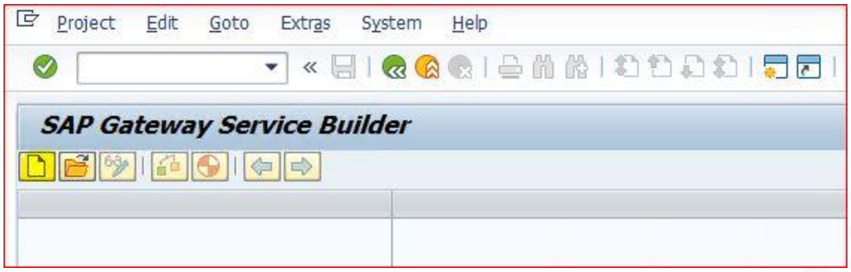
HTTP: It is a protocol used for communication, usually used to communicate with internet resources or any application with a web browser client.

Here are the different ways we create OData service in SAP –

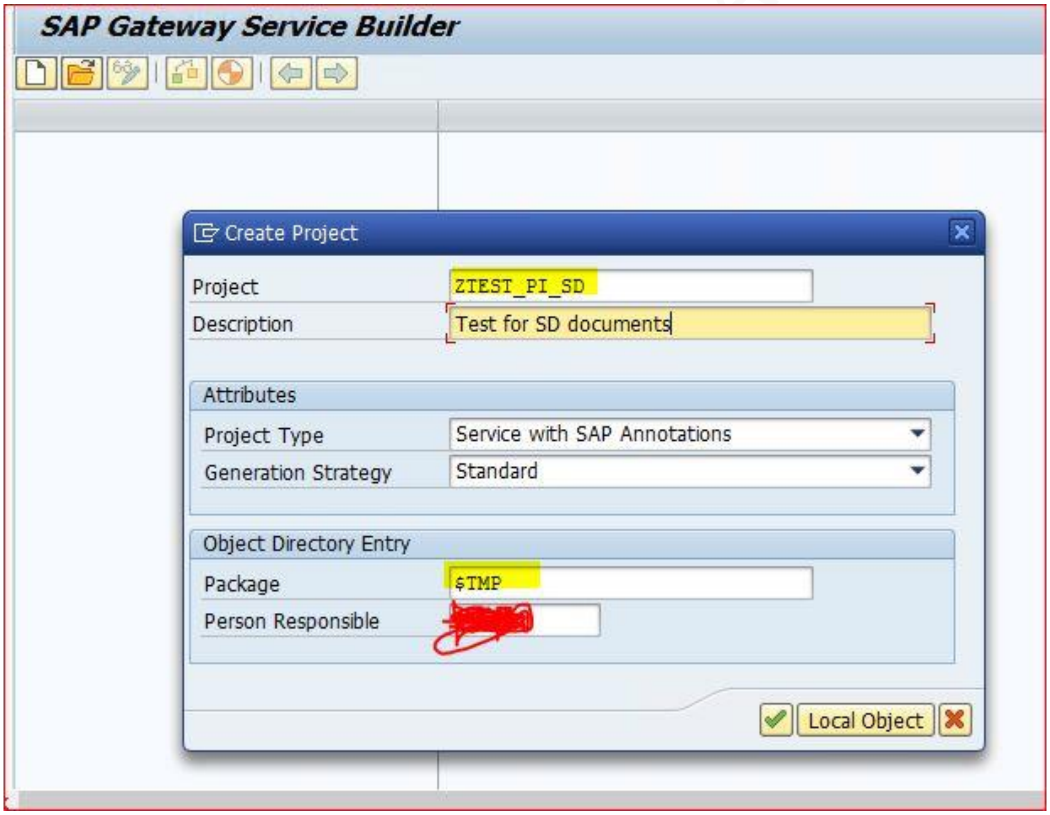
SAP Gateway Based OData Service –

Below steps concludes with the completion of OData service creation using the T-code: SEGW, including the generation of runtime artifacts, registration, and activation of the service. Additionally, it emphasizes the importance of redefining methods for header and item entity sets, loading metadata, and thoroughly testing entity sets with test data to ensure functionality. This is a classic SEGW based OData service creation way. Content is taken from a blog written by ***Laxmi Jibhe***.

Step 1. Go to transaction code - SEGW.



Click on Icon Create. A pop window will appear, Fill the details as per below mention in screen shot and click on check icon or enter.



Project: ZTEST_PI_SD

Description: Test for SD documents

Attributes

Project Type: Service with SAP Annotations

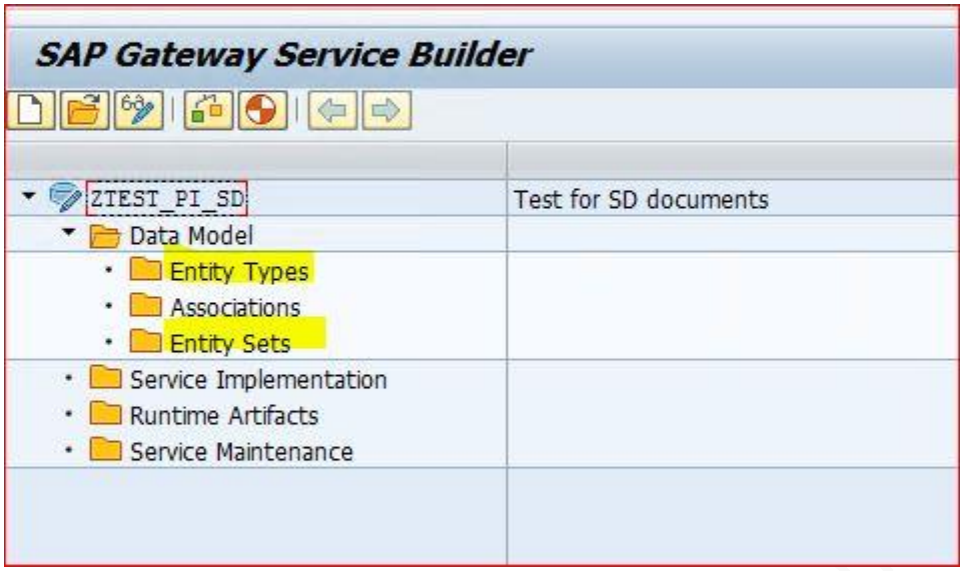
Generation Strategy: Standard

Object Directory Entry

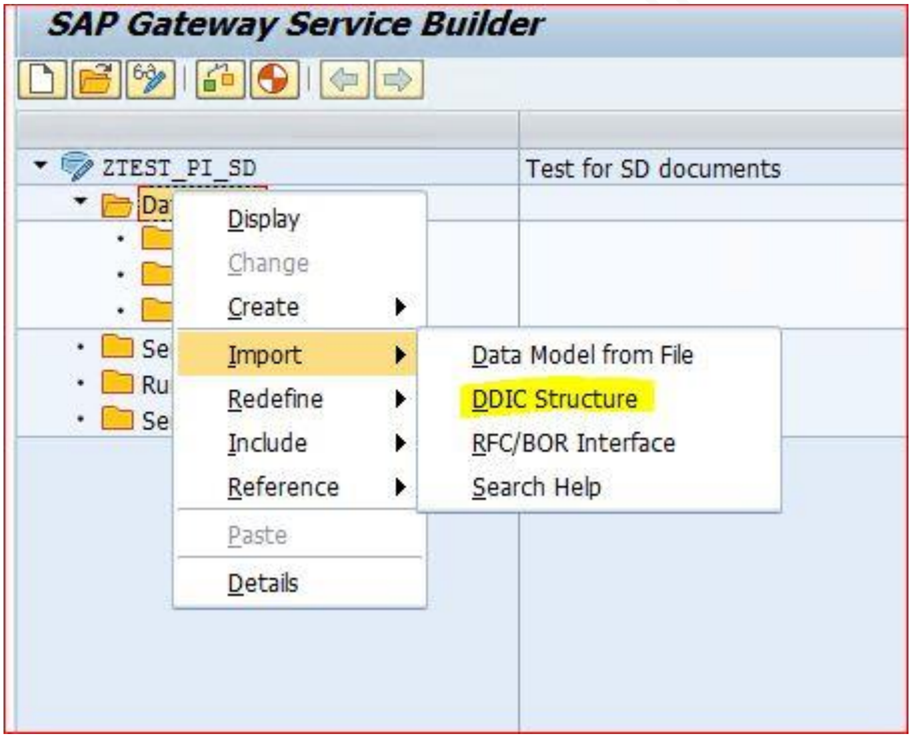
Package: $TMP

Person Responsible: [redacted]

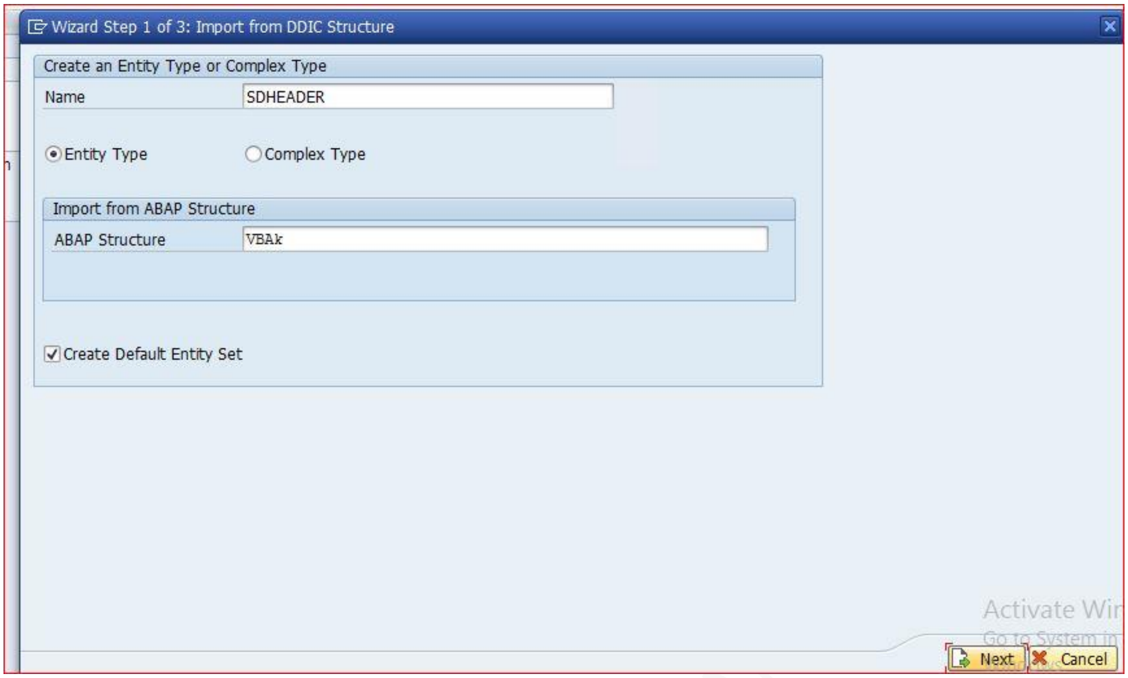
Below screen will appear where you can see below folder in project. In Folder data model, we can see three sub folders. Entity Type - it acts as work area, Entity Sets -It act as internal table and associations.



Now we are going to define structure of work area and internal table, Right click on Data model select import and select DDIC structure .



Here give the details of structure and structure name as per below screen below. Select radio button Entity type and click on check box entity set. fill ABAP structure as VBAK and click on next.



Wizard Step 1 of 3: Import from DDIC Structure

Create an Entity Type or Complex Type

Name: SDHEADER

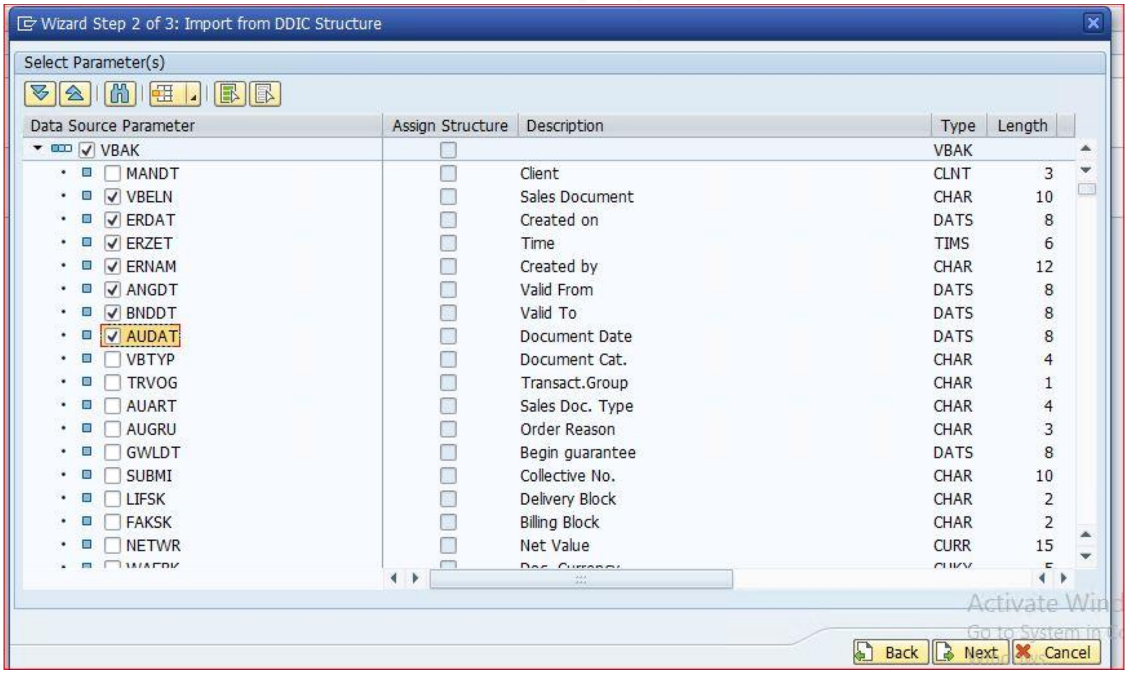
● Entity Type ○ Complex Type

Import from ABAP Structure

ABAP Structure: VBAk

☑ Create Default Entity Set

Now you will get the pop-up screen with VBAK table fields name. Select fields for your structure and click on next.

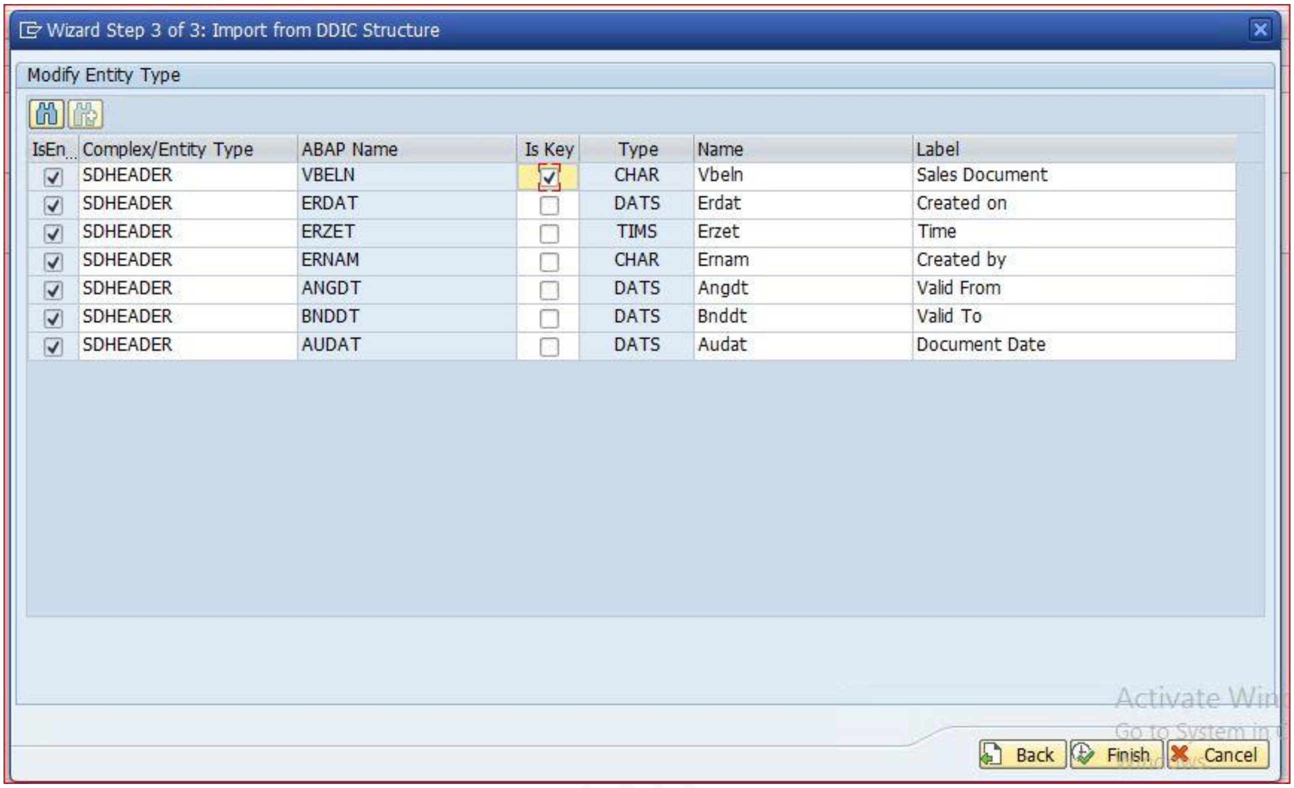


Wizard Step 2 of 3: Import from DDIC Structure

Select Parameter(s)

Data Source Parameter	Assign Structure	Description	Type	Length
☑ VBAK	□		VBAK	
□ MANDT	□	Client	CLNT	3
☑ VBELN	□	Sales Document	CHAR	10
☑ ERDAT	□	Created on	DATS	8
☑ ERZET	□	Time	TIMS	6
☑ ERNAM	□	Created by	CHAR	12
☑ ANGDT	□	Valid From	DATS	8
☑ BNDDT	□	Valid To	DATS	8
☑ AUDAT	□	Document Date	DATS	8
□ VBTYP	□	Document Cat.	CHAR	4
□ TRVOG	□	Transact.Group	CHAR	1
□ AUART	□	Sales Doc. Type	CHAR	4
□ AUGRU	□	Order Reason	CHAR	3
□ GWLDT	□	Begin guarantee	DATS	8
□ SUBMI	□	Collective No.	CHAR	10
□ LIFSK	□	Delivery Block	CHAR	2
□ FAKSK	□	Billing Block	CHAR	2
□ NETWR	□	Net Value	CURR	15

Another window will appear where we must select the key field. tick on VBELN as key field and click on finish. You may get warning message; it can be totally ignored.

