



Former Member

September 20, 2016 | 7 minute read

Create Fiori app using CDS with BOPF- For beginners Part 1

Follow

💬 18 👍 32 👁 77,008

👍 Like

📡 RSS Feed

This blog is for developers who wants to get started with the UI5/Fiori + OData + CDS + BOPF.

Prerequisite:

- 1) Concepts of Core Data Services(CDS)
- 2) Basic Understanding of OData service
- 3) Concepts of BOPF
- 4) Conceptual knowledge of Fiori(Smart Template)

The Fiori/Ui5 is for the UI part, the CDS is for the data retrieval(code push down) while the BOPF is for handling DB activities.

Technically in the S4 Hana world, with the Code-Push down paradigm the intense business logic should happen in DB layer rather than in ABAP Application server. This blog is for pure test demo using CDS + BOPF to display the Fiori App. The performance optimization is not the intention here, but rather on how to use CDS + BOPF to get started with in S4 Hana Cloud.

Am using Web IDE environment with cloud connector to display the Fiori app. Let me quickly show how does the app look like.

Sales Order ID	Business Partner ID	Currency	Total Gross Amount	Total Net Amount	Total Tax Amount
500003423	CARMEN	INR	3,300.000	900.000	100.000
500003430	SOFTTECH	INR	1,000.000	900.000	100.000
500003440	SAP LABS	INR	1,100.000	900.000	200.000
500003443	INDIGO	INR	1,100.000	900.000	200.000
500003448	SPICEJET	INR	1,100.000	900.000	200.000
500003449	AIRINDIA	INR	1,100.000	900.000	200.000
1000000001	LUFTHANSA	INR	1,000.000	910.000	100.000
1000000006	EMIRATES	INR	1,000.000	900.000	100.000
1000000008	AIRASIA	INR	1,000.000	900.000	100.000

The **+** and **Delete** buttons are handled by BOPF framework while the other action is manually created in BOPF. The Business logic of this other action shall be done in an BOPF action class. We shall get into the details later on.

Lets get started with the development.

Initial set up required

1) You need HANA studio or Eclipse tool as we need to use ADT(ABAP Development Tool)

2) For hosting the app, Web IDE has been used. I recommend checking the [link](#) on how to install and set up the Web IDE trial and how to set up the Cloud connector.

3) System is S4 HANA ON PREMISE 1.0 with ABAP 7.50

For those who are using Hana account trial, the link in step 2 will take care of the web IDE including the cloud connector. Make sure the cloud connector is connected to the backend system as per the set up in step 2.

If the Cloud Connector set up is done, it should look something like this:

Cloud connector

Account Dashboard

Security Status

Account: [redacted]

Content

- Connector State
- Access Control
- Cookie Domains
- Service Channels
- Trusted Applications
- Principal Propagation

Connector State

Connector is operational since 11-Sep-2016 22:41:21

Landscape Host: hanatrial.ondemand.com

HTTPS Proxy: proxy.8080

System Certificate: [redacted]

Account User: [redacted]

Location ID: n.a.

Connection Information

Status: CONNECTED

Tunnel ID: [redacted]

Connector ID: 191B3AF0E03D11E5905BE4D80A352787

Remote Name: connectivitynotification.hanatrial.ondemand.com

Local Name: [redacted]

Local IP: 192.168.0.144

Connections (0)

Account Dashboard

Security Status

Account: [redacted]

Content

- Connector State
- Access Control
- Cookie Domains
- Service Channels
- Trusted Applications
- Principal Propagation
- Audit
- Logs

Mapping Virtual To Internal System

Add... Edit... Import... Export Delete Check

State	Virtual Host	Internal Host	Protocol	Back-end Type
●	[redacted]:44300	[redacted]:44300	HTTPS	ABAP System
◇	[redacted]:44300	[redacted]:44300	HTTP	ABAP System

Resources Accessible On [redacted]:44300

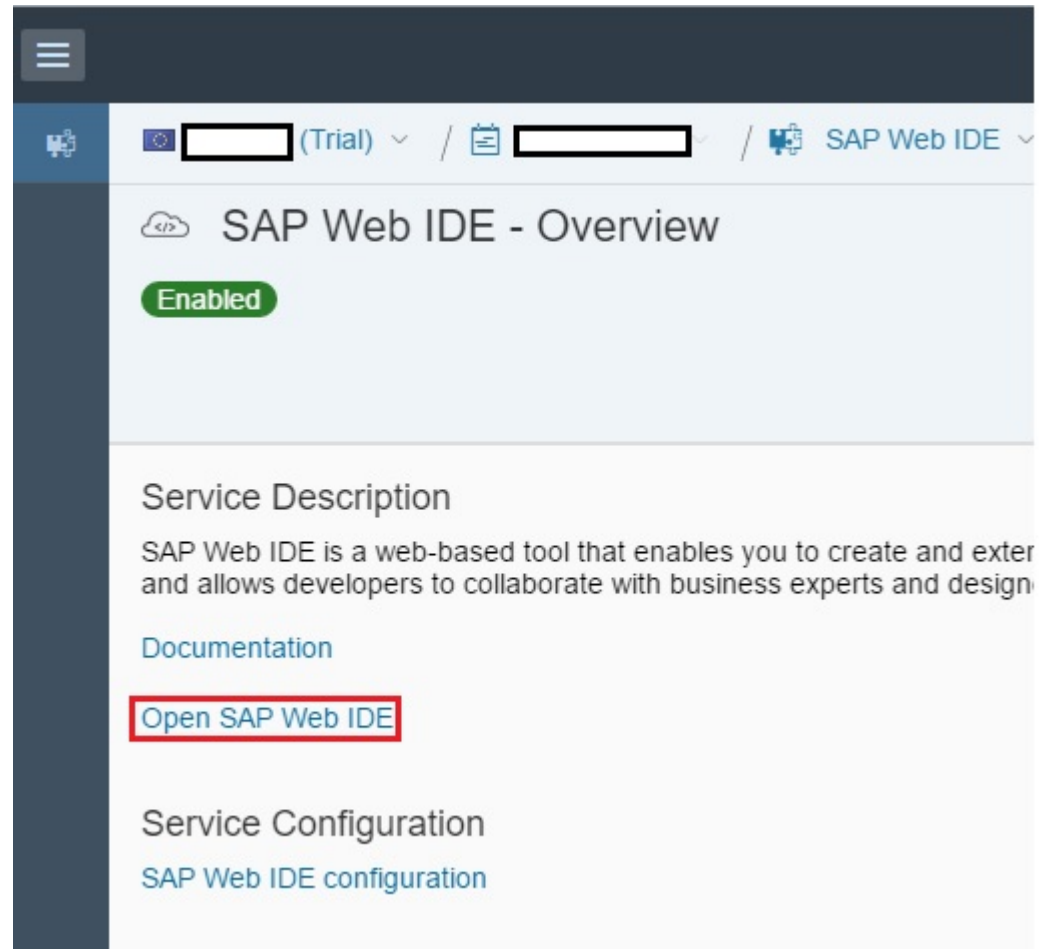
Add... Edit... Enable Disable Delete

State	URL Path	Access Policy
●	/sap/bc/adt	Path and all sub-paths
●	/sap/bc/ui5_ui5	Path and all sub-paths
●	/sap/opu/odata	Path and all sub-paths

Web IDE

After logging into HANA account trial and go to services and click on Web IDE.