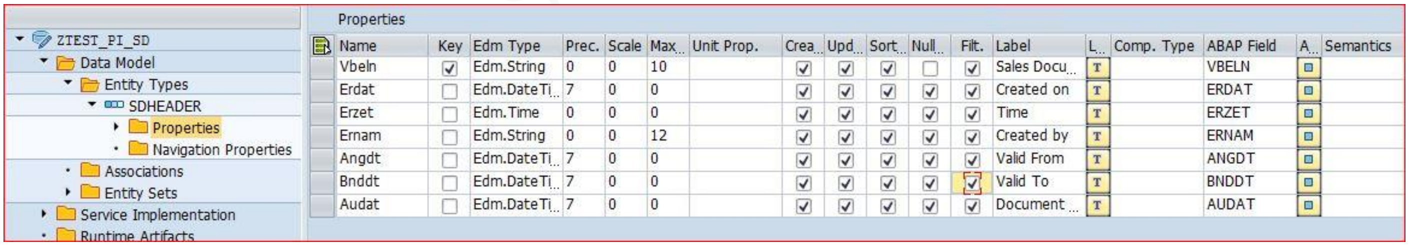


Wizard Step 3 of 3: Import from DDIC Structure

Modify Entity Type

IsEn...	Complex/Entity Type	ABAP Name	Is Key	Type	Name	Label
☑	SDHEADER	VBELN	☑	CHAR	Vbeln	Sales Document
☑	SDHEADER	ERDAT	□	DATS	Erdat	Created on
☑	SDHEADER	ERZET	□	TIMS	Erzet	Time
☑	SDHEADER	ERNAM	□	CHAR	Ernam	Created by
☑	SDHEADER	ANGDT	□	DATS	Angdt	Valid From
☑	SDHEADER	BNDDT	□	DATS	Bnddt	Valid To
☑	SDHEADER	AUDAT	□	DATS	Audat	Document Date

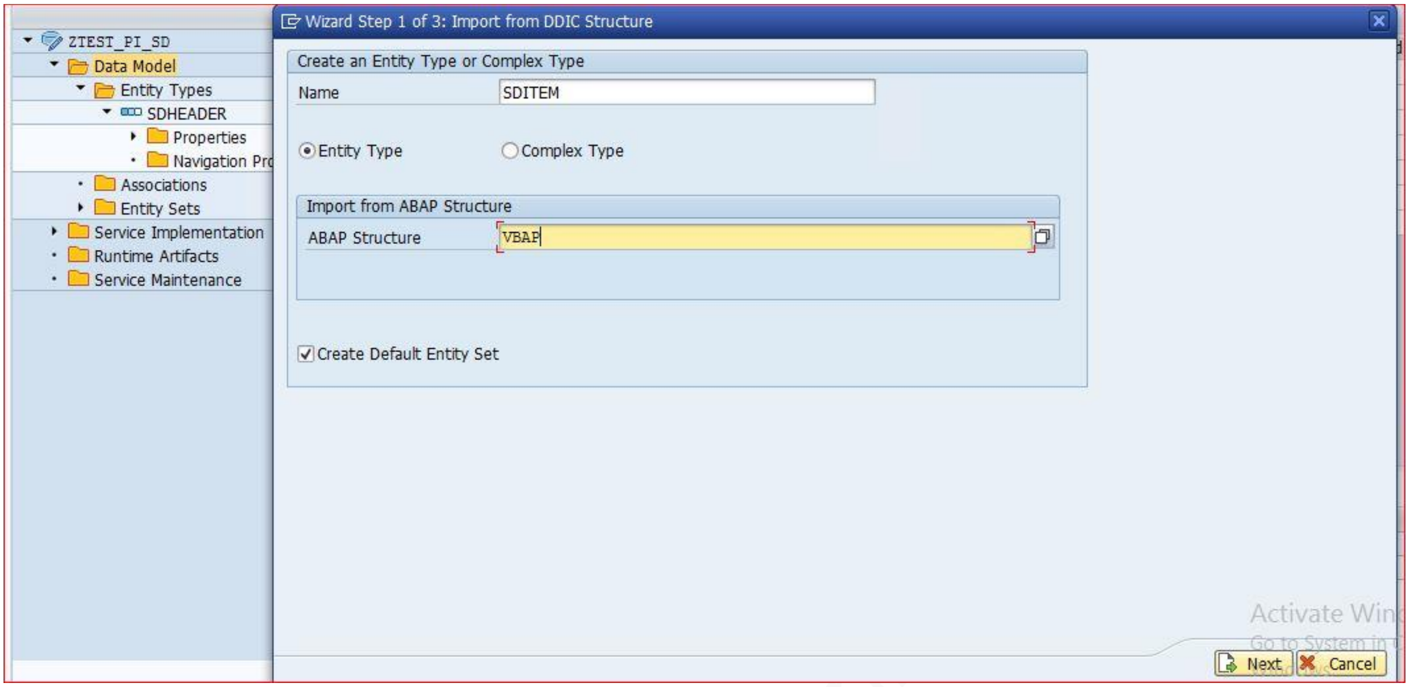
Now expand folder entity type and fill the below parameters in properties, please don't tick Null check box for VBELN as it is key field.



Properties

Name	Key	Edm Type	Prec.	Scale	Max...	Unit Prop.	Crea...	Upd...	Sort...	Null...	Filt.	Label	L...	Comp. Type	ABAP Field	A...	Semantics
Vbeln	☑	Edm.String	0	0	10		☑	☑	☑	□	☑	Sales Docu...	T		VBELN		
Erdat	□	Edm.DateTi...	7	0	0		☑	☑	☑	☑	☑	Created on	T		ERDAT		
Erzet	□	Edm.Time	0	0	0		☑	☑	☑	☑	☑	Time	T		ERZET		
Ernam	□	Edm.String	0	0	12		☑	☑	☑	☑	☑	Created by	T		ERNAM		
Angdt	□	Edm.DateTi...	7	0	0		☑	☑	☑	☑	☑	Valid From	T		ANGDT		
Bnddt	□	Edm.DateTi...	7	0	0		☑	☑	☑	☑	☑	Valid To	T		BNDDT		
Audat	□	Edm.DateTi...	7	0	0		☑	☑	☑	☑	☑	Document ...	T		AUDAT		

Similarly Create Structure for SD Item following same steps



- ZTEST_PI_SD
  - Data Model
    - Entity Types
      - SDHEADER
        - Properties
        - Navigation Pro
    - Associations
    - Entity Sets
  - Service Implementation
  - Runtime Artifacts
  - Service Maintenance

Wizard Step 1 of 3: Import from DDIC Structure

Create an Entity Type or Complex Type

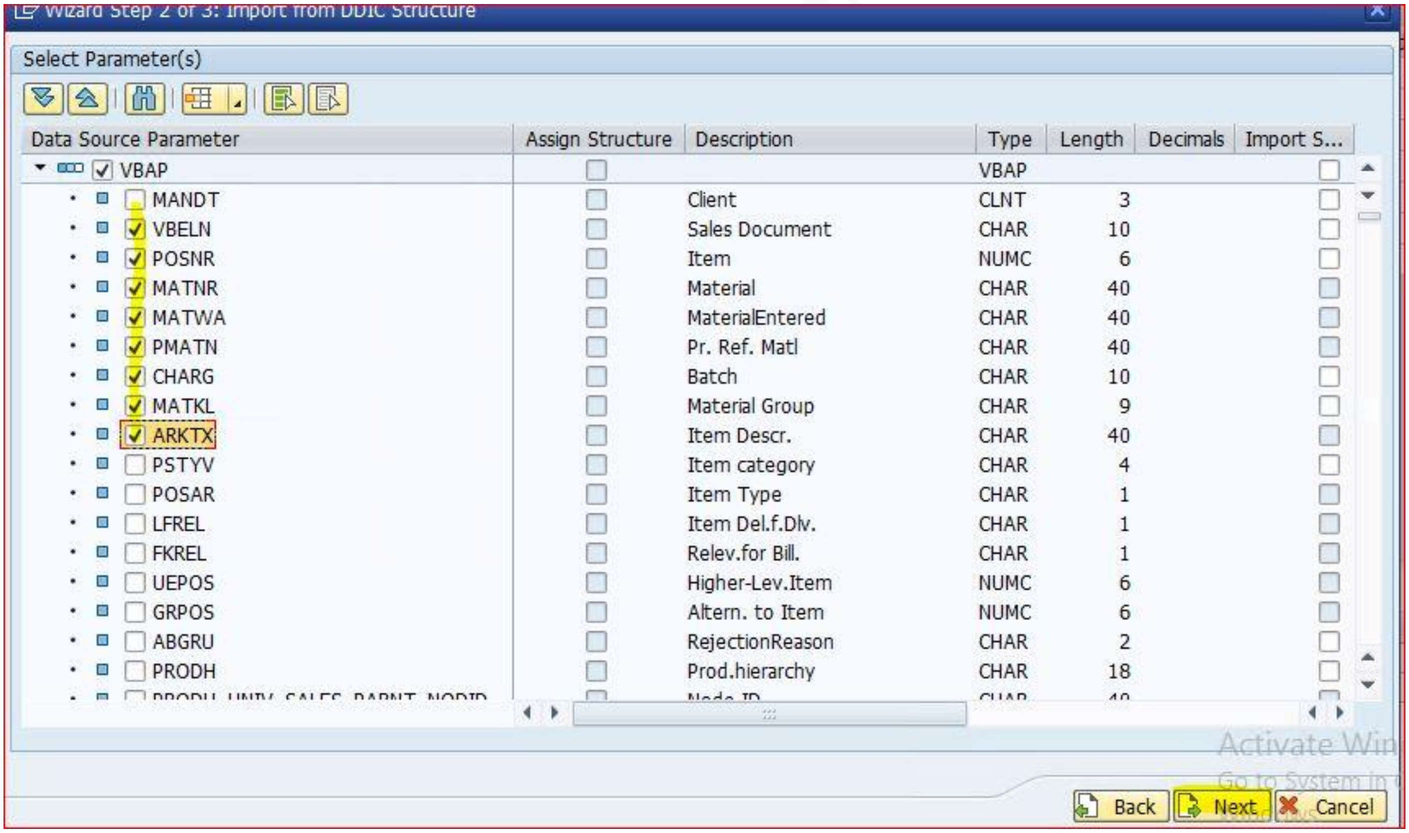
Name: SDITEM

● Entity Type ○ Complex Type

Import from ABAP Structure

ABAP Structure: VBAP

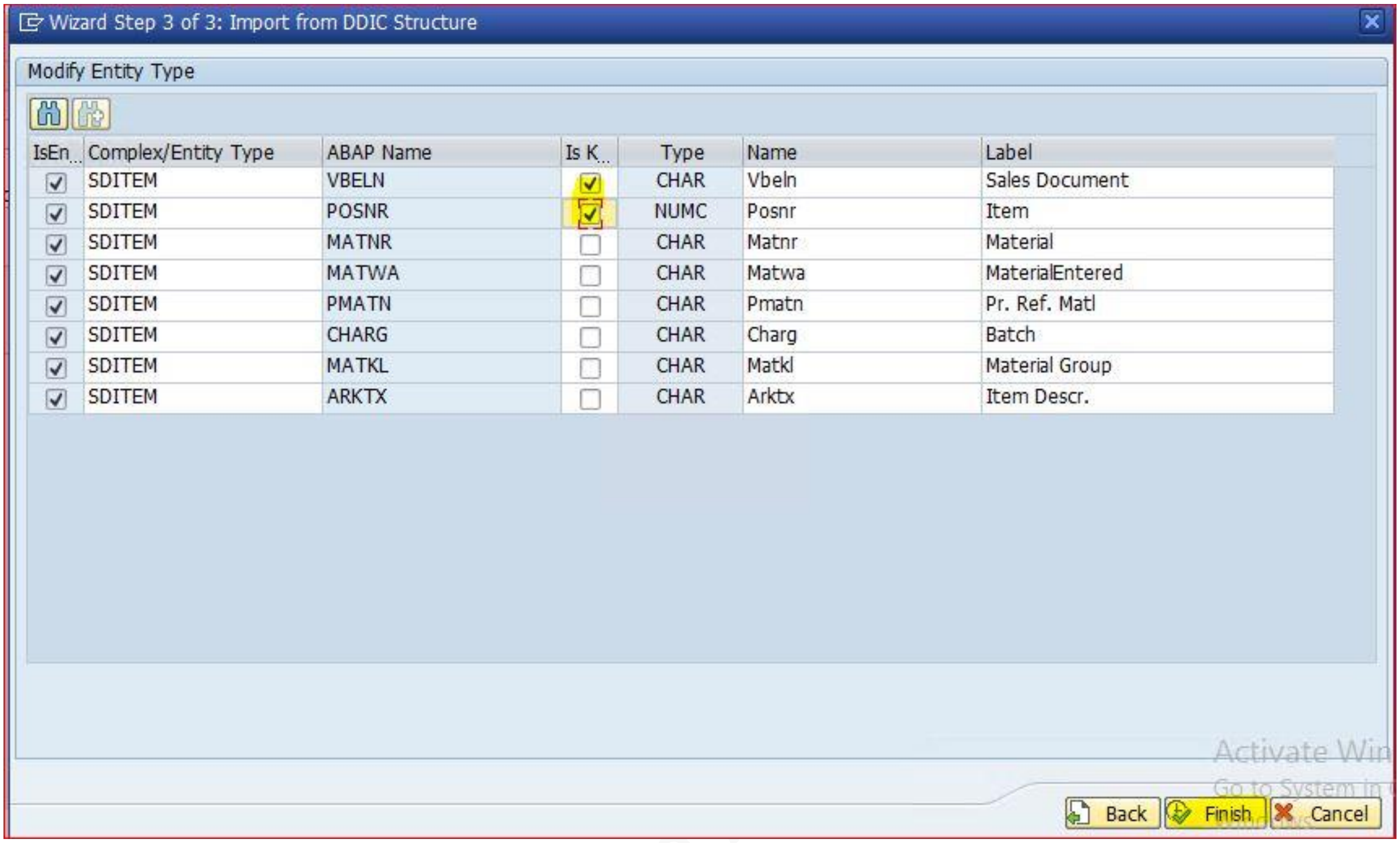
☑ Create Default Entity Set



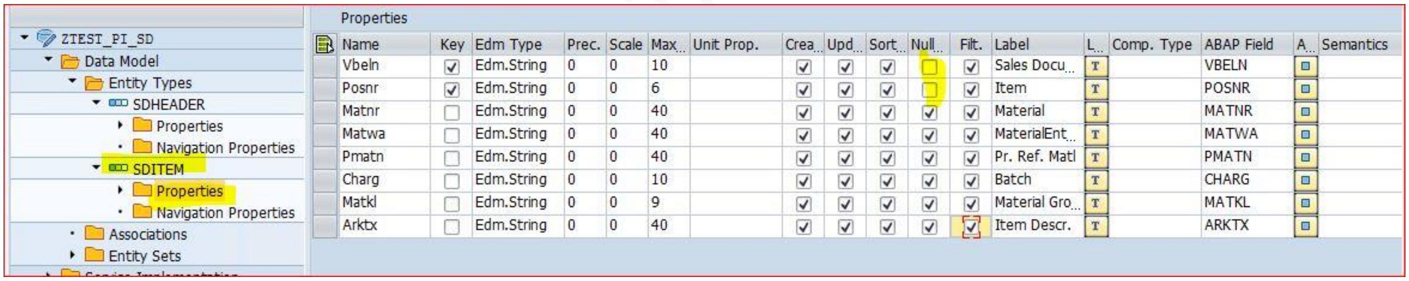
Wizard Step 2 of 3: Import from DDIC Structure

Select Parameter(s)

Data Source Parameter	Assign Structure	Description	Type	Length	Decimals	Import S...
☑ VBAP	□		VBAP			□
□ MANDT	□	Client	CLNT	3		□
☑ VBELN	□	Sales Document	CHAR	10		□
☑ POSNR	□	Item	NUMC	6		□
☑ MATNR	□	Material	CHAR	40		□
☑ MATWA	□	MaterialEntered	CHAR	40		□
☑ PMATN	□	Pr. Ref. Matl	CHAR	40		□
☑ CHARG	□	Batch	CHAR	10		□
☑ MATKL	□	Material Group	CHAR	9		□
☑ ARKTX	□	Item Descr.	CHAR	40		□
□ PSTYV	□	Item category	CHAR	4		□
□ POSAR	□	Item Type	CHAR	1		□
□ LFREL	□	Item Del.f.Dlv.	CHAR	1		□
□ FKREL	□	Relev.for Bill.	CHAR	1		□
□ UEPOS	□	Higher-Lev.Item	NUMC	6		□
□ GRPOS	□	Altern. to Item	NUMC	6		□
□ ABGRU	□	RejectionReason	CHAR	2		□
□ PRODH	□	Prod.hierarchy	CHAR	18		□



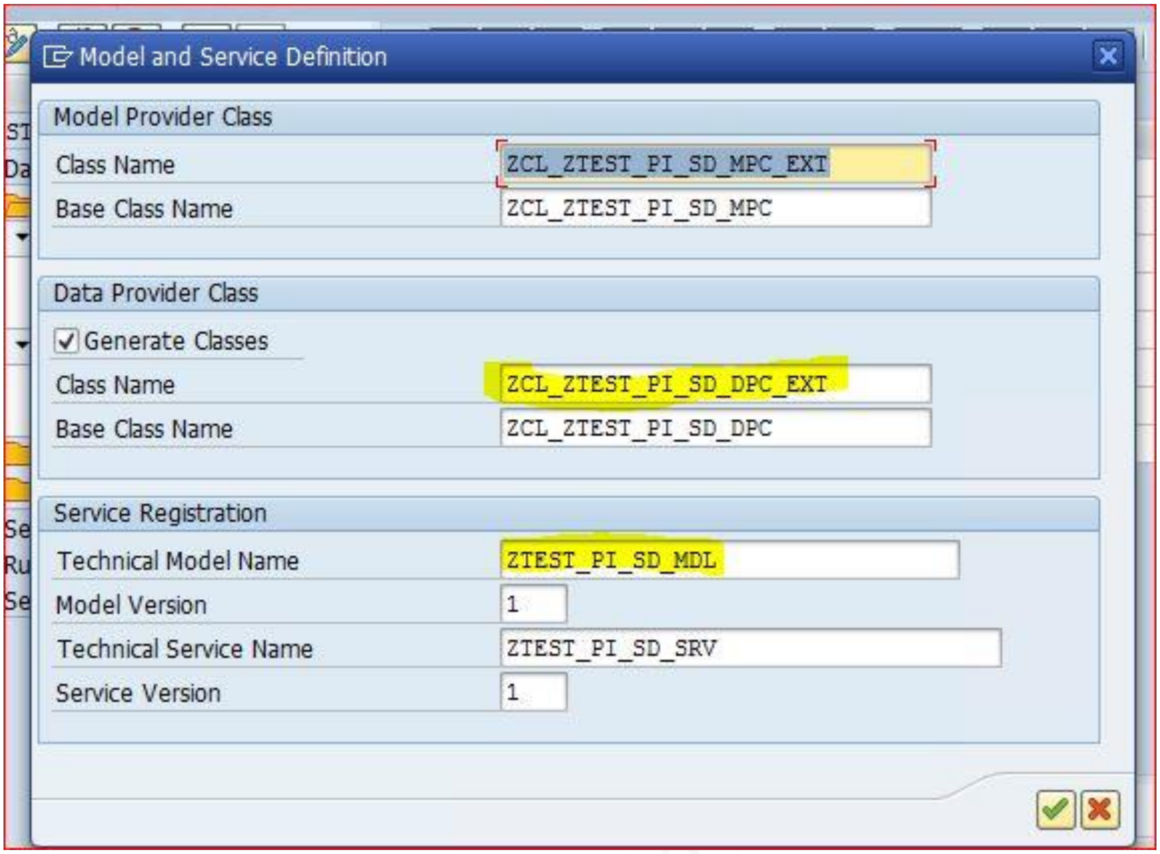
IsEn...	Complex/Entity Type	ABAP Name	Is K...	Type	Name	Label
☑	SDITEM	VBELN	☑	CHAR	Vbeln	Sales Document
☑	SDITEM	POSNR	☑	NUMC	Posnr	Item
☑	SDITEM	MATNR	□	CHAR	Matnr	Material
☑	SDITEM	MATWA	□	CHAR	Matwa	MaterialEntered
☑	SDITEM	PMATN	□	CHAR	Pmatn	Pr. Ref. Matl
☑	SDITEM	CHARG	□	CHAR	Charg	Batch
☑	SDITEM	MATKL	□	CHAR	Matkl	Material Group
☑	SDITEM	ARKTX	□	CHAR	Arktx	Item Descr.