Open Web IDE now.



We shall be creating an sales order app as an demo example.

Basic steps involved in creating this app are :

- Create 2 Consumption CDS views (Sales Order header & Sales Order Items)
- Use annotation (Consumption view) for generating OData service
- Use annotations (Basic views) for generating BOPF objects
- Use Fiori Smart Template to display Sales order application

Technically at a higher level :

- The Fiori app shall consume the OData
- The OData has data source as CDS views (ie consumption CDS)
- The BOPF takes care of table CRUD operations. Here the BOPF objects are generated from the CDS views via annotations.

Lets start with the CDS views.

Basically in the CDS, there are consumption CDS, Basic CDS and Composite CDS. Consumption CDS are exposed to the UI. The intention is not to get into CDS details here, but to give a quick information on CDS.

We have 2 consumption CDS view's. And the Header Consumption view shall be used by Fiori. For simplicity sake, the naming convention shall be as :

Consumption CDS with *_C_*

Basic CDS with *_I_*

1) Header Consumption ZDEMO_C_SALESORDER

1.1) The SO header consumption view utilize the Basic CDS view ZDEMO_I_SALESORDER

1.2) The Basic view use a Header Table "ZPROTO_SO_A". This shall form the BOPF root table for SO Header

2) Item consumption view is ZDEMO_C_SALESORDER_ITEM

2.1) The Item consumption view use Basic CDS view ZDEMO_I_SALESORDER_ITEM

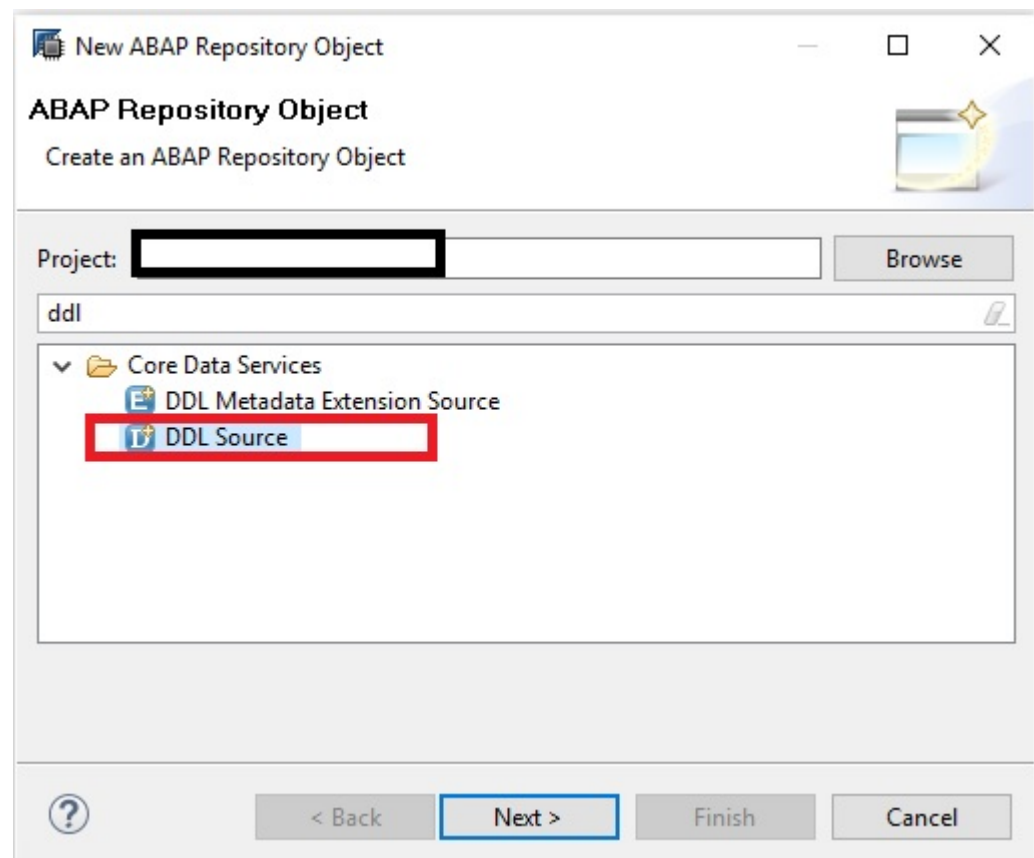
2.2) The Basic view use the Item table ZPROTO_SOI_A which shall form the BOPF item node table

Now, we can use these tables in Basic CDS views.

Lets start creating with Basic Item CDS view

Basic SO Item CDS

For creating CDS view, in the HANA studio, go to ABAP perspective.
Choose your system and create a Package underneath. Right click on the package and choose New -> Other repository object.



New DDL Source

DDL Source
Create a DDL Source

Project: * ER9_001_appukuttan_en_1 Browse...

Package: * \$TMP Browse...

☐ Add to favorite packages

Name: * ZDEMO_I_SALESORDER_ITEM

Description: * ZDEMO_I_SALESORDER_ITEM

Original language: EN

? < Back Next > Finish Cancel

Give the name of the view and click on Finish.

The CDS + annotations used are:

```

1 @AbapCatalog.sql View Name: 'ZDDL_I_SOI16' DDIC SQL View
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_REQUIRED
4 @EndUserText.label: 'ZDEMO_I_SALESORDER_ITEM'
5
6 @Search.searchable: true
7
8 //BOPF
9 @ObjectModel:{
10     writeActivePersistence: 'ZPROTO_SOI_A',      Item DB Table
11     semanticKey: ['SalesOrderID', 'SalesOrderItemID'], Item table key
12     representativeKey: 'SalesOrderItemID',      for BOPF instance
13     createEnabled: true,                          identification
14     updateEnabled: true,      CRUD enabled for BOPF
15     deleteEnabled: true
16 }

```



```

18 //CDS View
19 define view Zdemo_I_Salesorder_Item Item CDS
20
21 as select from zproto_soi_a as Item
22
23 //Association to Header Header Association
24 association [1..1] to Zdemo_I_Salesorder as _SalesOrder on $projection.SalesOrderID = _SalesOrder.SalesOrderID
25
26 {
27     //Sales order
28     @Search.defaultSearchElement: true
29     @ObjectModel.readOnly: true
30     key Item.salesorderid as SalesOrderID,
31
32     //Sales order item
33     @Search.defaultSearchElement: true
34     key Item.salesorderitemid as SalesOrderItemID,
35
36     //Product
37     @Search.defaultSearchElement: true
38     Item.productid as ProductID,
39
40     //Measure fields
41     Item.currencycode as CurrencyCode,
42     Item.grossamount as GrossAmount,
43     Item.netamount as NetAmount,
44     Item.taxamount as TaxAmount,
45
46     //BOFF Association to SO Header
47     @ObjectModel.association.type: [#TO_COMPOSITION_PARENT,#TO_COMPOSITION_ROOT]
48     _SalesOrder Declare header association
49 }

```

Here the association between item and header imply an LEFT OUTER JOIN. If you want to force an inner join with association, example you can do something like this for example

```

46 //BOFF Association to SO Header
47 @ObjectModel.association.type: [#TO_COMPOSITION_PARENT,#TO_COMPOSITION_ROOT]
48 _SalesOrder[1:inner].BusinessPartnerID
49 }

```

For more clarity on path expression joins, pls refer to [here](#)

Save and activate the ZDEMO_I_SALESORDER CDS view.

Post activation, following objects are created in Dictionary :

- 1) DB view : ZDEMO_I_SALESORDER_ITEM (CDS Entity)
- 2) DD SQL view (Columnar) : ZDDL_I_SOI16

Since this Basic view cannot be used directly for UI display, an Consumption view shall be created.

Consumption SO Item CDS

The CDS code + Annotations are as below:

```
1 @AbapCatalog.sqlViewName: 'zddl_c_so116' DDIC SQL View
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_REQUIRED
4 @EndUserText.label: 'zdemo_c_salesorder_item'
5
6 @Search.searchable: true
7 @UI.headerInfo.typeName: 'Sales Order Item'
8 @UI.headerInfo.title.value: 'SalesOrderItemID'
9
10 @ObjectModel.transactionalProcessingDelegated: true
11 @ObjectModel.semanticKey: ['SalesOrderID', 'SalesOrderItemID'] — Item table Keys
12 @ObjectModel.representativeKey: 'SalesOrderItemID' — BO Instance key identification
13 @ObjectModel.createEnabled: true
14 @ObjectModel.updateEnabled: true BO Change/Update/enabling
15 @ObjectModel.deleteEnabled: true
16
17 //CDS view
18 define view Zdemo_C_Salesorder_Item
19 as select from Zdemo_I_Salesorder_Item — Item Basic View
20 //Association to SO header
21 association [1..1] to Zdemo_C_Salesorder as SalesOrder on $projection.SalesOrderID = _SalesOrder.SalesOrderID
22 — Item to Header Association
23 {
24 //Sale order
25 @Search.defaultSearchElement: true
26 @ObjectModel.readOnly: true -- not for edit
27 @UI.identification: (position: 10, importance: #HIGH) -- to be seen during edit
28 @UI.lineItem: (position: 10, importance: #HIGH) -- field position in UI app
29 key Zdemo_I_Salesorder_Item.SalesOrderID,
30
31 //Sale order item
32 @Search.defaultSearchElement: true
33 @ObjectModel.readOnly: true
34 @UI.identification: (position: 20, importance: #HIGH)
35 @UI.lineItem: (position: 20, importance: #HIGH)
36 key Zdemo_I_Salesorder_Item.SalesOrderItemID,
37
38 //Currency code
39 @UI.identification: (position: 70, importance: #HIGH)
40 @UI.lineItem: (position: 70, importance: #HIGH)
41 Zdemo_I_Salesorder_Item.CurrencyCode,
42
43 //Gross Amount
44 @UI.identification: (position: 80, importance: #HIGH)
45 @UI.lineItem: (position: 80, importance: #HIGH)
46 Zdemo_I_Salesorder_Item.GrossAmount,
47
48 //Net Amount
49 @UI.identification: (position: 90, importance: #HIGH)
50 @UI.lineItem: (position: 90, importance: #HIGH)
51 Zdemo_I_Salesorder_Item.NetAmount,
52
53 //Tax Amount
54 @UI.identification: (position: 100, importance: #HIGH)
55 @UI.lineItem: (position: 100, importance: #HIGH)
56 Zdemo_I_Salesorder_Item.TaxAmount,
57
58 //BOFF association
59 @ObjectModel.association.type: [#TO_COMPOSITION_ROOT, #TO_COMPOSITION_PARENT]
60 _SalesOrder — Item to Header Association
61 }
```

Save and Activate.

Post activation the objects created are in Dictionary :

- 1) DB view – ZDEMO_C_SALESORDER_ITEM
- 2) DD SQL View – ZDDL_C_SOI16

We are done with sales order Item.

Lets move to Sales Order Header starting with Basic view.

SO Header Basic CDS ZDEMO_I_SALESORDER

The CDS code + Annotations are:

```
1 @AbapCatalog.sqlViewName: 'zddl_i_so16'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_REQUIRED
4 @EndUserText.label: 'zdemo_i_salesorder'
5 @Search.searchable: true
6
7 //BOPF
8 @ObjectModel: {
9     compositionRoot: true,
10    transactionalProcessingEnabled: true,
11    writeActivePersistence: 'ZPROTO_SO_A',
12    semanticKey: 'SalesOrderID',
13    representativeKey: 'SalesOrderID',
14    createEnabled: true,
15    updateEnabled: true,
16    deleteEnabled: true
17 }
```

Annotations:

- BOPF root node** (points to line 9)
- Root table** (points to line 11)
- Table Key field** (points to line 12)
- Identify BOPF instance key** (points to line 13)
- BOPF enabled for CRUD** (points to line 14)

Optional: there is an Object model annotation for modelCategory: #BusinessObject. If we use this the create sales order logic has to be written as an BOPF action. And this action shall be triggered when the + icon is clicked.

```

19 //CDS View
20 define view Zdemo_I_Salesorder
21
22     as select from zproto_so_a as SalesOrder
23
24 //Association between SO header and item Item Table Association
25 association [0..*] to Zdemo I Salesorder Item as _Item on $projection.SalesOrderID = _Item.SalesOrderID
26
27 {
28     //Sale order
29     @Search.defaultSearchElement: true -- to be appeared as search in app
30
31     key SalesOrder.salesorderid                as SalesOrderID,
32
33     //Business Partner
34     @Search.defaultSearchElement: true
35     SalesOrder.businesspartnerid            as BusinessPartnerID,
36
37     //Currency Code
38     SalesOrder.currencycode                  as CurrencyCode, Fields to be used for
39                                                     consumption
40
41     //Gross amount
42     SalesOrder.grossamount                    as GrossAmount,
43
44     //Net Amount
45     SalesOrder.netamount                      as NetAmount,
46
47     //Tax amount
48     SalesOrder.taxamount                      as TaxAmount,
49
50     //Lifecycle status
51     SalesOrder.lifecyclestatus                as LifecycleStatus,
52
53     //SO Item display in detail page - used for navigation
54     @ObjectModel.association.type: #TO_COMPOSITION_CHILD
55     _Item Association/Navigation

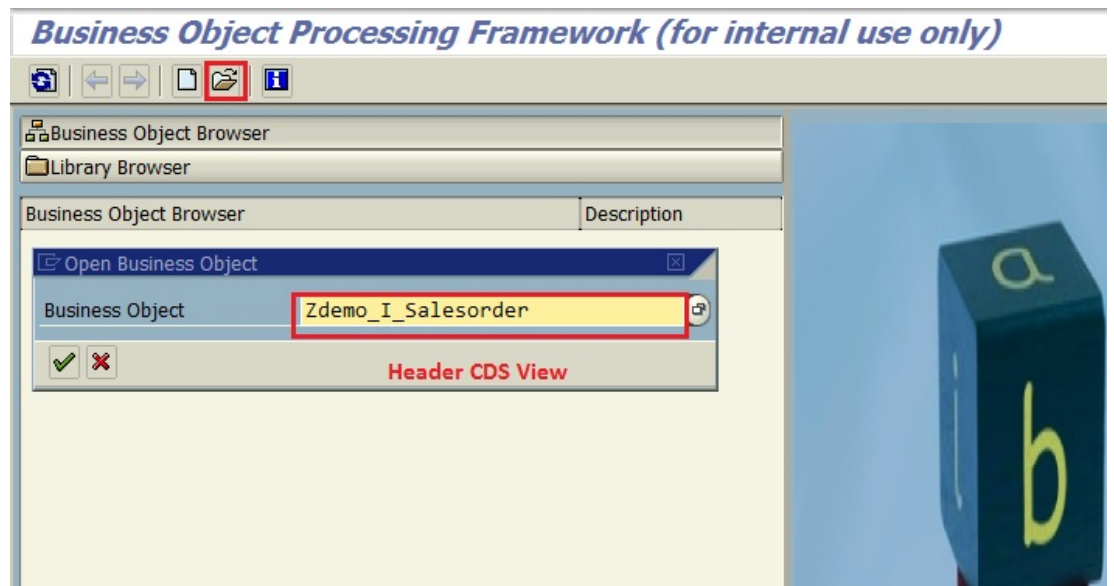
```