Name	Key	Edm Type	Prec.	Scale	Max...	Unit Prop.	Crea...	Upd...	Sort...	Null...	Filt.	Label	L...	Comp. Type	ABAP Field	A...	Semantics
Vbeln	☑	Edm.String	0	0	10		☑	☑	☑	□	☑	Sales Docu...	T		VBELN	▣	
Posnr	☑	Edm.String	0	0	6		☑	☑	☑	□	☑	Item	T		POSNR	▣	
Matnr	□	Edm.String	0	0	40		☑	☑	☑	☑	☑	Material	T		MATNR	▣	
Matwa	□	Edm.String	0	0	40		☑	☑	☑	☑	☑	MaterialEnt...	T		MATWA	▣	
Pmatn	□	Edm.String	0	0	40		☑	☑	☑	☑	☑	Pr. Ref. Matl	T		PMATN	▣	
Charg	□	Edm.String	0	0	10		☑	☑	☑	☑	☑	Batch	T		CHARG	▣	
Matkl	□	Edm.String	0	0	9		☑	☑	☑	☑	☑	Material Gro...	T		MATKL	▣	
Arktx	□	Edm.String	0	0	40		☑	☑	☑	☑	☑	Item Descr.	T		ARKTX	▣	

Click on Save button at the top of menu bar.

Now click on generate Icon, A pop up window will appear with class details, click on tick icon, and proceed further. It will ask for package, give the details, and proceed further. These are also knowns as runtime artifacts.



Model Provider Class

Class Name	ZCL_ZTEST_PI_SD_MPC_EXT
Base Class Name	ZCL_ZTEST_PI_SD_MPC

Data Provider Class

☑ Generate Classes

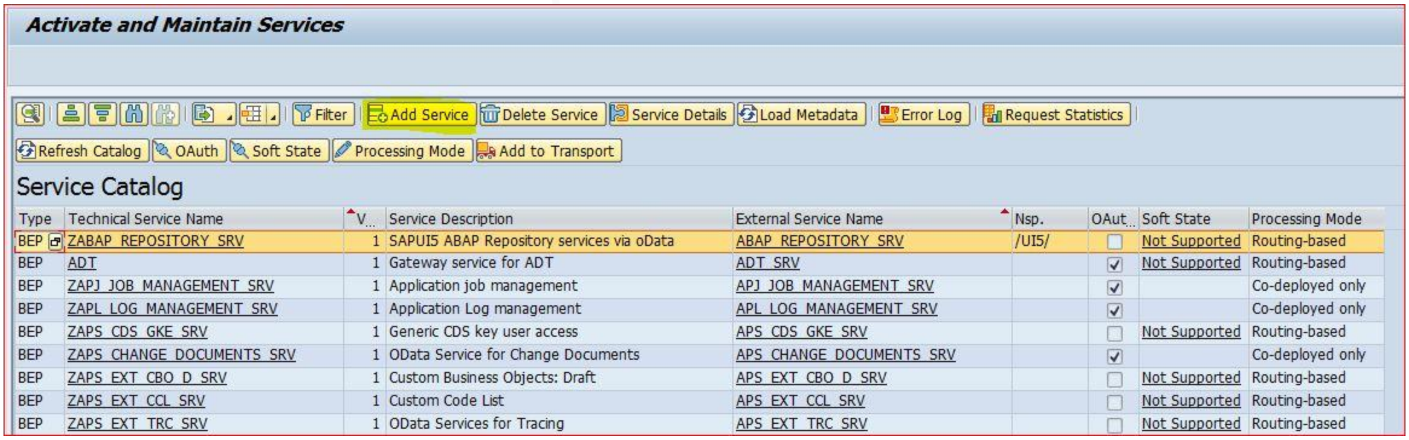
Class Name	ZCL_ZTEST_PI_SD_DPC_EXT
Base Class Name	ZCL_ZTEST_PI_SD_DPC

Service Registration

Technical Model Name	ZTEST_PI_SD_MDL
Model Version	1
Technical Service Name	ZTEST_PI_SD_SRV
Service Version	1

Sachin

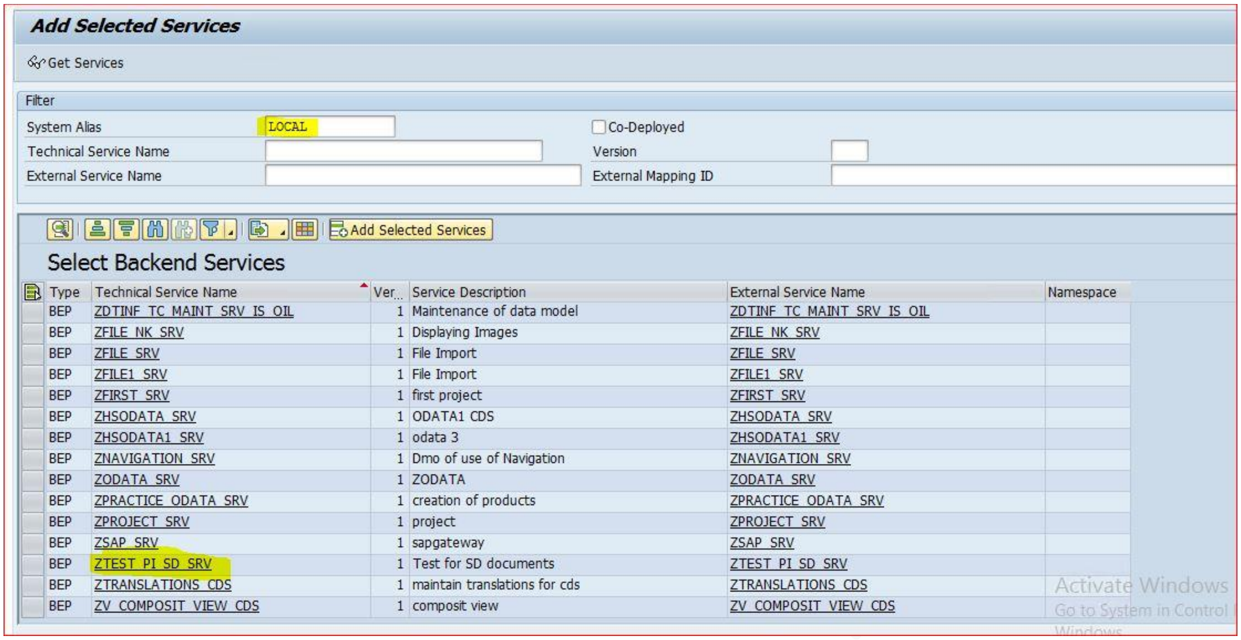
Step 2. Go to transaction code /IWFND/MAINT_SERVICE and click on push button - Add services.



Service Catalog

Type	Technical Service Name	V...	Service Description	External Service Name	Nsp.	OAut...	Soft State	Processing Mode
BEP	ZABAP_REPOSITORY_SRV	1	SAPUI5 ABAP Repository services via oData	ABAP_REPOSITORY_SRV	/UI5/	☐	Not Supported	Routing-based
BEP	ADT	1	Gateway service for ADT	ADT_SRV		☑	Not Supported	Routing-based
BEP	ZAPJ_JOB_MANAGEMENT_SRV	1	Application job management	APJ_JOB_MANAGEMENT_SRV		☑		Co-deployed only
BEP	ZAPL_LOG_MANAGEMENT_SRV	1	Application Log management	APL_LOG_MANAGEMENT_SRV		☑		Co-deployed only
BEP	ZAPS_CDS_GKE_SRV	1	Generic CDS key user access	APS_CDS_GKE_SRV		☐	Not Supported	Routing-based
BEP	ZAPS_CHANGE_DOCUMENTS_SRV	1	OData Service for Change Documents	APS_CHANGE_DOCUMENTS_SRV		☑		Co-deployed only
BEP	ZAPS_EXT_CBO_D_SRV	1	Custom Business Objects: Draft	APS_EXT_CBO_D_SRV		☐	Not Supported	Routing-based
BEP	ZAPS_EXT_CCL_SRV	1	Custom Code List	APS_EXT_CCL_SRV		☐	Not Supported	Routing-based
BEP	ZAPS_EXT_TRC_SRV	1	OData Services for Tracing	APS_EXT_TRC_SRV		☐	Not Supported	Routing-based

Next screen will appear, give the alias name, and execute, Search for your project and double click on your project name.



Add Selected Services

Get Services

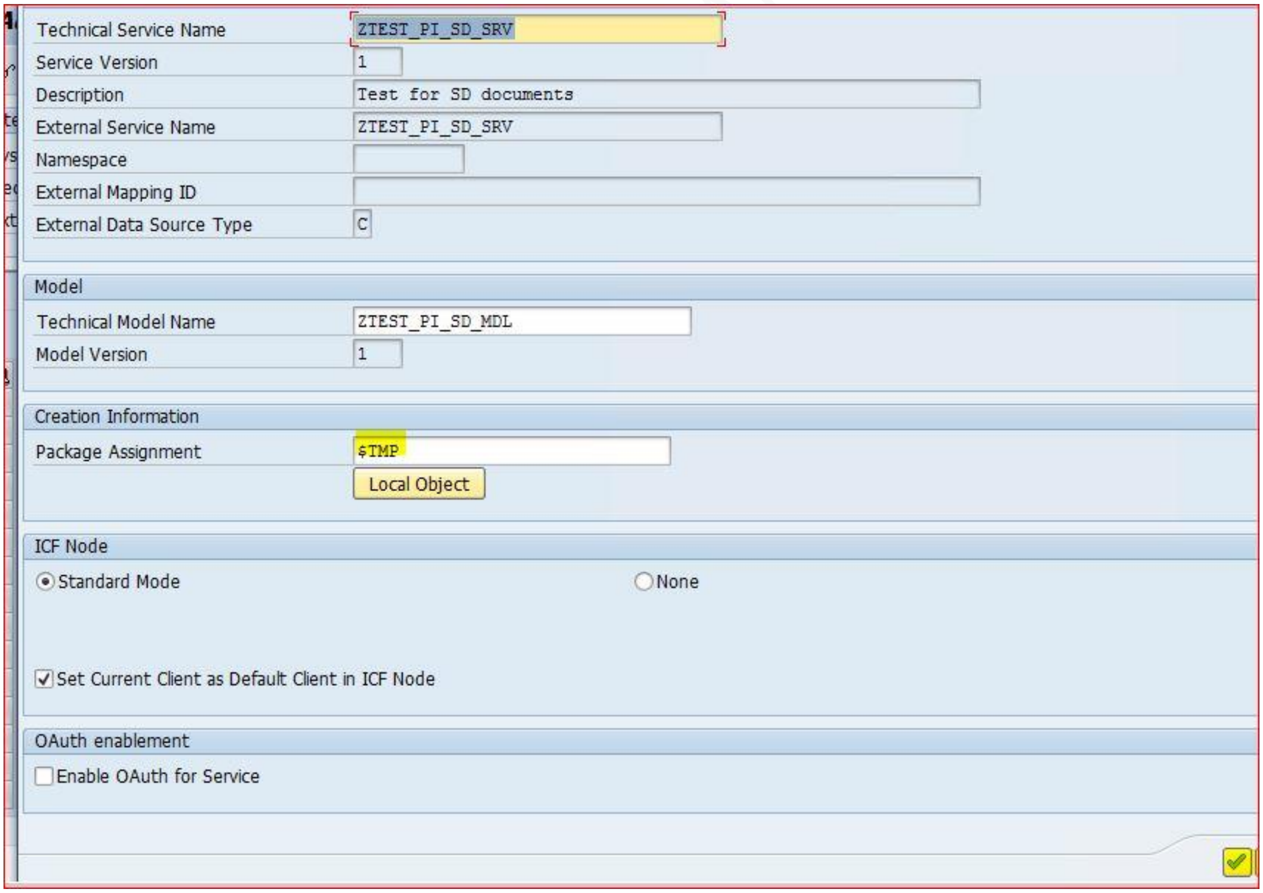
Filter

System Alias	LOCAL	□ Co-Deployed	
Technical Service Name		Version	
External Service Name		External Mapping ID	

Select Backend Services

Type	Technical Service Name	Ver...	Service Description	External Service Name	Namespace
BEP	ZDTINF_TC_MAINT_SRV_IS_OIL	1	Maintenance of data model	ZDTINF_TC_MAINT_SRV_IS_OIL	
BEP	ZFILE_NK_SRV	1	Displaying Images	ZFILE_NK_SRV	
BEP	ZFILE_SRV	1	File Import	ZFILE_SRV	
BEP	ZFILE1_SRV	1	File Import	ZFILE1_SRV	
BEP	ZFIRST_SRV	1	first project	ZFIRST_SRV	
BEP	ZHSODATA_SRV	1	ODATA1 CDS	ZHSODATA_SRV	
BEP	ZHSODATA1_SRV	1	odata 3	ZHSODATA1_SRV	
BEP	ZNAVIGATION_SRV	1	Dmo of use of Navigation	ZNAVIGATION_SRV	
BEP	ZODATA_SRV	1	ZODATA	ZODATA_SRV	
BEP	ZPRACTICE_ODATA_SRV	1	creation of products	ZPRACTICE_ODATA_SRV	
BEP	ZPROJECT_SRV	1	project	ZPROJECT_SRV	
BEP	ZSAP_SRV	1	sapgateway	ZSAP_SRV	
BEP	ZTEST_PI_SD_SRV	1	Test for SD documents	ZTEST_PI_SD_SRV	
BEP	ZTRANSLATIONS_CDS	1	maintain translations for cds	ZTRANSLATIONS_CDS	
BEP	ZV_COMPOSIT_VIEW_CDS	1	composit view	ZV_COMPOSIT_VIEW_CDS	

A pop-up window will appear, enter the package details, and click on tick icon. An Information message will be shown where it will confirm about the service is created and metadata loaded successfully.



Technical Service Name	ZTEST_PI_SD_SRV
Service Version	1
Description	Test for SD documents
External Service Name	ZTEST_PI_SD_SRV
Namespace	
External Mapping ID	
External Data Source Type	C

Model

Technical Model Name	ZTEST_PI_SD_MDL
Model Version	1

Creation Information

Package Assignment	$TMP

ICF Node

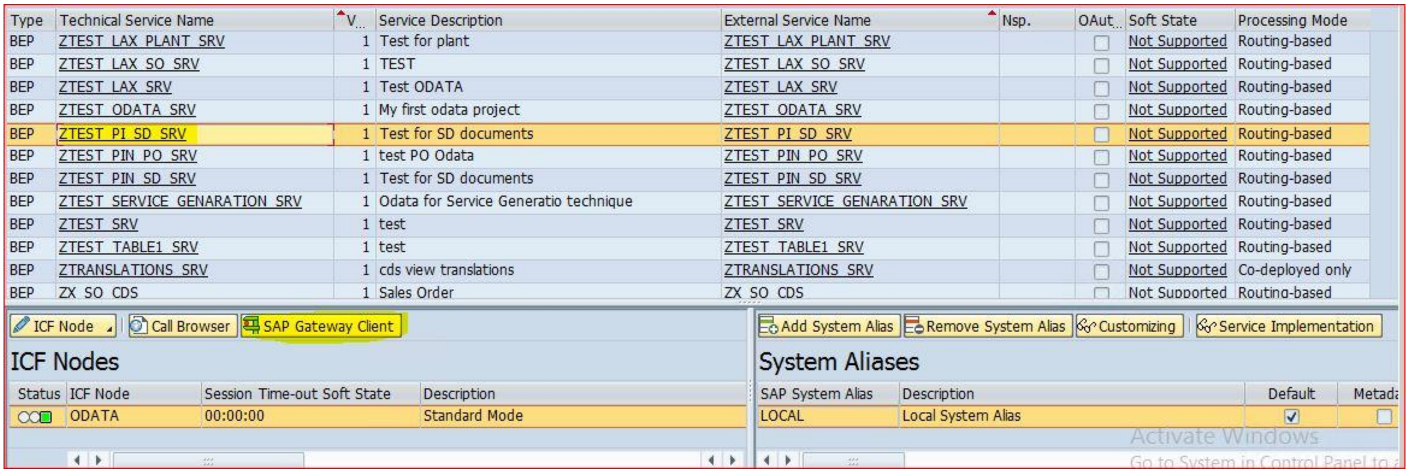
● Standard Mode ○ None

☑ Set Current Client as Default Client in ICF Node

OAuth enablement

□ Enable OAuth for Service

Now click back and go to main screen of transaction /IWFND/MAINT_SERVICE and find your service. Click on SAP Gateway Client.