Save and Activate.

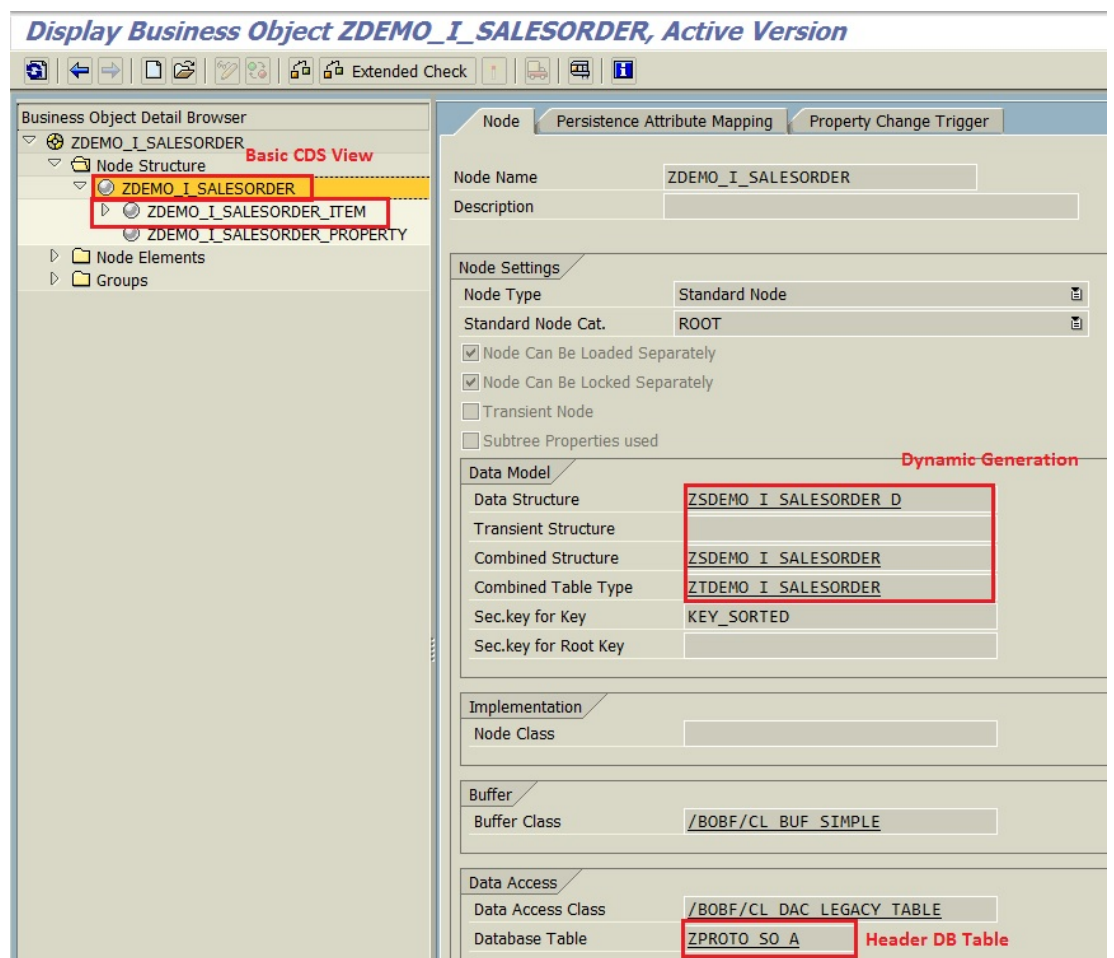
Post activation, following objects are generated :

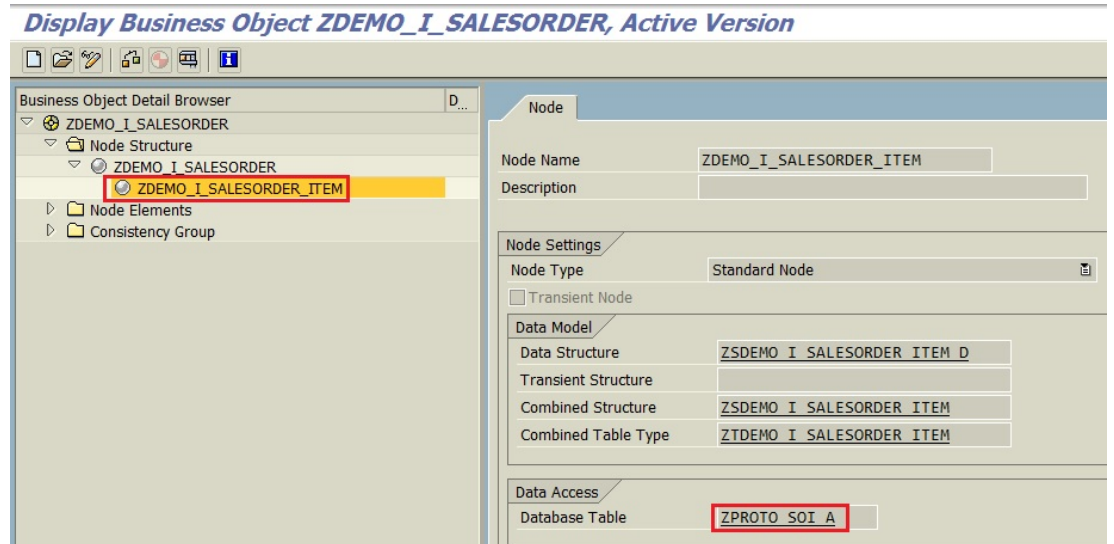
- 1) DD Sql view ZDDL_I_SO16
- 2) DB view ZDEMO_I_SALESORDER
- 3) And an BOPF Business Object by the CDS name ie Business Object **ZDEMO_I_SALESORDER**

The BOPF Business Object can be viewed in Hana Studio/Eclipse via ABAP perspective. Go to the package, under Business Objects and here you shall see the generated BOPF object. Other option is via SAP Gui. You can open GUI in studio as well. Use Tcode BOBX or /BOBF/CONF_UI.