Type	Technical Service Name	V..	Service Description	External Service Name	Nsp.	OAut..	Soft State	Processing Mode
BEP	ZTEST_LAX_PLANT_SRV	1	Test for plant	ZTEST_LAX_PLANT_SRV		□	Not Supported	Routing-based
BEP	ZTEST_LAX_SO_SRV	1	TEST	ZTEST_LAX_SO_SRV		□	Not Supported	Routing-based
BEP	ZTEST_LAX_SRV	1	Test ODATA	ZTEST_LAX_SRV		□	Not Supported	Routing-based
BEP	ZTEST_ODATA_SRV	1	My first odata project	ZTEST_ODATA_SRV		□	Not Supported	Routing-based
BEP	ZTEST_PI_SD_SRV	1	Test for SD documents	ZTEST_PI_SD_SRV		□	Not Supported	Routing-based
BEP	ZTEST_PIN_PO_SRV	1	test PO Odata	ZTEST_PIN_PO_SRV		□	Not Supported	Routing-based
BEP	ZTEST_PIN_SD_SRV	1	Test for SD documents	ZTEST_PIN_SD_SRV		□	Not Supported	Routing-based
BEP	ZTEST_SERVICE_GENARATION_SRV	1	Odata for Service Generatio technique	ZTEST_SERVICE_GENARATION_SRV		□	Not Supported	Routing-based
BEP	ZTEST_SRV	1	test	ZTEST_SRV		□	Not Supported	Routing-based
BEP	ZTEST_TABLE1_SRV	1	test	ZTEST_TABLE1_SRV		□	Not Supported	Routing-based
BEP	ZTRANSLATIONS_SRV	1	cds view translations	ZTRANSLATIONS_SRV		□	Not Supported	Co-deployed only
BEP	ZX_SO_CDS	1	Sales Order	ZX_SO_CDS		□	Not Supported	Routing-based

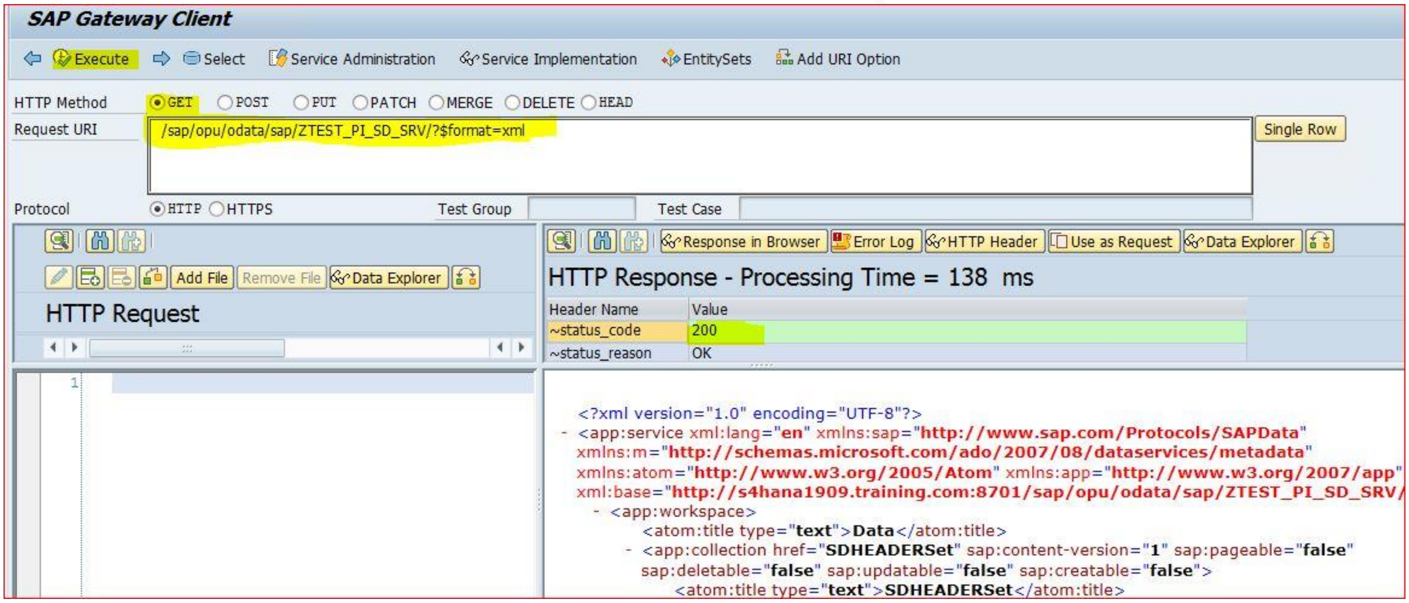
ICF Nodes

Status	ICF Node	Session Time-out Soft State	Description
○○■	ODATA	00:00:00	Standard Mode

System Aliases

SAP System Alias	Description	Default	Metad
LOCAL	Local System Alias	☑	□

A new screen will come, Execute the transaction, and check the response.



HTTP Method ● GET ○ POST ○ PUT ○ PATCH ○ MERGE ○ DELETE ○ HEAD

Request URI /sap/opu/odata/sap/ZTEST_PI_SD_SRV/?$format=xml

Protocol ● HTTP ○ HTTPS Test Group Test Case

HTTP Response - Processing Time = 138 ms

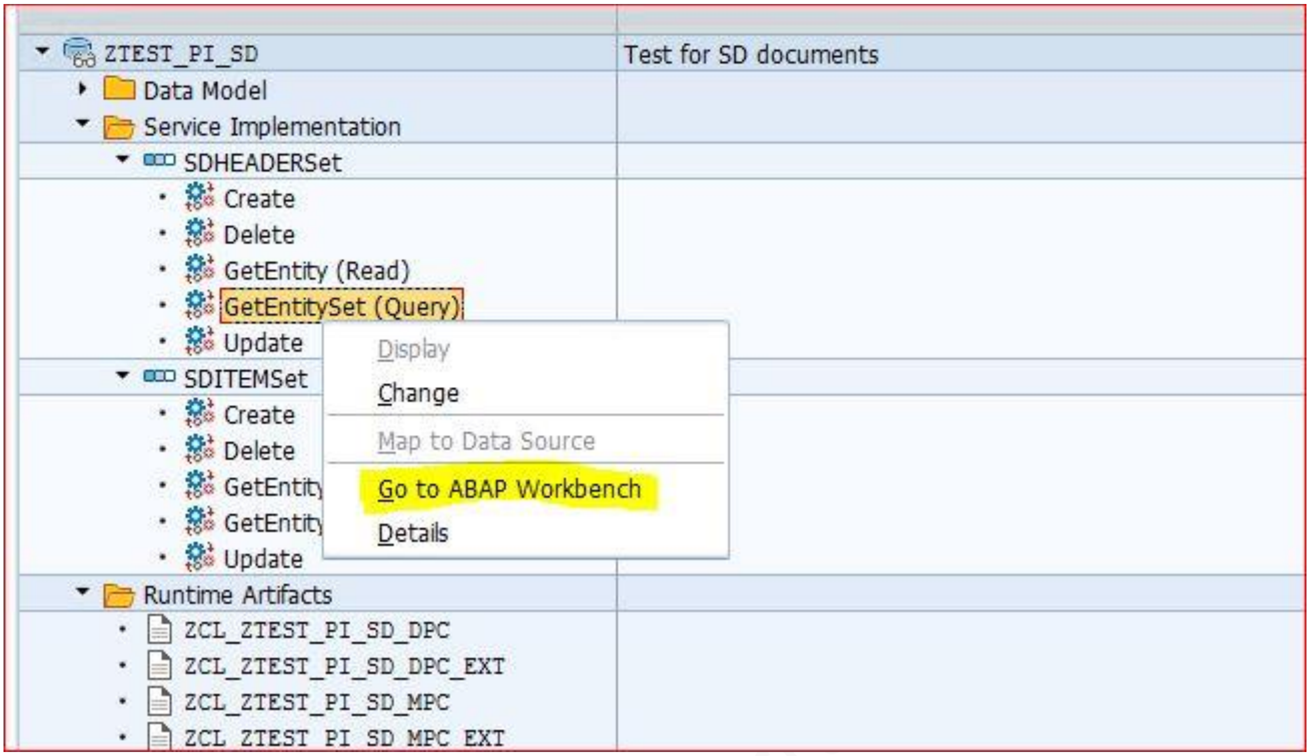
Header Name	Value
~status_code	200
~status_reason	OK

```
  <?xml version="1.0" encoding="UTF-8"?>
- <app:service xml:lang="en" xmlns:sap="http://www.sap.com/Protocols/SAPData"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:atom="http://www.w3.org/2005/Atom" xmlns:app="http://www.w3.org/2007/app"
  xml:base="http://s4hana1909.training.com:8701/sap/opu/odata/sap/ZTEST_PI_SD_SRV/
  - <app:workspace>
        <atom:title type="text">Data</atom:title>
      - <app:collection href="SDHEADERSet" sap:content-version="1" sap:pageable="false"
        sap:deletable="false" sap:updatable="false" sap:creatable="false">
            <atom:title type="text">SDHEADERSet</atom:title>
```

Status code 200 means that the OData service we created works well and now, we will write code to get data.

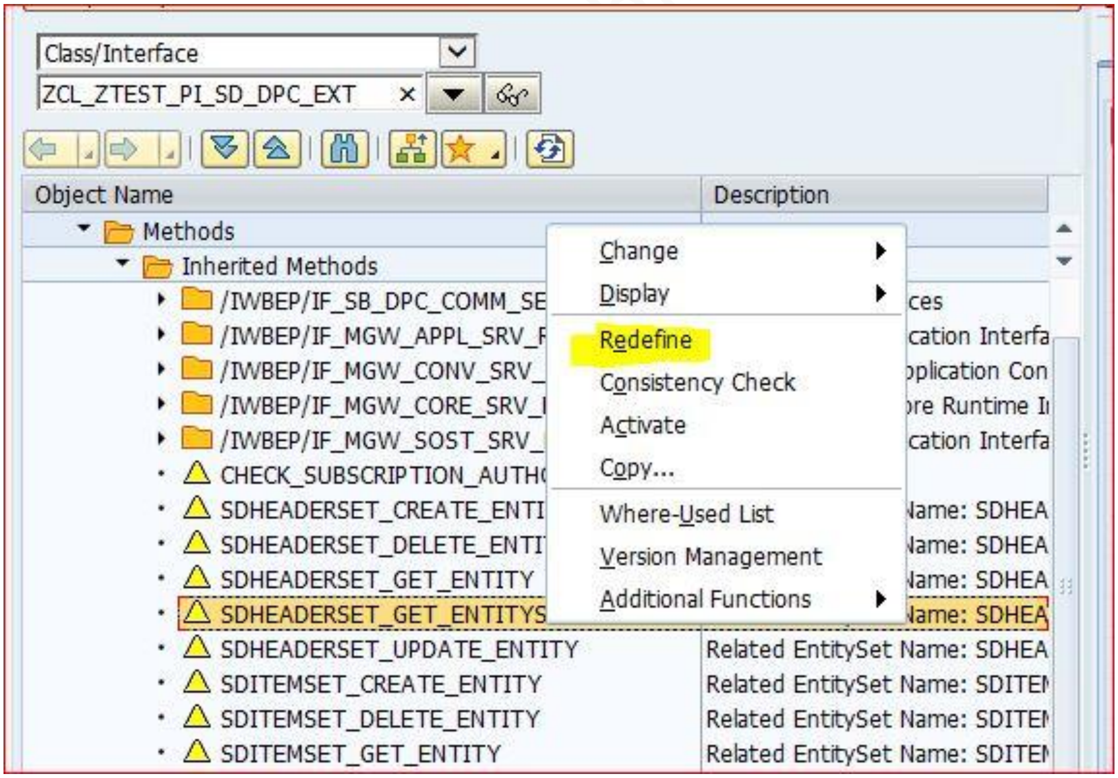
Step 3. Go to transaction code SEGW and expand folder service Implementation.

Under service implementation - Expand SDheaderSet. you will find different options. Right click on GetEntitySet and select GO to ABAP work Bench. It will redirect to the class.



ZTEST_PI_SD — Test for SD documents
- Data Model
- Service Implementation
  - SDHEADERSet
    - Create
    - Delete
    - GetEntity (Read)
    - GetEntitySet (Query)
    - Update
  - SDITEMSet
    - Create
    - Delete
    - GetEntity
    - GetEntity
    - Update
- Runtime Artifacts
  - ZCL_ZTEST_PI_SD_DPC
  - ZCL_ZTEST_PI_SD_DPC_EXT
  - ZCL_ZTEST_PI_SD_MPC
  - ZCL_ZTEST_PI_SD_MPC_EXT

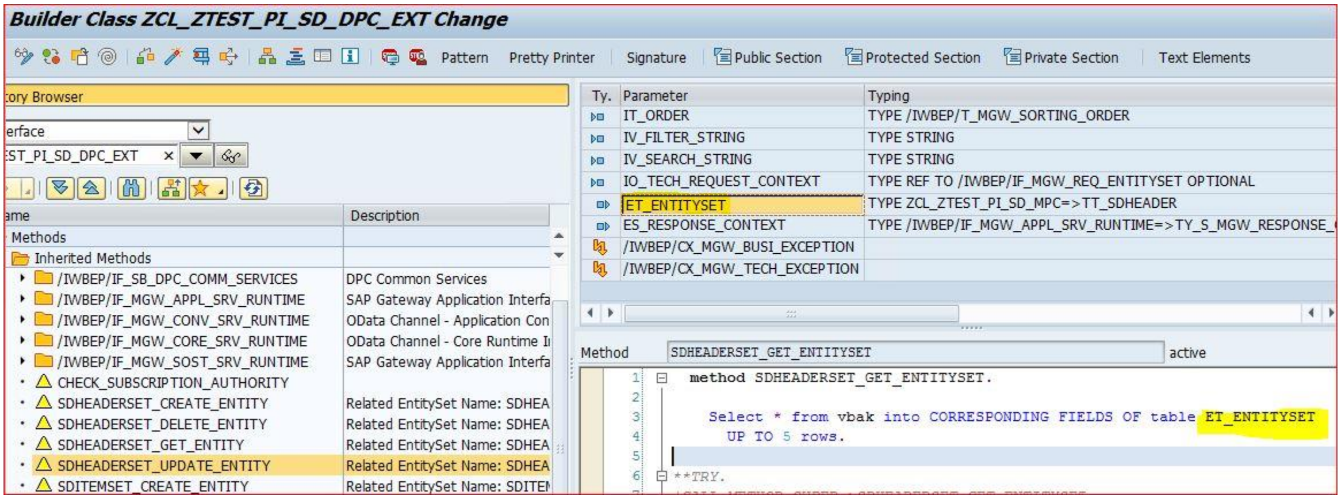
Now expand the method drop down and select method for GetEntitySet for Header data and right click and select redefine.



Class/Interface
ZCL_ZTEST_PI_SD_DPC_EXT

Object Name	Description
Methods	
Inherited Methods	
/IWBEP/IF_SB_DPC_COMM_SE	ces
/IWBEP/IF_MGW_APPL_SRV_F	cation Interfa
/IWBEP/IF_MGW_CONV_SRV_	pplication Con
/IWBEP/IF_MGW_CORE_SRV_I	ore Runtime I
/IWBEP/IF_MGW_SOST_SRV_	cation Interfa
CHECK_SUBSCRIPTION_AUTH	
SDHEADERSET_CREATE_ENTI	Name: SDHEA
SDHEADERSET_DELETE_ENTI	Name: SDHEA
SDHEADERSET_GET_ENTITY	Name: SDHEA
SDHEADERSET_GET_ENTITYS	Name: SDHEA
SDHEADERSET_UPDATE_ENTITY	Related EntitySet Name: SDHEA
SDITEMSET_CREATE_ENTITY	Related EntitySet Name: SDITEN
SDITEMSET_DELETE_ENTITY	Related EntitySet Name: SDITEN
SDITEMSET_GET_ENTITY	Related EntitySet Name: SDITEN

Write below code in method. Save and activate all related objects for classes.



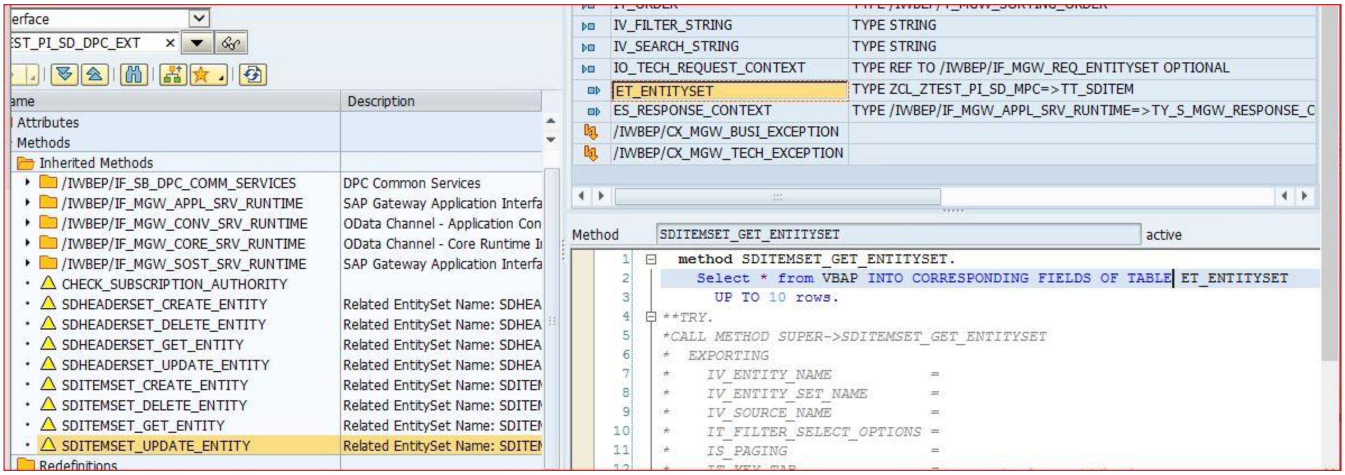
ame	Description
Methods	
Inherited Methods	
/IWBEP/IF_SB_DPC_COMM_SERVICES	DPC Common Services
/IWBEP/IF_MGW_APPL_SRV_RUNTIME	SAP Gateway Application Interfa
/IWBEP/IF_MGW_CONV_SRV_RUNTIME	OData Channel - Application Con
/IWBEP/IF_MGW_CORE_SRV_RUNTIME	OData Channel - Core Runtime I
/IWBEP/IF_MGW_SOST_SRV_RUNTIME	SAP Gateway Application Interfa
CHECK_SUBSCRIPTION_AUTHORITY	
SDHEADERSET_CREATE_ENTITY	Related EntitySet Name: SDHEA
SDHEADERSET_DELETE_ENTITY	Related EntitySet Name: SDHEA
SDHEADERSET_GET_ENTITY	Related EntitySet Name: SDHEA
SDHEADERSET_UPDATE_ENTITY	Related EntitySet Name: SDHEA
SDITEMSET_CREATE_ENTITY	Related EntitySet Name: SDITEI

Ty.	Parameter	Typing
	IT_ORDER	TYPE /IWBEP/T_MGW_SORTING_ORDER
	IV_FILTER_STRING	TYPE STRING
	IV_SEARCH_STRING	TYPE STRING
	IO_TECH_REQUEST_CONTEXT	TYPE REF TO /IWBEP/IF_MGW_REQ_ENTITYSET OPTIONAL
	ET_ENTITYSET	TYPE ZCL_ZTEST_PI_SD_MPC=>TT_SDHEADER
	ES_RESPONSE_CONTEXT	TYPE /IWBEP/IF_MGW_APPL_SRV_RUNTIME=>TY_S_MGW_RESPONSE_
	/IWBEP/CX_MGW_BUSI_EXCEPTION	
	/IWBEP/CX_MGW_TECH_EXCEPTION	

Method SDHEADERSET_GET_ENTITYSET active

```
 method SDHEADERSET_GET_ENTITYSET.

   Select * from vbak into CORRESPONDING FIELDS OF table ET_ENTITYSET
     UP TO 5 rows.

**TRY.
```

Similarly, Write the code for GetEntitySet for Item and activate.