BOPF Business Object





Now lets use this Basic Header view in a Consumption Header View

Consumption SO Header CDS ZDEMO_C_SALESORDER

The CDS code + Annotations are:

```

1 @AbapCatalog.sqlViewName: 'zddl_c_so16' DDIC Sql View
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_REQUIRED
4 @EndUserText.label: 'zdemo_c_salesorder'
5 @Search.searchable: true
6
7 //Odata
8 @OData.publish: true Publish OData
9
10 //BOPF
11 @ObjectModel.transactionalProcessingDelegated: true
12 @ObjectModel.semanticKey: 'SalesOrderID'
13 @ObjectModel.representativeKey: 'SalesOrderID' BOPF related
14 @ObjectModel.createEnabled: true
15 @ObjectModel.updateEnabled: true
16 @ObjectModel.deleteEnabled: true
17
18
19 @UI.headerInfo.typeName: 'Sales Order'
20 @UI.headerInfo.title.value: 'SalesOrderID'
21

```



```

22 //CDS View
23 define view Zdemo_C_Salesorder as Basic CDS - SO Header
24 as select from Zdemo_C_Salesorder
25 association [1..*] to Zdemo_C_Salesorder-Item as _Item on $projection.SalesOrderID = _Item.SalesOrderID
26 Consumption CDS - SO Item
27 {
28   BUI.LineItem(position: 10
29   BUI.LineItem: {
30     type: #FOR_ACTION, position: 1, dataAction: BOPF:SET_LIFECYCLE_STATUS_TO_PAID, label: 'Set Lifecycle status to Pa
31   }
32   $search.defaultSearchElement: true
33   BUI.identification: (position: 10, importance: #HIGH1)
34   BUI.selectionField: (position: 10)
35   key Zdemo_C_Salesorder.SalesOrderID,
36
37   $search.defaultSearchElement: true
38   BUI.identification: (position: 20, importance: #HIGH2)
39   BUI.LineItem: (position: 20, importance: #HIGH)
40   BUI.selectionField: (position: 20)
41   Zdemo_C_Salesorder.BusinessPartnerID,
42
43   BUI.identification: (position: 30, importance: #HIGH3)
44   Zdemo_C_Salesorder.CurrencyCode,
45
46   BUI.LineItem: (position: 40, importance: #HIGH)
47   BUI.identification: (position: 40, importance: #HIGH4)
48   BUI.dataPoint: (title: 'GrossAmount')
49   Zdemo_C_Salesorder.GrossAmount,
50
51   BUI.LineItem: (position: 50, importance: #HIGH5)
52   BUI.dataPoint: (title: 'NetAmount')
53   Zdemo_C_Salesorder.NetAmount,
54
55   BUI.LineItem: (position: 60, importance: #HIGH)
56   BUI.dataPoint: (title: 'TaxAmount')
57   Zdemo_C_Salesorder.TaxAmount,
58
59   BUI.LineItem: (position: 70, importance: #HIGH)
60   BUI.dataPoint: (title: 'LifecycleStatus')
61   Zdemo_C_Salesorder.LifecycleStatus,
62
63   BUI.publish: { [qualifier: 'Item', groupLabel: 'Item', position: 40, importance: #HIGH]
64   $projection.association-type: #FOR_COMPOSITION_CHILD
65   Item To Item Navigation
66 }

```

Standard						Set Lifecycle status to Paid		Delete	+	
1 Sales Order ID	2 Business Partner ID	3 Currency	4 Total Gross Amount	5 Total Net Amount	Total Tax Amount	SO Lifecycle Status				
<input type="radio"/> 500003423	CARMEN	INR	3,300.000	900.000	100.000	P				
<input type="radio"/> 500003430	SOFTTECH	INR	1,000.000	900.000	100.000	P				
<input type="radio"/> 500003440	SAP LABS	INR	1,100.000	900.000	200.000	P				
<input type="radio"/> 500003443	INDIGO	INR	1,100.000	900.000	200.000	P				
<input type="radio"/> 500003448	SPICEJET	INR	1,100.000	900.000	200.000	P				
<input type="radio"/> 500003449	AIRINDIA	INR	1,100.000	900.000	200.000	P				
<input type="radio"/> 1000000001	LUFTHANSA	INR	1,000.000	910.000	100.000	P				
<input type="radio"/> 1000000006	EMIRATES	INR	1,000.000	900.000	100.000	P				
<input type="radio"/> 1000000008	AIRASIA	INR	1,000.000	900.000	100.000	P				

Save and Activate.

Post activation, the following objects are created:

- 1) DD Sql View DDL_C_SO16
- 2) DB view ie CDS entity ZDEMO_C_SALESORDER (DDIC)
- 3) OData service (<CDS view>_CDS)

```

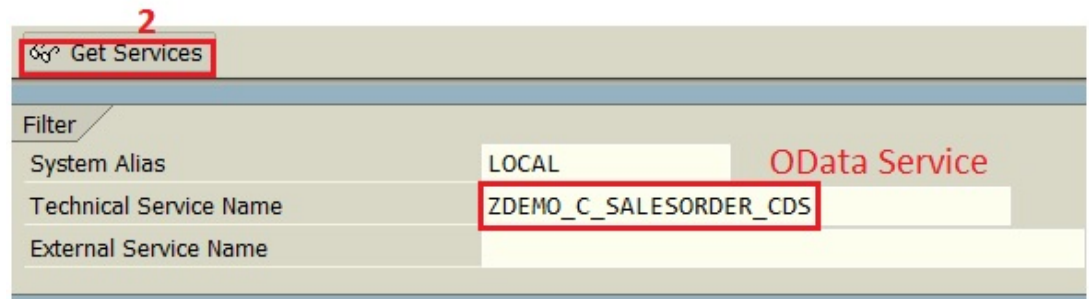
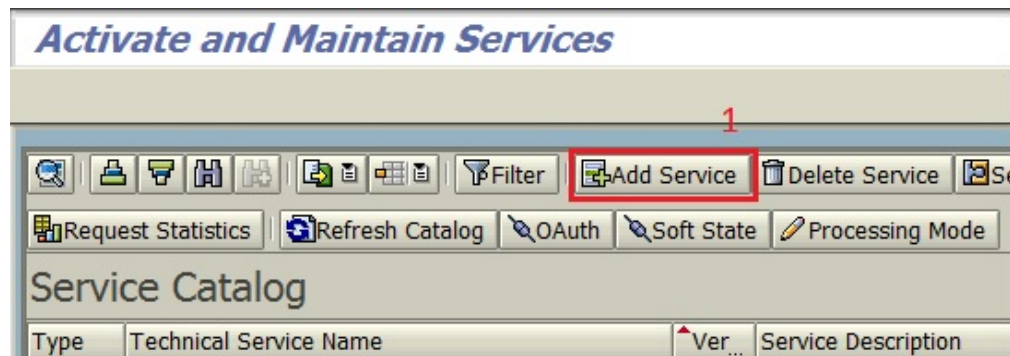
6
7 //Odata
8 @Odata.publish: true
9

```

Click to find the Generated OData service

The OData service generated is not active. To activate go to Tcode Gui Tcode /IWFND/MAINT_SERVICE.

Click on Add service. Choose Local and enter the Name of CDS in the Technical Service and click Get services.

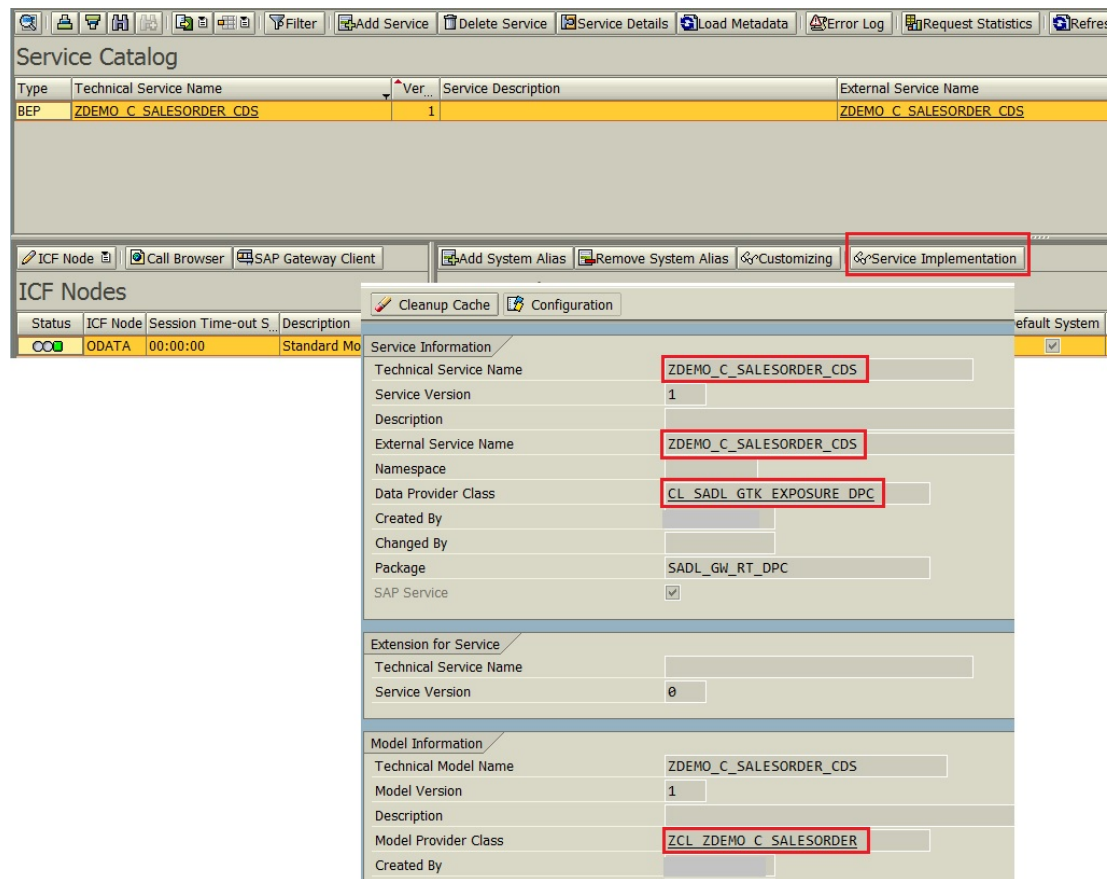


Click on “Get Services. Click on Service or click on “Add Selected Service”. In the next screen enter the package or local object and click ok. The OData service is now **Active**.

In the background when the CDS view with OData annotations get activated, the SADL generates Gateway artifacts such as Model Provider Class (MPC) and Data Provider Class (DPC) which form the backbone of OData.

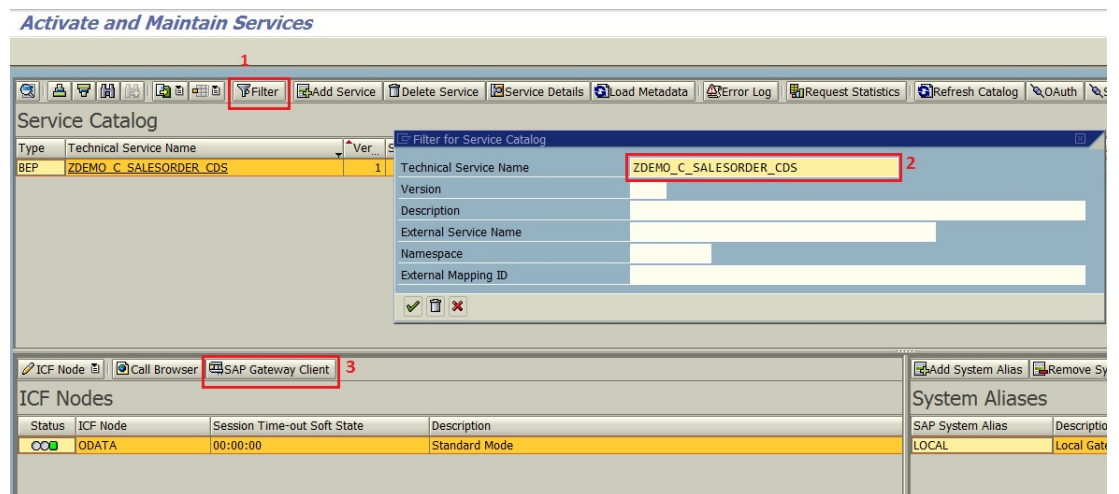
Normally to create an OData service we go to Tcode SEGW and create a project and a Data Model(Data Reference as Final Consumption CDS view) and the MPC and DPC are generated.

In our case we can view these artifacts as shown below.



Test the Odata service.

Go back to the service maintenance screen. Click on Filter and enter the service name as Technical service in our case ZDEMO_C_SALESORDER_CDS



Click on Gateway Client. In the new window click on execute. If the status is 200, all is well.