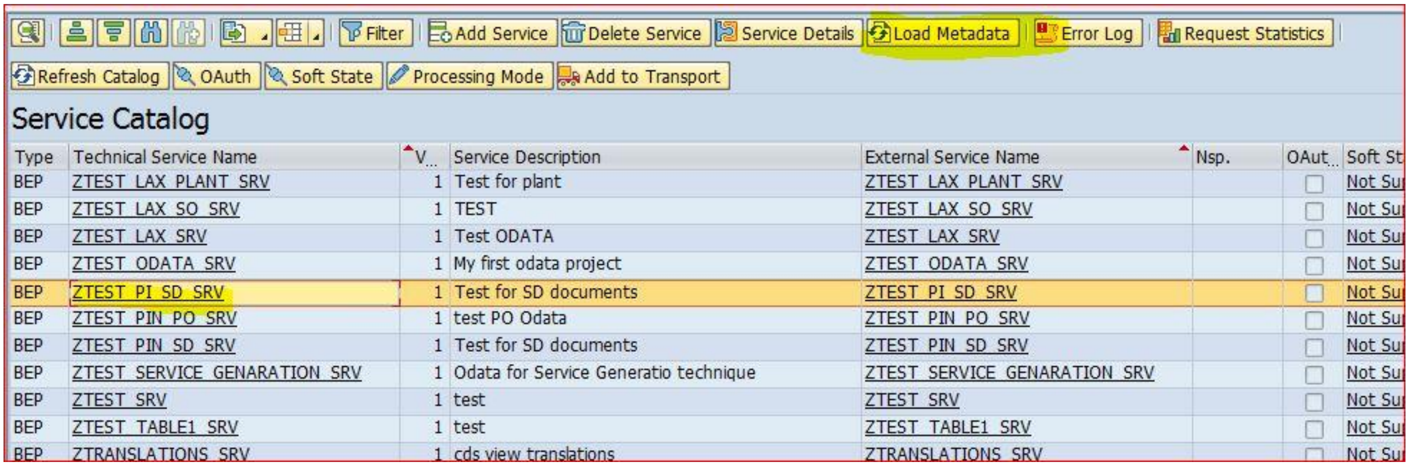
ame	Description
Attributes	
Methods	
Inherited Methods	
/IWBEP/IF_SB_DPC_COMM_SERVICES	DPC Common Services
/IWBEP/IF_MGW_APPL_SRV_RUNTIME	SAP Gateway Application Interfa
/IWBEP/IF_MGW_CONV_SRV_RUNTIME	OData Channel - Application Con
/IWBEP/IF_MGW_CORE_SRV_RUNTIME	OData Channel - Core Runtime I
/IWBEP/IF_MGW_SOST_SRV_RUNTIME	SAP Gateway Application Interfa
CHECK_SUBSCRIPTION_AUTHORITY	
SDHEADERSET_CREATE_ENTITY	Related EntitySet Name: SDHEA
SDHEADERSET_DELETE_ENTITY	Related EntitySet Name: SDHEA
SDHEADERSET_GET_ENTITY	Related EntitySet Name: SDHEA
SDHEADERSET_UPDATE_ENTITY	Related EntitySet Name: SDHEA
SDITEMSET_CREATE_ENTITY	Related EntitySet Name: SDITEI
SDITEMSET_DELETE_ENTITY	Related EntitySet Name: SDITEI
SDITEMSET_GET_ENTITY	Related EntitySet Name: SDITEI
SDITEMSET_UPDATE_ENTITY	Related EntitySet Name: SDITEI
Redefinitions	

Ty.	Parameter	Typing
	IV_FILTER_STRING	TYPE STRING
	IV_SEARCH_STRING	TYPE STRING
	IO_TECH_REQUEST_CONTEXT	TYPE REF TO /IWBEP/IF_MGW_REQ_ENTITYSET OPTIONAL
	ET_ENTITYSET	TYPE ZCL_ZTEST_PI_SD_MPC=>TT_SDITEM
	ES_RESPONSE_CONTEXT	TYPE /IWBEP/IF_MGW_APPL_SRV_RUNTIME=>TY_S_MGW_RESPONSE_C
	/IWBEP/CX_MGW_BUSI_EXCEPTION	
	/IWBEP/CX_MGW_TECH_EXCEPTION	

Method SDITEMSET_GET_ENTITYSET active

```
 method SDITEMSET_GET_ENTITYSET.
   Select * from VBAP INTO CORRESPONDING FIELDS OF TABLE ET_ENTITYSET
     UP TO 10 rows.
**TRY.
*CALL METHOD SUPER->SDITEMSET_GET_ENTITYSET
*  EXPORTING
*    IV_ENTITY_NAME           =
*    IV_ENTITY_SET_NAME       =
*    IV_SOURCE_NAME           =
*    IT_FILTER_SELECT_OPTIONS =
*    IS_PAGING                =
```

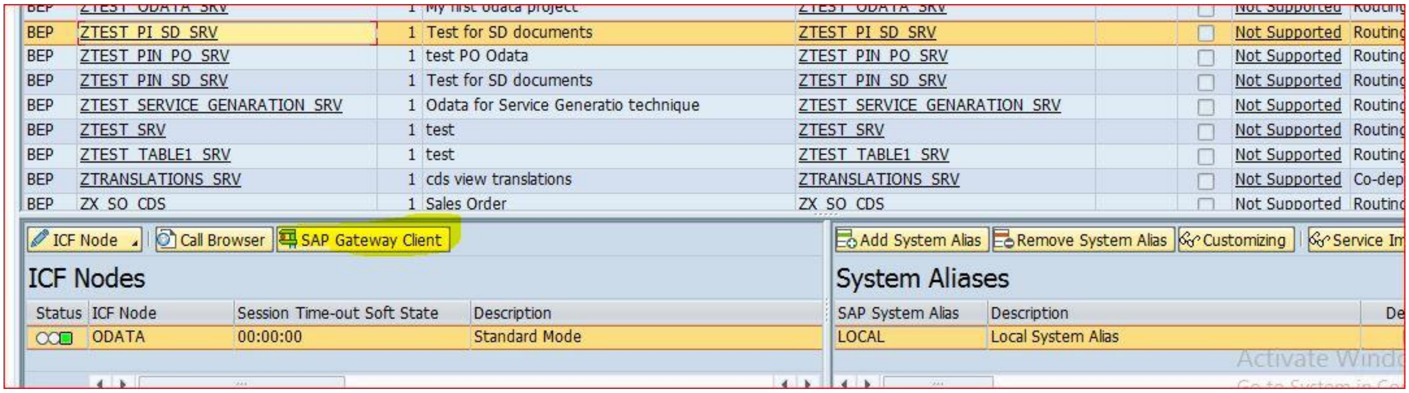
Step -4: Go to transaction /IWFND/MAINT_SERVICE and find out you service, select your service and click on load metadata.



Service Catalog

Type	Technical Service Name	V..	Service Description	External Service Name	Nsp.	OAut..	Soft St
BEP	ZTEST_LAX_PLANT_SRV	1	Test for plant	ZTEST_LAX_PLANT_SRV		☐	Not Su
BEP	ZTEST_LAX_SO_SRV	1	TEST	ZTEST_LAX_SO_SRV		☐	Not Su
BEP	ZTEST_LAX_SRV	1	Test ODATA	ZTEST_LAX_SRV		☐	Not Su
BEP	ZTEST_ODATA_SRV	1	My first odata project	ZTEST_ODATA_SRV		☐	Not Su
BEP	ZTEST_PI_SD_SRV	1	Test for SD documents	ZTEST_PI_SD_SRV		☐	Not Su
BEP	ZTEST_PIN_PO_SRV	1	test PO Odata	ZTEST_PIN_PO_SRV		☐	Not Su
BEP	ZTEST_PIN_SD_SRV	1	Test for SD documents	ZTEST_PIN_SD_SRV		☐	Not Su
BEP	ZTEST_SERVICE_GENARATION_SRV	1	Odata for Service Generatio technique	ZTEST_SERVICE_GENARATION_SRV		☐	Not Su
BEP	ZTEST_SRV	1	test	ZTEST_SRV		☐	Not Su
BEP	ZTEST_TABLE1_SRV	1	test	ZTEST_TABLE1_SRV		☐	Not Su
BEP	ZTRANSLATIONS_SRV	1	cds view translations	ZTRANSLATIONS_SRV		☐	Not Su

Information message will pop up - Metadata has been loaded successfully. Now Click on SAP Gateway Client.



BEP	ZTEST_ODATA_SRV	1	My first odata project	ZTEST_ODATA_SRV	□	Not Supported	Routing
BEP	ZTEST_PI_SD_SRV	1	Test for SD documents	ZTEST_PI_SD_SRV	□	Not Supported	Routing
BEP	ZTEST_PIN_PO_SRV	1	test PO Odata	ZTEST_PIN_PO_SRV	□	Not Supported	Routing
BEP	ZTEST_PIN_SD_SRV	1	Test for SD documents	ZTEST_PIN_SD_SRV	□	Not Supported	Routing
BEP	ZTEST_SERVICE_GENARATION_SRV	1	Odata for Service Generatio technique	ZTEST_SERVICE_GENARATION_SRV	□	Not Supported	Routing
BEP	ZTEST_SRV	1	test	ZTEST_SRV	□	Not Supported	Routing
BEP	ZTEST_TABLE1_SRV	1	test	ZTEST_TABLE1_SRV	□	Not Supported	Routing
BEP	ZTRANSLATIONS_SRV	1	cds view translations	ZTRANSLATIONS_SRV	□	Not Supported	Co-dep
BEP	ZX_SO_CDS	1	Sales Order	ZX_SO_CDS	□	Not Supported	Routing

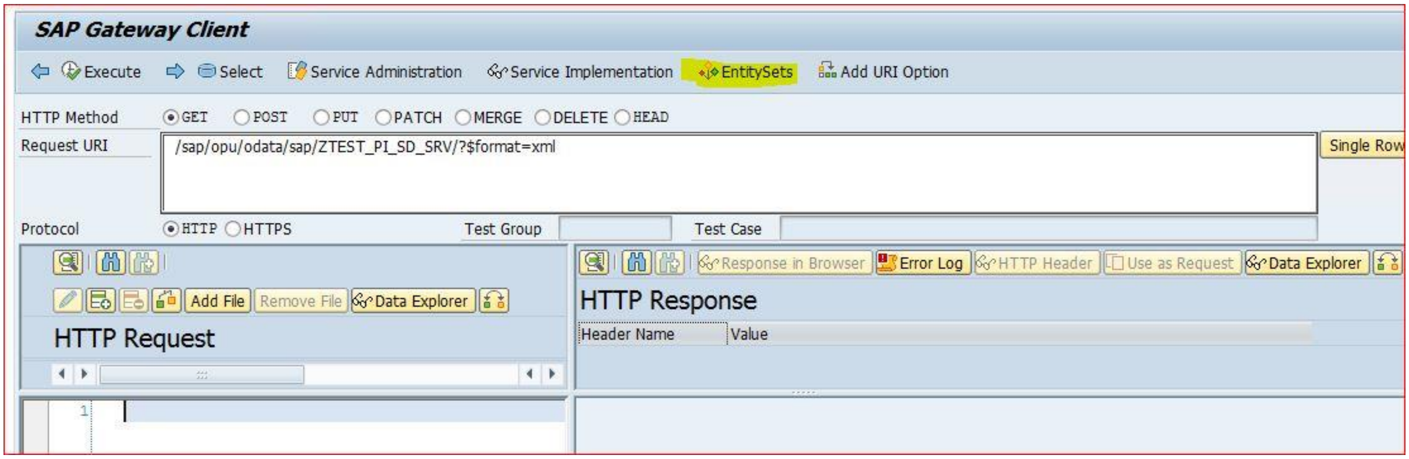
ICF Nodes

Status	ICF Node	Session Time-out Soft State	Description
○○🟩	ODATA	00:00:00	Standard Mode

System Aliases

SAP System Alias	Description	De
LOCAL	Local System Alias	

Now Click on Entity set.



SAP Gateway Client

HTTP Method ● GET ○ POST ○ PUT ○ PATCH ○ MERGE ○ DELETE ○ HEAD

Request URI /sap/opu/odata/sap/ZTEST_PI_SD_SRV/?$format=xml

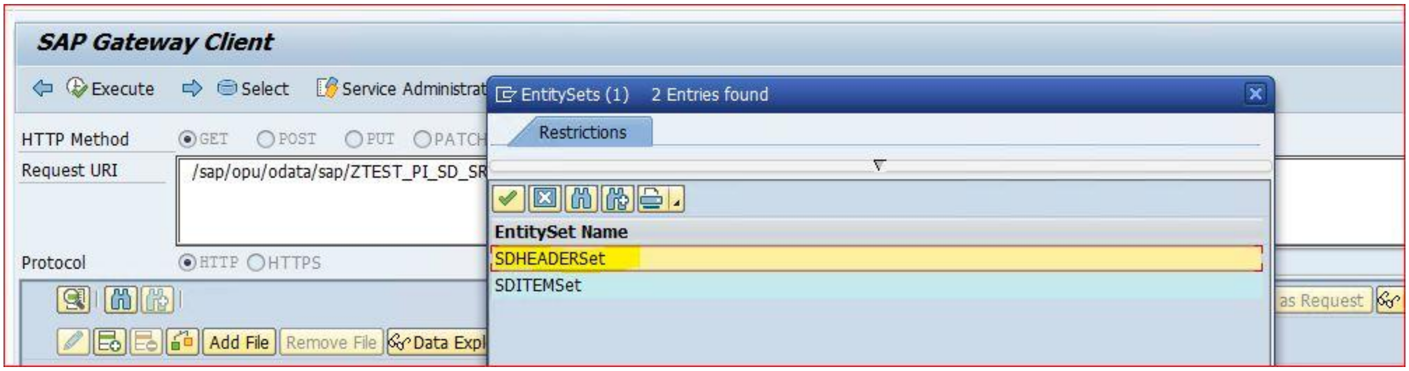
Protocol ● HTTP ○ HTTPS Test Group Test Case

HTTP Request

HTTP Response

Header Name	Value

Select entity for header and enter -



SAP Gateway Client

HTTP Method ● GET ○ POST ○ PUT ○ PATCH

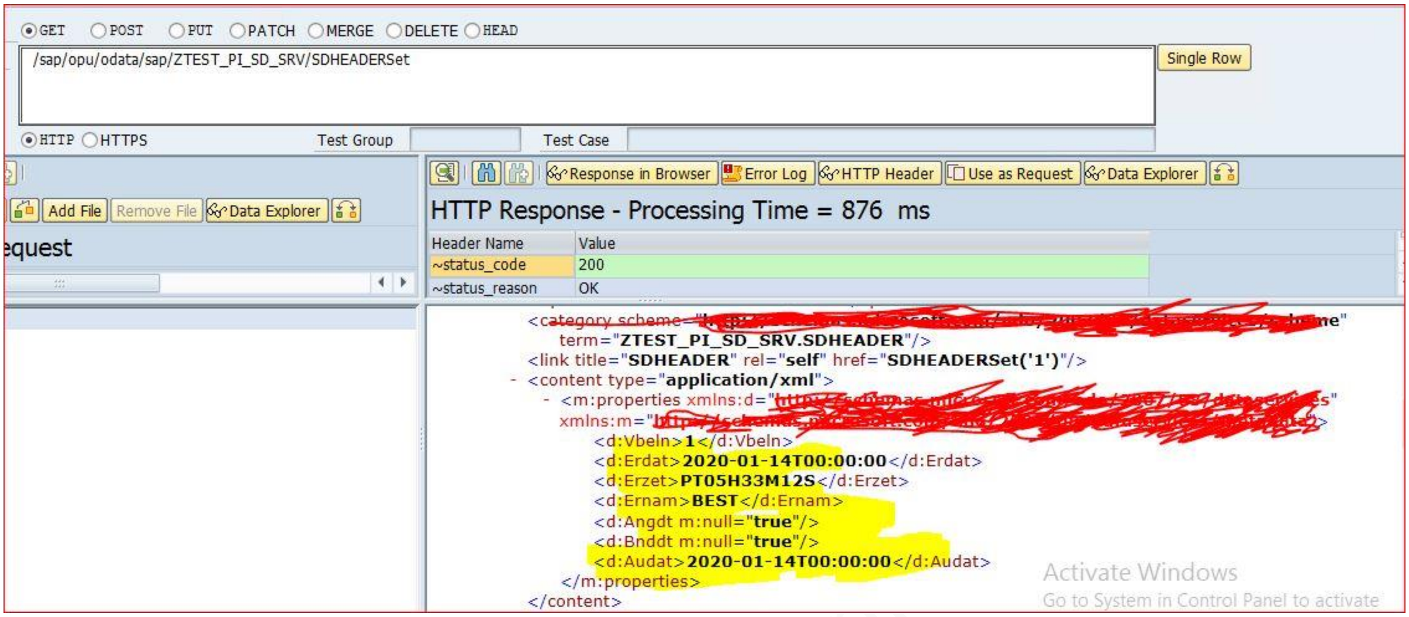
Request URI /sap/opu/odata/sap/ZTEST_PI_SD_SR

Protocol ● HTTP ○ HTTPS

EntitySets (1) 2 Entries found

EntitySet Name
SDHEADERSet
SDITEMSet

Click and execute and you will get the header data in response.



● GET ○ POST ○ PUT ○ PATCH ○ MERGE ○ DELETE ○ HEAD

/sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDHEADERSet

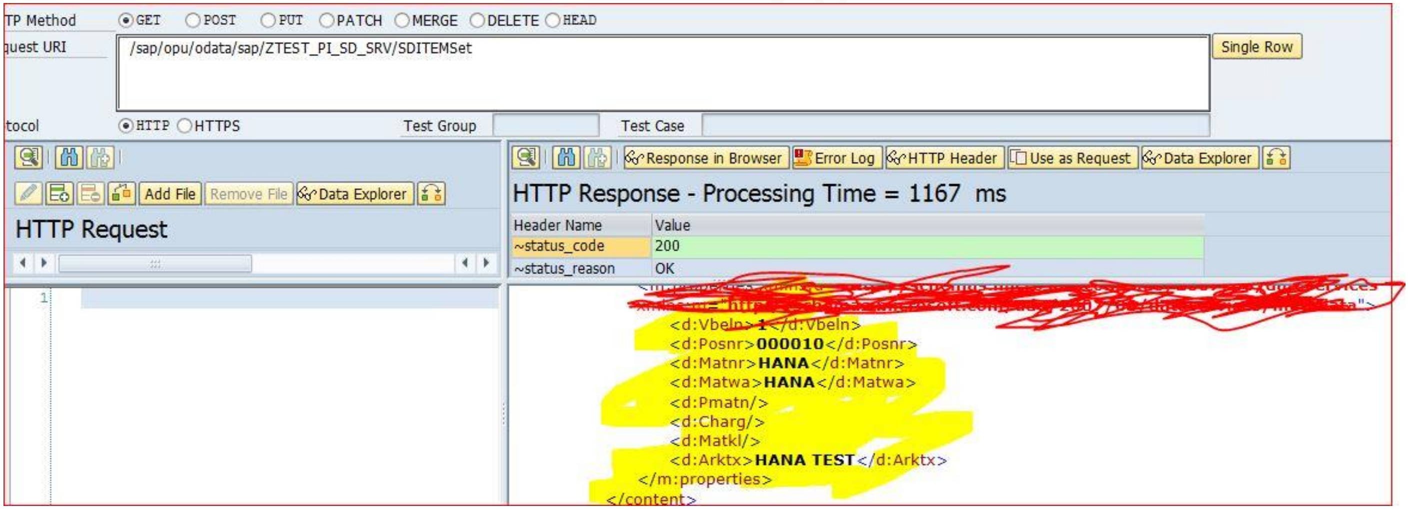
● HTTP ○ HTTPS Test Group Test Case

HTTP Response - Processing Time = 876 ms

Header Name	Value
~status_code	200
~status_reason	OK

```
<category scheme="[redacted]me"
  term="ZTEST_PI_SD_SRV.SDHEADER"/>
<link title="SDHEADER" rel="self" href="SDHEADERSet('1')"/>
- <content type="application/xml">
  - <m:properties xmlns:d="[redacted]"
    xmlns:m="[redacted]">
      <d:Vbeln>1</d:Vbeln>
      <d:Erdat>2020-01-14T00:00:00</d:Erdat>
      <d:Erzet>PT05H33M12S</d:Erzet>
      <d:Ernam>BEST</d:Ernam>
      <d:Angdt m:null="true"/>
      <d:Bnddt m:null="true"/>
      <d:Audat>2020-01-14T00:00:00</d:Audat>
    </m:properties>
</content>
```

Similarly, select entity set for item and execute. You will get data for item



TP Method ● GET ○ POST ○ PUT ○ PATCH ○ MERGE ○ DELETE ○ HEAD

quest URI /sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDITEMSet

tocol ● HTTP ○ HTTPS Test Group Test Case

HTTP Request

HTTP Response - Processing Time = 1167 ms

Header Name	Value
~status_code	200
~status_reason	OK

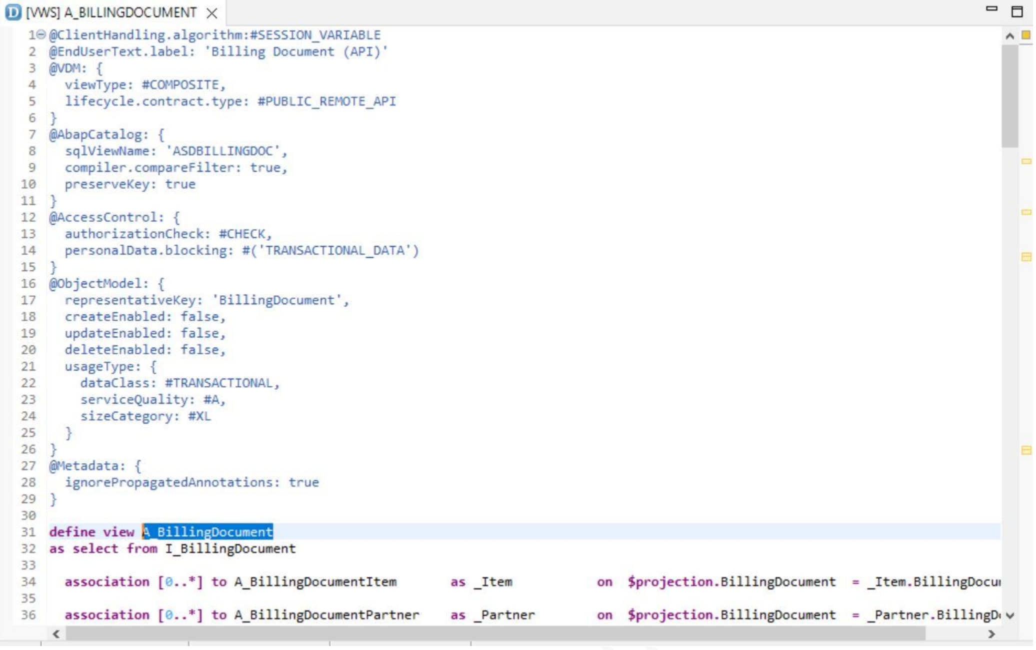
```
[redacted]
    <d:Vbeln>1</d:Vbeln>
    <d:Posnr>000010</d:Posnr>
    <d:Matnr>HANA</d:Matnr>
    <d:Matwa>HANA</d:Matwa>
    <d:Pmatn/>
    <d:Charg/>
    <d:Matkl/>
    <d:Arktx>HANA TEST</d:Arktx>
  </m:properties>
</content>
```

That’s all. We have completed our Odata service creation with a Service Gateway Builder Project with two entities and two entity-sets. We have generated the runtime artifacts and registered and activated our OData service. We had redefined methods of header and item entity set, load metadata and test the entity set with for test data.

SAP Gateway Based OData Service using SADL Framework –

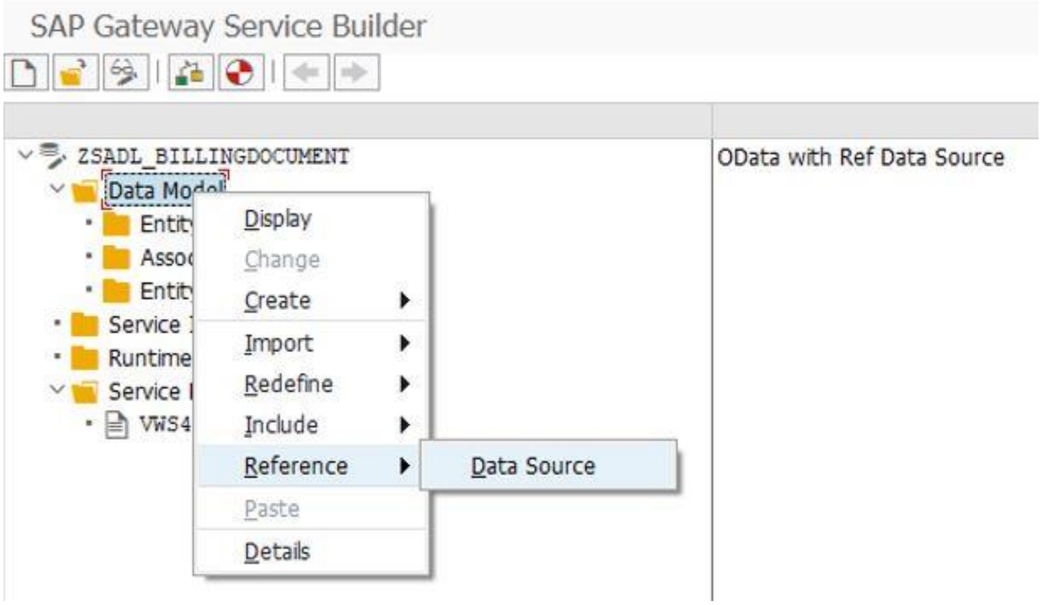
Let’s go with the steps on how to create an OData service using the Service Adaptation Description Language (SADL) framework and referencing SAP ABAP CDS views in SEGW Gateway Builder. Content is taken from a blog written by ***Vishal Kumar***.

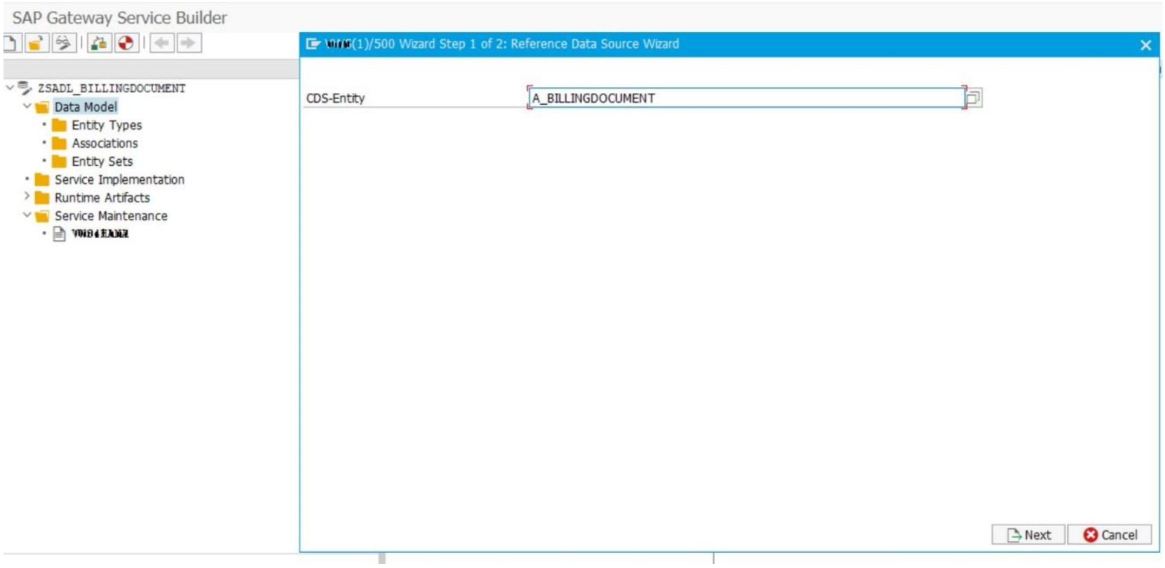
We will use a standard CDS view - A_BILLINGDOCUMENT.



```
@ClientHandling.algorithm:#SESSION_VARIABLE
@EndUserText.label: 'Billing Document (API)'
@VDM: {
  viewType: #COMPOSITE,
  lifecycle.contract.type: #PUBLIC_REMOTE_API
}
@AbapCatalog: {
  sqlViewName: 'ASDBILLINGDOC',
  compiler.compareFilter: true,
  preserveKey: true
}
@AccessControl: {
  authorizationCheck: #CHECK,
  personalData.blocking: #('TRANSACTIONAL_DATA')
}
@ObjectModel: {
  representativeKey: 'BillingDocument',
  createEnabled: false,
  updateEnabled: false,
  deleteEnabled: false,
  usageType: {
    dataClass: #TRANSACTIONAL,
    serviceQuality: #A,
    sizeCategory: #XL
  }
}
@Metadata: {
  ignorePropagatedAnnotations: true
}

define view A_BillingDocument
as select from I_BillingDocument

  association [0..*] to A_BillingDocumentItem       as _Item          on  $projection.BillingDocument  = _Item.BillingDocu

  association [0..*] to A_BillingDocumentPartner    as _Partner       on  $projection.BillingDocument  = _Partner.BillingD
```

Create an empty project in Service Gateway Builder using the T-code: SEGW. To create the Data Model, we need to refer to Data Source. (CDS View in this case).





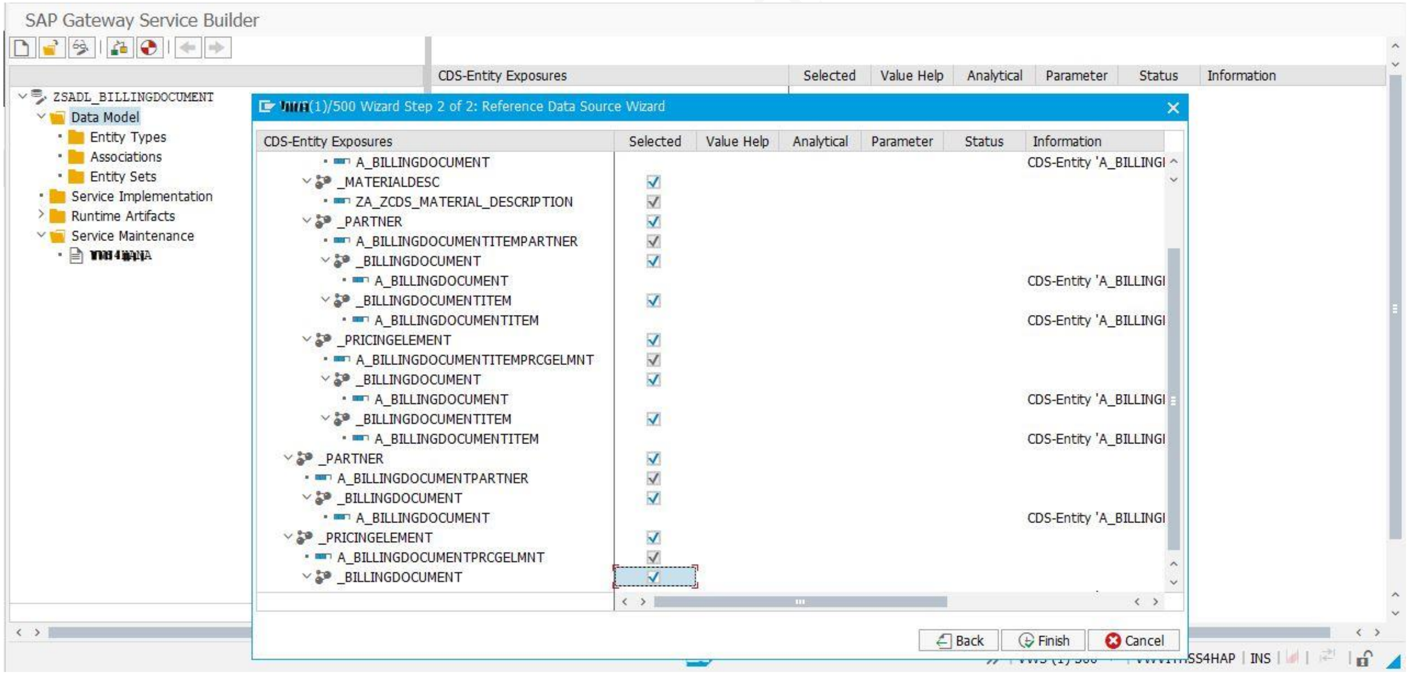
SAP Gateway Service Builder

- ZSADL_BILLINGDOCUMENT
  - Data Model
    - Entity Types
    - Associations
    - Entity Sets
  - Service Implementation
  - Runtime Artifacts
  - Service Maintenance
    - [redacted]

[redacted](1)/500 Wizard Step 1 of 2: Reference Data Source Wizard

CDS-Entity: A_BILLINGDOCUMENT

In the next screen, we can see all the elements of CDS entity and corresponding Associations. Select all the elements and proceed further. (We can select only the fields which are relevant for as per Business Need and we always have the option to edit that later).



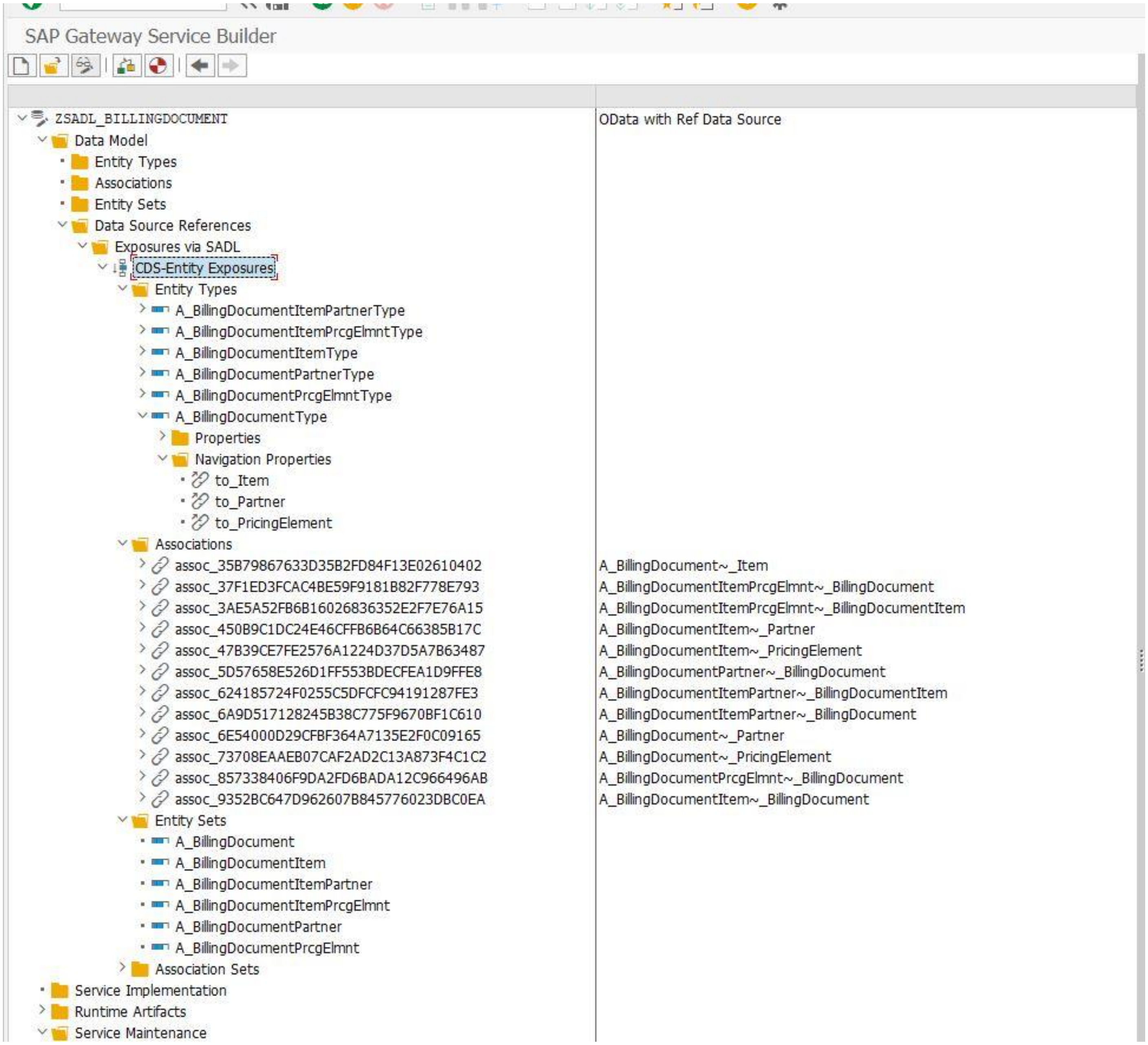
SAP Gateway Service Builder

- ZSADL_BILLINGDOCUMENT
  - Data Model
    - Entity Types
    - Associations
    - Entity Sets
  - Service Implementation
  - Runtime Artifacts
  - Service Maintenance
    - [redacted]