SAP Gateway Client

Execute Select Service Administration Service Implementation EntitySets Add URI Option

HTTP Method: GET POST PUT PATCH MERGE DELETE HEAD

Request URI: /sap/opu/odata/sap/ZDEMO_C_SALESORDER_CDS/?\$format=xml

Protocol: HTTP HTTPS

Test Group Test Case

Response in Browser Error Log HTTP Header Use as Request Data Explorer

HTTP Response - Processing Time = 1688 ms

Header Name	Value
~status_code	200
~status_reason	OK
sap-processing-info	microhub=,crp=,st=,MedCacheHub=,codeployed=X,softstate=
last-modified	Sat, 10 Sep 2016 17:21:27 GMT
sap-metadata-last-modi...	Sat, 10 Sep 2016 17:21:27 GMT
content-type	application/xml

1

```
<?xml version="1.0" encoding="UTF-8"?>
- <app:service xmlns: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="https://ldai5er9.wdf.sap.corp:44300/sap/opu/odata/sap/ZDEMO_C_S
  - <app:workspace>
    <atom:title type="text">Data</atom:title>
    - <app:collection href="Zdemo_C_Salesorder" sap:content-version="1" sap:search
      <atom:title type="text">Zdemo_C_Salesorder</atom:title>
      <sap:member-title>zdemo_c_salesorder</sap:member-title>
      <atom:link title="searchZdemo_C_Salesorder" type="application/opense
        href="Zdemo_C_Salesorder/OpenSearchDescription.xml" rel="search
      </app:collection>
    - <app:collection href="Zdemo_C_Salesorder_Item" sap:content-version="1" sap
      <atom:title type="text">Zdemo_C_Salesorder_Item</atom:title>
      <sap:member-title>zdemo_c_salesorder_item</sap:member-title>
      <atom:link title="searchZdemo_C_Salesorder_Item" type="application/o
        href="Zdemo_C_Salesorder_Item/OpenSearchDescription.xml" rel="
      </app:collection>
    </app:workspace>
    <atom:link href="https://ldai5er9.wdf.sap.corp:44300/sap/opu/odata/sap/ZD
    <atom:link href="https://ldai5er9.wdf.sap.corp:44300/sap/opu/odata/sap/ZD
```

The OData metadata is fine. Test if the OData returns records by choosing the entity sets.

Execute Select Service Administration Service Implementation EntitySets Add URI Option

HTTP Method: GET POST PUT PATCH MERGE DELETE HEAD

Request URI: /sap/opu/odata/sap/ZDEMO_C_SALESORDER_CDS/Zdemo_C_Salesorder

Protocol: HTTP HTTPS

Test Group Test Case

Response in Browser Error Log HTTP Header Use as Request Data Explorer

HTTP Response - Processing Time = 442 ms

Header Name	Value
~status_code	200
~status_reason	OK
sap-processing-info	microhub=,crp=,st=,MedCacheHub=SHM,MedCacheBEP=Table,codeployed=X,softstate=
sap-metadata-last-modi...	Sat, 10 Sep 2016 17:21:27 GMT
~server_protocol	HTTP/1.0
content-type	application/atom+xml;type=feed; charset=utf-8

HTTP Re...

So we are done with Final Consumption CDS view. And we can use this view in Fiori Smart Template.

Recap of what we have done so far.

- 1) Created Basic Item CDS
- 2) Created Consumption Item CDS
- 3) Created Basic Header CDS
- 4) Created Consumption Header CDS

Before we start with Fiori App, we shall create BOPF action. The action shall change the Lifecycle Status of an Sales order



The BOPF action and the App display will be covered in the [Part 2](#).

Alert Moderator

Assigned tags

ABAP Development

SAP Fiori

SAP Fiori for SAP ERP

SAP S/4HANA

abap cds

abap cds views

abap for hana

[View more...](#) >

Similar Blog Posts

[Create Fiori app using CDS with BOPF- For beginners Part 2](#)

By Former Member Sep 20, 2016

[Creating a draft enabled Sales Order Fiori App using the new ABAP Programming Model - Part 1: Overview](#)

By Geert-Jan Klaps Mar 11, 2019

[Getting Started with ABAP Core Data Services \(CDS\)](#)

By Carine Tchoutouo Djomo Feb 01, 2016

Related Questions

[CDS + BOPF + UI5. Call BAPI?](#)

By Suwandi Cahyadi Jun 24, 2019

[Why need multiple authorization check for FIORI app?](#)

By Akshath LT Dec 14, 2018

[Interactive / Draft Fiori Apps in NW 7.51 SP1 ?](#)

By Attila Berencsi Feb 08, 2017

Join the Conversation



[SAP TechEd](#)

Tune in for tech talk. Stay for inspiration. Upskill your future.



[Coffee Corner](#)

Join the new Coffee Corner Discussion Group.

18 Comments

You must be [Logged on](#) to comment or reply to a post.



Former Member

October 19, 2016 at 9:14 am

hi sujin,

thanks for this tutorial. i tried to do all the steps but if i activate the basic cds view of sales order in eclipse there is no business object generated by the BOPF runtime.

Do you know why this could happen?

Like 1 | Share



Former Member | Blog Post Author

November 2, 2016 at 6:02 am

Hi Hergen,

I can think of 2 reasons why the BO must not be generated:

1) the use of @ObjectModel annotations as shown below.

```
1 @AbapCatalog.sqlViewName: 'zddl_i_so16'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_REQUIRED
4 @EndUserText.label: 'zdemo_i_salesorder'
5 @Search.searchable: true
6
7 //BOPF
8 @ObjectModel: {
9     compositionRoot: true,
10    transactionalProcessingEnabled: true,
11    writeActivePersistence: 'ZPROTO_SO_A',
12    semanticKey: 'SalesOrderID',
13    representativeKey: 'SalesOrderID',
14    createEnabled: true,
15    updateEnabled: true,
16    deleteEnabled: true
17 }
```

Annotations explained:

- compositionRoot:** BOPF root node
- transactionalProcessingEnabled:** true
- writeActivePersistence:** 'ZPROTO_SO_A', Root table
- semanticKey:** 'SalesOrderID', Table Key field
- representativeKey:** 'SalesOrderID', Identify BOPF instance key
- createEnabled:** true
- updateEnabled:** true, BOPF enabled for CRUD
- deleteEnabled:** true

2) Make sure that the DB table (in this example ZPROTO_SO_A) is available in the dictionary.

If the BO is not generated, the error messages shown during CDS activation. You can see these when you mouse over the icon next to @objectmodel. Good luck !

Regards,
Sujin

Like 0 | Share



Jason Muzzy

March 10, 2021 at 12:55 am

I realize this question is old, but I found it while looking for a solution to the same problem. I noticed that if I activated the CDS view as a local object then it would generate the BOPF object. Based on that I found and implemented note 2948149 which allowed the BOPF object to generate even though the package was assigned to transport layer SAP.

Like 0 | Share



Jason Scott

October 20, 2016 at 3:16 am

This is a great simple example... but what has me scratching my head is that this whole CDS/BOPF thing is designed around completely separate custom data tables.

The reality of real-life is that you nearly always need to integrate with standard sap business processes.. For example you will need to call BAPI's to post purchase orders or goods movements or whatever... Not just on the write-side either - often its far easy to get the correct data out of the system by calling a bapi like BAPI_PR_GETDETAIL for example.

It seems that this technique of using CDS annotations to auto-generate a BOPF object will not work for these more real-life scenarios... Or am I missing something and it is just that all tutorials on this subject are extremely basic using a Z table?!?