[redacted](1)/500 Wizard Step 2 of 2: Reference Data Source Wizard

CDS-Entity Exposures	Selected	Value Help	Analytical	Parameter	Status	Information
A_BILLINGDOCUMENT						CDS-Entity 'A_BILLINGI
_MATERIALDESC	☑					
ZA_ZCDS_MATERIAL_DESCRIPTION	☑					
_PARTNER	☑					
A_BILLINGDOCUMENTITEMPARTNER	☑					
_BILLINGDOCUMENT	☑					
A_BILLINGDOCUMENT						CDS-Entity 'A_BILLINGI
_BILLINGDOCUMENTITEM	☑					
A_BILLINGDOCUMENTITEM						CDS-Entity 'A_BILLINGI
_PRICINGELEMENT	☑					
A_BILLINGDOCUMENTITEMPRCGELMNT	☑					
_BILLINGDOCUMENT	☑					
A_BILLINGDOCUMENT						CDS-Entity 'A_BILLINGI
_BILLINGDOCUMENTITEM	☑					
A_BILLINGDOCUMENTITEM						CDS-Entity 'A_BILLINGI
_PARTNER	☑					
A_BILLINGDOCUMENTPARTNER	☑					
_BILLINGDOCUMENT	☑					
A_BILLINGDOCUMENT						CDS-Entity 'A_BILLINGI
_PRICINGELEMENT	☑					
A_BILLINGDOCUMENTPRCGELMNT	☑					
_BILLINGDOCUMENT	☑					

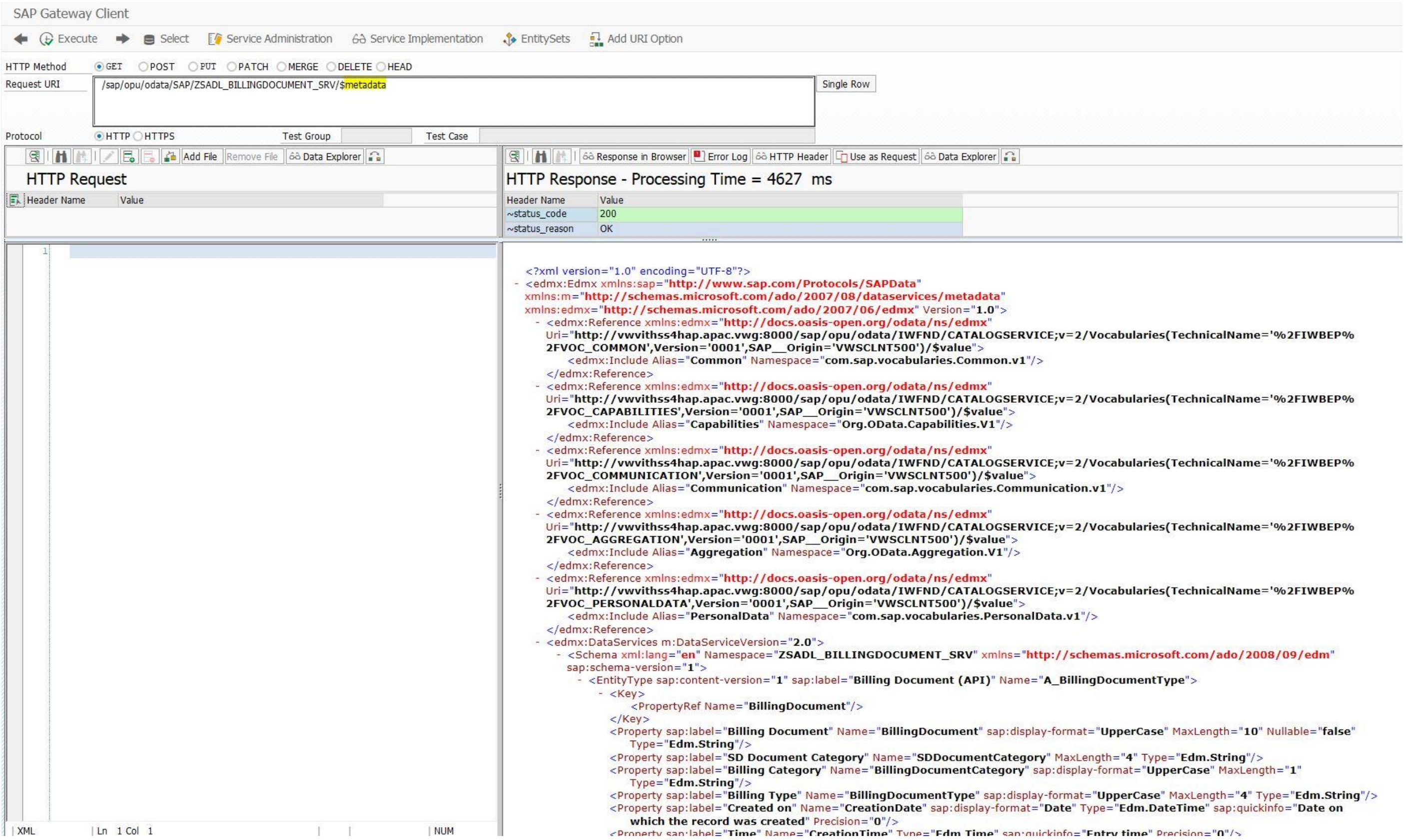
After completing this step all the Entity, Entity Sets, Association & corresponding Navigation is automatically generated in the Gateway Project.



SAP Gateway Service Builder

```
ZSADL_BILLINGDOCUMENT                                              OData with Ref Data Source
  Data Model
    Entity Types
    Associations
    Entity Sets
    Data Source References
      Exposures via SADL
        CDS-Entity Exposures
          Entity Types
            A_BillingDocumentItemPartnerType
            A_BillingDocumentItemPrcgElmntType
            A_BillingDocumentItemType
            A_BillingDocumentPartnerType
            A_BillingDocumentPrcgElmntType
            A_BillingDocumentType
              Properties
              Navigation Properties
                to_Item
                to_Partner
                to_PricingElement
          Associations
            assoc_35B79867633D35B2FD84F13E02610402                 A_BillingDocument~_Item
            assoc_37F1ED3FCAC4BE59F9181B82F778E793                 A_BillingDocumentItemPrcgElmnt~_BillingDocument
            assoc_3AE5A52FB6B16026836352E2F7E76A15                 A_BillingDocumentItemPrcgElmnt~_BillingDocumentItem
            assoc_450B9C1DC24E46CFFB6B64C66385B17C                 A_BillingDocumentItem~_Partner
            assoc_47B39CE7FE2576A1224D37D5A7B63487                 A_BillingDocumentItem~_PricingElement
            assoc_5D57658E526D1FF553BDECFEA1D9FFE8                 A_BillingDocumentPartner~_BillingDocument
            assoc_624185724F0255C5DFCFC94191287FE3                 A_BillingDocumentItemPartner~_BillingDocumentItem
            assoc_6A9D517128245B38C775F9670BF1C610                 A_BillingDocumentItemPartner~_BillingDocument
            assoc_6E54000D29CFBF364A7135E2F0C09165                 A_BillingDocument~_Partner
            assoc_73708EAAEB07CAF2AD2C13A873F4C1C2                 A_BillingDocument~_PricingElement
            assoc_857338406F9DA2FD6BADA12C966496AB                 A_BillingDocumentPrcgElmnt~_BillingDocument
            assoc_9352BC647D962607B845776023DBC0EA                 A_BillingDocumentItem~_BillingDocument
          Entity Sets
            A_BillingDocument
            A_BillingDocumentItem
            A_BillingDocumentItemPartner
            A_BillingDocumentItemPrcgElmnt
            A_BillingDocumentPartner
            A_BillingDocumentPrcgElmnt
          Association Sets
  Service Implementation
  Runtime Artifacts
  Service Maintenance
```

Now, generate the service. After generating Register and Maintain the service. Test the OData service in Gateway Client (/IWFND/GW_CLIENT).
We can see all the Entity and corresponding Association in the Metadata Information.



SAP Gateway Client

HTTP Method ● GET ○ POST ○ PUT ○ PATCH ○ MERGE ○ DELETE ○ HEAD

Request URI /sap/opu/odata/SAP/ZSADL_BILLINGDOCUMENT_SRV/$metadata

Protocol ● HTTP ○ HTTPS Test Group Test Case

HTTP Request

Header Name	Value

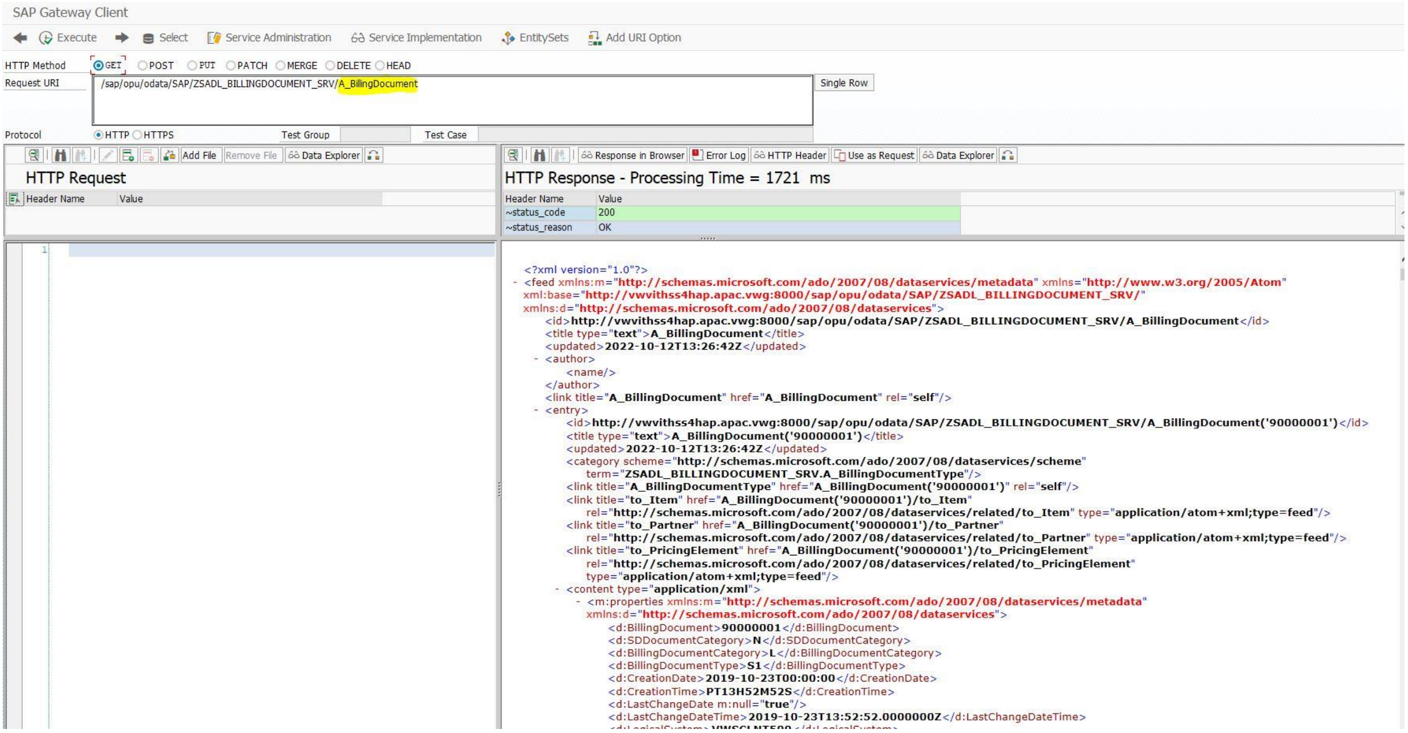
HTTP Response - Processing Time = 4627 ms

Header Name	Value
~status_code	200
~status_reason	OK

XML Ln 1 Col 1 NUM

```
  <?xml version="1.0" encoding="UTF-8"?>
- <edmx:Edmx xmlns:sap="http://www.sap.com/Protocols/SAPData"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  - <edmx:Reference xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx"
    Uri="http://vwvithss4hap.apac.vwg:8000/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%
    2FVOC_COMMON',Version='0001',SAP__Origin='VWSCLNT500')/$value">
        <edmx:Include Alias="Common" Namespace="com.sap.vocabularies.Common.v1"/>
    </edmx:Reference>
  - <edmx:Reference xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx"
    Uri="http://vwvithss4hap.apac.vwg:8000/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%
    2FVOC_CAPABILITIES',Version='0001',SAP__Origin='VWSCLNT500')/$value">
        <edmx:Include Alias="Capabilities" Namespace="Org.OData.Capabilities.V1"/>
    </edmx:Reference>
  - <edmx:Reference xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx"
    Uri="http://vwvithss4hap.apac.vwg:8000/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%
    2FVOC_COMMUNICATION',Version='0001',SAP__Origin='VWSCLNT500')/$value">
        <edmx:Include Alias="Communication" Namespace="com.sap.vocabularies.Communication.v1"/>
    </edmx:Reference>
  - <edmx:Reference xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx"
    Uri="http://vwvithss4hap.apac.vwg:8000/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%
    2FVOC_AGGREGATION',Version='0001',SAP__Origin='VWSCLNT500')/$value">
        <edmx:Include Alias="Aggregation" Namespace="Org.OData.Aggregation.V1"/>
    </edmx:Reference>
  - <edmx:Reference xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx"
    Uri="http://vwvithss4hap.apac.vwg:8000/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%
    2FVOC_PERSONALDATA',Version='0001',SAP__Origin='VWSCLNT500')/$value">
        <edmx:Include Alias="PersonalData" Namespace="com.sap.vocabularies.PersonalData.v1"/>
    </edmx:Reference>
  - <edmx:DataServices m:DataServiceVersion="2.0">
      - <Schema xml:lang="en" Namespace="ZSADL_BILLINGDOCUMENT_SRV" xmlns="http://schemas.microsoft.com/ado/2008/09/edm"
        sap:schema-version="1">
          - <EntityType sap:content-version="1" sap:label="Billing Document (API)" Name="A_BillingDocumentType">
              - <Key>
                    <PropertyRef Name="BillingDocument"/>
                </Key>
                <Property sap:label="Billing Document" Name="BillingDocument" sap:display-format="UpperCase" MaxLength="10" Nullable="false"
                    Type="Edm.String"/>
                <Property sap:label="SD Document Category" Name="SDDocumentCategory" MaxLength="4" Type="Edm.String"/>
                <Property sap:label="Billing Category" Name="BillingDocumentCategory" sap:display-format="UpperCase" MaxLength="1"
                    Type="Edm.String"/>
                <Property sap:label="Billing Type" Name="BillingDocumentType" sap:display-format="UpperCase" MaxLength="4" Type="Edm.String"/>
                <Property sap:label="Created on" Name="CreationDate" sap:display-format="Date" Type="Edm.DateTime" sap:quickinfo="Date on
                    which the record was created" Precision="0"/>
                <Property sap:label="Time" Name="CreationTime" Type="Edm.Time" sap:quickinfo="Entry time" Precision="0"/>
```

We can execute the service with different Entities to see the results.



SAP Gateway Client

HTTP Method ● GET ○ POST ○ PUT ○ PATCH ○ MERGE ○ DELETE ○ HEAD

Request URI /sap/opu/odata/SAP/ZSADL_BILLINGDOCUMENT_SRV/A_BillingDocument

Protocol ● HTTP ○ HTTPS Test Group Test Case

HTTP Request

Header Name	Value

HTTP Response - Processing Time = 1721 ms

Header Name	Value
~status_code	200
~status_reason	OK

```
  <?xml version="1.0"?>
- <feed xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns="http://www.w3.org/2005/Atom"
  xml:base="http://vwvithss4hap.apac.vwg:8000/sap/opu/odata/SAP/ZSADL_BILLINGDOCUMENT_SRV/"
  xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
      <id>http://vwvithss4hap.apac.vwg:8000/sap/opu/odata/SAP/ZSADL_BILLINGDOCUMENT_SRV/A_BillingDocument</id>
      <title type="text">A_BillingDocument</title>
      <updated>2022-10-12T13:26:42Z</updated>
    - <author>
          <name/>
      </author>
      <link title="A_BillingDocument" href="A_BillingDocument" rel="self"/>
    - <entry>
          <id>http://vwvithss4hap.apac.vwg:8000/sap/opu/odata/SAP/ZSADL_BILLINGDOCUMENT_SRV/A_BillingDocument('90000001')</id>
          <title type="text">A_BillingDocument('90000001')</title>
          <updated>2022-10-12T13:26:42Z</updated>
          <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"
              term="ZSADL_BILLINGDOCUMENT_SRV.A_BillingDocumentType"/>
          <link title="A_BillingDocumentType" href="A_BillingDocument('90000001')" rel="self"/>
          <link title="to_Item" href="A_BillingDocument('90000001')/to_Item"
              rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/to_Item" type="application/atom+xml;type=feed"/>
          <link title="to_Partner" href="A_BillingDocument('90000001')/to_Partner"
              rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/to_Partner" type="application/atom+xml;type=feed"/>
          <link title="to_PricingElement" href="A_BillingDocument('90000001')/to_PricingElement"
              rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/to_PricingElement"
              type="application/atom+xml;type=feed"/>
        - <content type="application/xml">
            - <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
              xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
                  <d:BillingDocument>90000001</d:BillingDocument>
                  <d:SDDocumentCategory>N</d:SDDocumentCategory>
                  <d:BillingDocumentCategory>L</d:BillingDocumentCategory>
                  <d:BillingDocumentType>S1</d:BillingDocumentType>
                  <d:CreationDate>2019-10-23T00:00:00</d:CreationDate>
                  <d:CreationTime>PT13H52M52S</d:CreationTime>
                  <d:LastChangeDate m:null="true"/>
                  <d:LastChangeDateTime>2019-10-23T13:52:52.0000000Z</d:LastChangeDateTime>
                  <d:LogicalSystem>VWSCLNT500</d:LogicalSystem>
```

Publishing CDS View as OData Service –

This method is also referred as annotation-based approach using CDS views. Content is taken from a blog written by ***Jagdish Patil***.

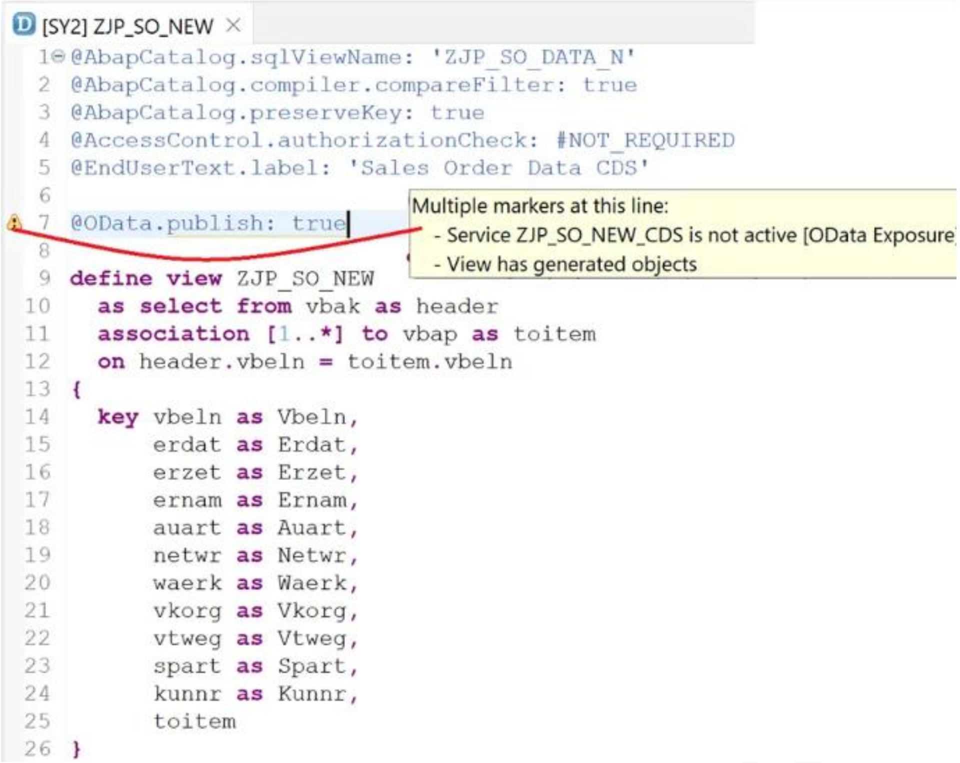
Create a CDS View with the below code. The annotation **@OData.publish: true** is used to publish the CDS as OData Service.

```
@AbapCatalog.sqlViewName: 'ZJP_SO_DATA_N'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Sales Order Data CDS'
@OData.publish: true
define view ZJP_SO_NEW
  as select from vbak as header
  association [1..*] to vbap as _item
  on header.vbeln = _item.vbeln
{
  key vbeln as Vbeln,
      erdat as Erdat,
      erzet as Erzet,
      ernam as Ernam,
      auart as Auart,
      netwr as Netwr,
      waerk as Waerk,
      vkorg as Vkorg,
      vtweg as Vtweg,
      spart as Spart,
      kunnr as Kunnr,
      _item
}
```

Sachin Artani

Activate the CDS View created. Post Activation, a yellow warning symbol appears beside the annotation **@OData.publish: true**. Hover the cursor over the line to get a pop up like below.

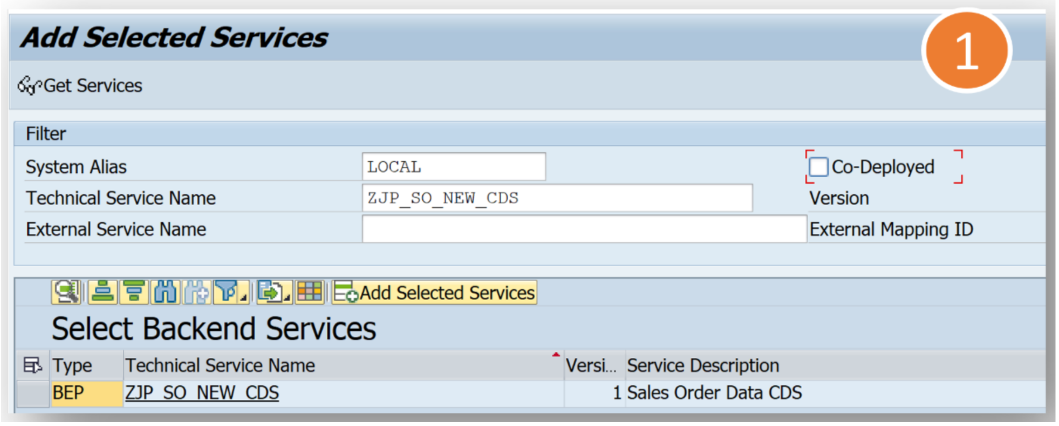
The Service Name appears in the pop-up and the warning says that the Service is not active. The service name will be the CDS View name followed by _CDS.



```
@AbapCatalog.sqlViewName: 'ZJP_SO_DATA_N'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Sales Order Data CDS'

@OData.publish: true

define view ZJP_SO_NEW
  as select from vbak as header
  association [1..*] to vbap as toitem
  on header.vbeln = toitem.vbeln
{
  key vbeln as Vbeln,
      erdat as Erdat,
      erzet as Erzet,
      ernam as Ernam,
      auart as Auart,
      netwr as Netwr,
      waerk as Waerk,
      vkorg as Vkorg,
      vtweg as Vtweg,
      spart as Spart,
      kunnr as Kunnr,
      toitem
}
```

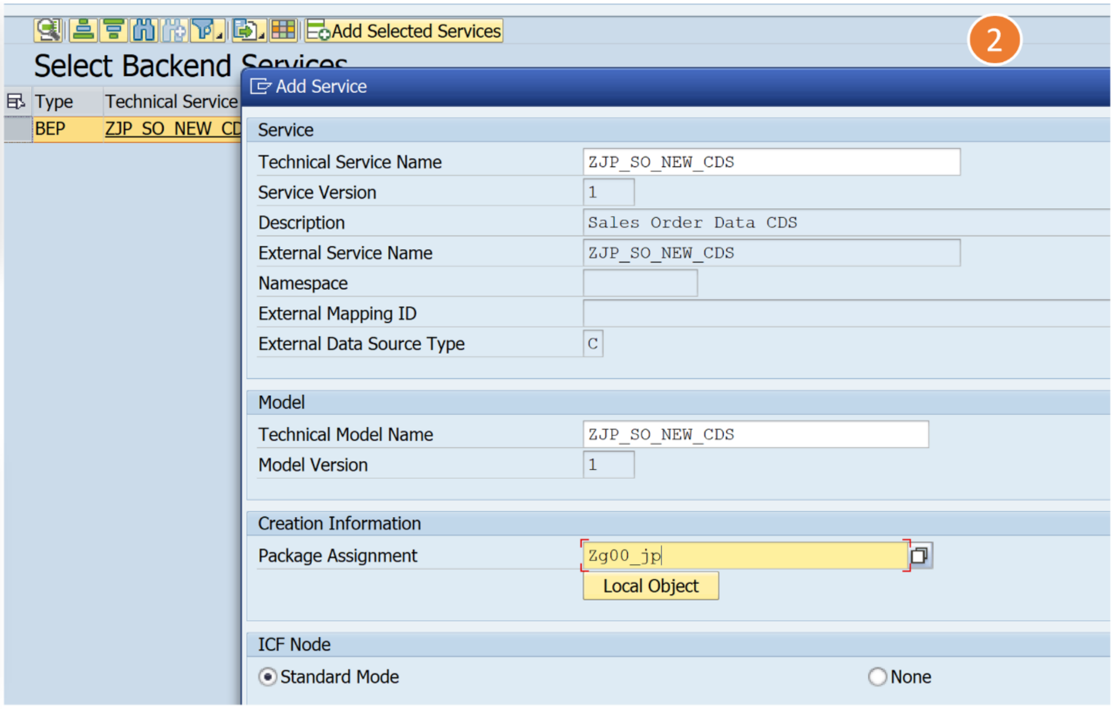
Multiple markers at this line:
- Service ZJP_SO_NEW_CDS is not active [OData Exposure
- View has generated objects

Register the service ZJP_SO_NEW_CDS using transaction /IWFND/MAINT_SERVICE. Use the add service button.



System Alias	LOCAL	□ Co-Deployed	
Technical Service Name	ZJP_SO_NEW_CDS	Version	
External Service Name		External Mapping ID	

Type	Technical Service Name	Versi...	Service Description
BEP	ZJP_SO_NEW_CDS	1	Sales Order Data CDS



Select Backend Services

Type	Technical Service
BEP	ZJP_SO_NEW_CD

Service

Technical Service Name	ZJP_SO_NEW_CDS
Service Version	1
Description	Sales Order Data CDS
External Service Name	ZJP_SO_NEW_CDS
Namespace	
External Mapping ID	
External Data Source Type	C

Model

Technical Model Name	ZJP_SO_NEW_CDS
Model Version	1

Creation Information

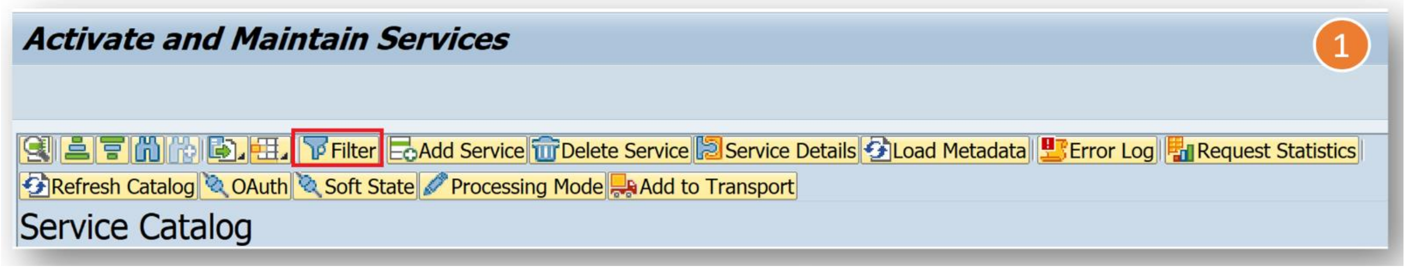
Package Assignment	Zg00_jp

Local Object

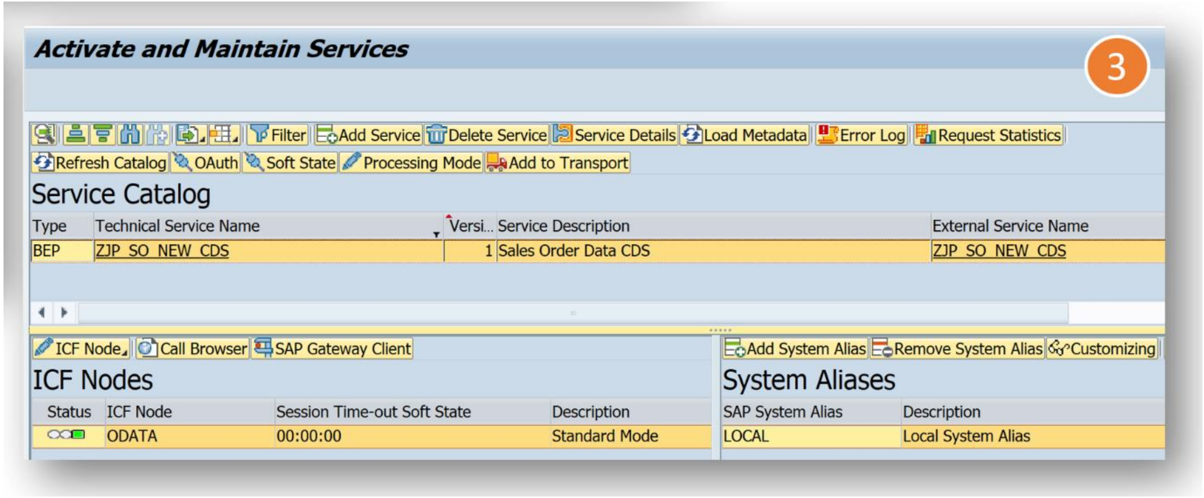
ICF Node

● Standard Mode ○ None

And the service is registered. Now, Test the Service. Use Filter to get the registered service. Start from the transaction /IWFND/MAINT_SERVICE. Click on the Filter button.



Use the exact service name or something like ZJP* to filter the services. Click on the right service and use SAP Gateway Client to test the service.



Service Catalog

Type	Technical Service Name	Versi..	Service Description	External Service Name
BEP	ZJP_SO_NEW_CDS	1	Sales Order Data CDS	ZJP_SO_NEW_CDS

ICF Nodes

Status	ICF Node	Session Time-out Soft State	Description
○○●	ODATA	00:00:00	Standard Mode

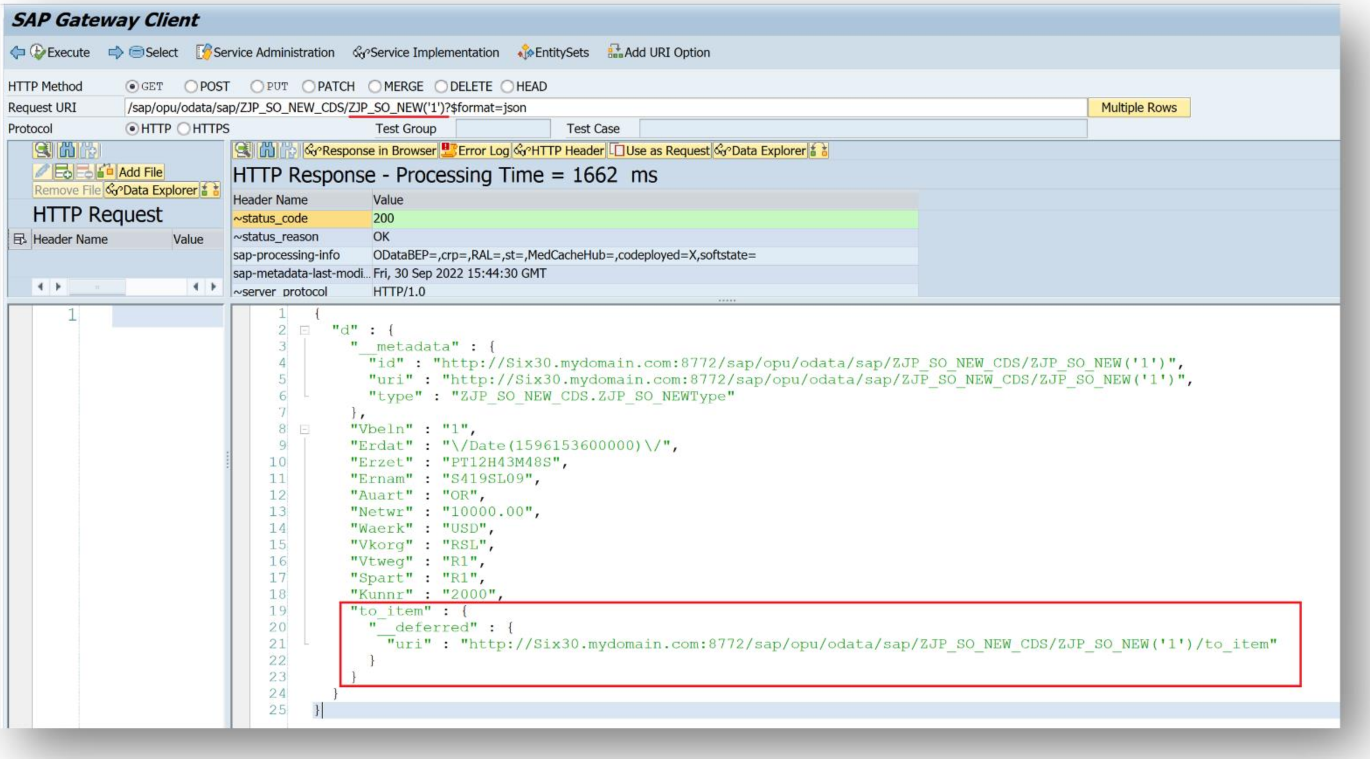
System Aliases

SAP System Alias	Description
LOCAL	Local System Alias

This service has 2 sets, ZJP_SO_NEW and VBAP. VBAP can be called directly or with Association.

URIs to test
/sap/opu/odata/sap/ZJP_SO_NEW_CDS/ZJP_SO_NEW?$format=json
/sap/opu/odata/sap/ZJP_SO_NEW_CDS/ZJP_SO_NEW(‘1’)?$format=json
/sap/opu/odata/sap/ZJP_SO_NEW_CDS/ZJP_SO_NEW(‘1’)/to_item?$format=json
/sap/opu/odata/sap/ZJP_SO_NEW_CDS/VBAP?$format=json
/sap/opu/odata/sap/ZJP_SO_NEW_CDS/ZJP_SO_NEW(‘1’)?$expand=to_item&$format=json

An example response with Read operation testing is shown as below.



HTTP Method ● GET ○ POST ○ PUT ○ PATCH ○ MERGE ○ DELETE ○ HEAD

Request URI /sap/opu/odata/sap/ZJP_SO_NEW_CDS/ZJP_SO_NEW('1')?$format=json

Protocol ● HTTP ○ HTTPS Test Group Test Case

HTTP Request

Header Name	Value

HTTP Response - Processing Time = 1662 ms

Header Name	Value
~status_code	200
~status_reason	OK
sap-processing-info	ODataBEP=,crp=,RAL=,st=,MedCacheHub=,codeployed=X,softstate=
sap-metadata-last-modi...	Fri, 30 Sep 2022 15:44:30 GMT
~server_protocol	HTTP/1.0

```
{
  "d" : {
    "__metadata" : {
      "id" : "http://Six30.mydomain.com:8772/sap/opu/odata/sap/ZJP_SO_NEW_CDS/ZJP_SO_NEW('1')",
      "uri" : "http://Six30.mydomain.com:8772/sap/opu/odata/sap/ZJP_SO_NEW_CDS/ZJP_SO_NEW('1')",
      "type" : "ZJP_SO_NEW_CDS.ZJP_SO_NEWType"
    },
    "Vbeln" : "1",
    "Erdat" : "\/Date(1596153600000)\/",
    "Erzet" : "PT12H43M48S",
    "Ernam" : "S419SL09",
    "Auart" : "OR",
    "Netwr" : "10000.00",
    "Waerk" : "USD",
    "Vkorg" : "RSL",
    "Vtweg" : "R1",
    "Spart" : "R1",
    "Kunnr" : "2000",
    "to_item" : {
      "__deferred" : {
        "uri" : "http://Six30.mydomain.com:8772/sap/opu/odata/sap/ZJP_SO_NEW_CDS/ZJP_SO_NEW('1')/to_item"
      }
    }
  }
}
```

SAP RAP Based OData Service –

SAP RAP, or the ABAP RESTful Application Programming Model, is a framework designed for efficient development within the SAP ecosystem, particularly for building enterprise SAP Fiori Apps and Web APIs. It's optimized for SAP HANA on SAP Business Technology Platform (BTP) ABAP Environment and SAP S/4HANA, accommodating both greenfield and brownfield implementations. RAP offers a streamlined approach to development, providing standardized architecture, native testability, and supportability. It supports various service types, including OData-based services for UI development and Web APIs, as well as analytical and SQL-based services.

To know how to develop an OData service based on SAP RAP, follow the series of ***blogs by Jagdish Patil***.

SAP CAPM Based OData Service –

SAP CAP, the SAP Cloud Application Programming Model, is a framework for building powerful business apps easily. It simplifies app development by offering tools and best practices, saving time and improving quality. CAP handles tasks like database interactions, creating web services, handling events, and security features like authentication. It's useful for various scenarios, from building new apps to extending existing ones. With CAP, developers can focus more on solving business problems and less on coding details.

To know how to develop an OData service based on SAP CAPM, check out this ***video from Anubhav Oberoy***.

Sachin Arte