Is there an "Extension" concept for these generated BOPF objects so that you can code your own ABAP update routines?

Because... if there isn't then this whole CDS/BOPF is purely academic. Great for creating mars explorer demos at a TechEd keynote... but thats about it...

Like 1 | Share



Former Member | Blog Post Author

November 2, 2016 at 5:54 am

Hi Jason,

Thank you for your comment. You are right with the real time reality scenarios. Here's my take on that.

With the new code to data Paradigm in place and with the S/4 Architecture along side, seems the CDS-BOPF combination will persists for long now.

Lets say for an write scenario, the CDS-BOPF will work just perfectly as BOPF framework takes care of the DB write/lock/buffer/validation/consistency. In a typical real time scenario, say we have some ERP tables in S4. For query purpose the CDS should work faster than a BAPI call, because the CDS is interacting with the DB layer.

Regarding extension and ABAP routines am not sure. But you can do something like this as mentioned in the Blog which is Generate the BOPF via annotations and then go to the BOPF framework and edit to add actions/determinations/validations etc.

On another note, for query/read, there are basically 2 approaches keeping in mind the code-to-data concept.

Top down approach – using CDS to interact with DB and

Bottom up approach – using HANA modelling and using the HANA artifacts for reports/query.

Regards,
Sujin

Like 1 | Share



Florian Royer

December 15, 2017 at 11:22 pm

First of all, thank you Sujin for your detailed guide !

This is exactly what I am thinking about when I hear buzzwords like BOPF & CDS in one sentence. Still cannot figure out, when to use what.

My one and only productive use of a BOPF Object was:

- Z-Table HEAD
- Z-Table ITEM

Business logic in BOPF object, vizualisation in FPM, generated by Wizards. Very simple, but yet very powerful compared to other development approaches (to be honest – I think I wrote about 200 lines of code in total, only because I wanted to implement a Drag and Drop feature, otherwise it would be less code).

I agree per 100% to Jason's post. When can I use BOPF / CDS in a real-life scenario, where I can't base on Z-Tables, like create a purchase order using custom BOPF?

The answer to this question is still open, looking forward to your replies.

Like 0 | Share



Former Member

March 13, 2017 at 5:16 am

Hi Sujin,

Thanks for the so cleared blog about cds-bopf.

Like 0 | Share



Former Member

April 21, 2017 at 1:18 pm

Hi Suijin,

thanks for the tutorial. What I need to do that I can see "Smart Template" as Project Type in SAP WebIDE? I don't get this option.

Is this already available for Partners/Ccustomers?

Thomas

Like 0 | Share



Cain Sun

May 26, 2017 at 6:19 am

Thanks for the tutorial, I noticed that the DB table for SO header does not have GUID and NODE GUID fields, how can BOPF store the data? In fact, I encounter some errors during testing the result.

Like 0 | Share

Sebastian Freilinger-Huber

February 19, 2018 at 3:40 pm

Hi Cain,

as far as I know in classical BOPF you need GUIDs on the database. With BOPF in S/4 context now this has been reworked and improved by SAP.

In case for example your database table contains a common ID field or some other sort of key (instead of a GUID), you can generate the BOPF object on it as well using the corresponding CDS annotations (ObjectModel). From my understanding to make this work the ID has to be defined as key field in the CDS view, which is used for the BOPF generation.

Best regards,

Sebastian

Like 0 | Share

Joseph BERTHE

October 28, 2017 at 6:19 pm

Hello,

Thanks for your blog, I try to follow it and I encounter some difficulties 😊 During the activation of the zdemo_i_SalesOrder CDS it through me this error :

[BO check] Element SALESORDERITEMID is not a DB field (view ZDEMO_I_SALESORDER_ITEM, table ZPROTO_SOI_A)

As you can imagine, the field **SALESORDERITEMID** is in the table.

What could be the problem ?

Regards

Like 1 | Share



Former Member

November 18, 2017 at 8:21 pm

Hello,

Thanks for the blog.

I would like to try the tutorial, but i don't know where i get the tables mentioned in the tutorial. So far i only know sflight.

Regards

Dennis

Like 0 | Share

Sourabh Gandhi

November 30, 2017 at 7:51 am

Thanks Sujin. This was very helpful. Please provide more complex scenarios. Regards Prateek Sonthalia

Like 0 | Share

Sebastian Freilinger-Huber

February 19, 2018 at 3:46 pm

Hi Sujin,

thanks for this blog. Just one question considering the maintenance of the BO Objects.

Is there a reason why you don't use Eclipse as you are already working on a 7.50 system?

Best regards,

Sebastian

Like 0 | Share

Terry Huang

March 9, 2018 at 1:57 am

Hi,

why we need to create seprate cusumption and basic views, I just tried only create two basic views. it wroks properly.

Like 0 | Share



I'm trying to consume the service to insert the detail and I'm getting the follow

SAP Gateway Client

Last Request Execute Next Request Select Service Administration Service Implementation EntitySets Add URI Option Más >

HTTP Method: GET POST PUT PATCH MERGE DELETE HEAD

Request Uri: /sap/opu/odata.sap/ZDEMO_C_SALESORDER_CDS/ZDEMO_C_SALESORDER_ITEM Multiple Rows

Protocol: HTTP HTTPS Test Group: Test Case:

Header Name	Value
<?xml version='1.0' encoding='utf-8'?><entry xml:base='http://www.w3.org/2005/Atom' xmlns='http://schemas.microsoft.com/ado/2007/08/dataservices/metadata' xmlns:d='http://schemas.microsoft.com/ado/2007/08/dataservices'><id>http://S4HNDEVIPIC.cmh.com.pe:8006/sap/opu/odata.sap/ZDEMO_C_SALESORDER_CDS/ZDEMO_C_SALESORDER_ITEM(SalesOrderID=1',SalesOrderItemID=1')/</id><title type='text':ZDEMO_C_SALESORDER_ITEM(SalesOrderID=1',SalesOrderItemID=1')</title><updated>2019-03-06T15:20:19Z</updated><category term='ZDEMO_C_SALESORDER_CDS.ZDEMO_C_SALESORDER_ITEMType' scheme='http://schemas.microsoft.com/ado/2007/08/dataservices/schema'/><link href='/ZDEMO_C_SALESORDER_ITEM(SalesOrderID=1',SalesOrderItemID=1')' rel='edit' title='ZDEMO_C_SALESORDER_ITEMId'/><link href='/ZDEMO_C_SALESORDER_ITEM(SalesOrderID=1',SalesOrderItemID=1')/'to_SalesOrder'' rel='http://schemas.microsoft.com/ado/2007/08/dataservices/related/to_SalesOrder' type='application/atom+xml:type=entry' title='to_SalesOrder'/><content type='application/xml'><mproperties><d:SalesOrderID></d:SalesOrderID><d:SalesOrderItemID>4</d:SalesOrderItemID><d:CurrencyCode>PERU</d:CurrencyCode><d:GrossAmount>4.00</d:GrossAmount><d:NetAmount>4.00</d:NetAmount><d:TaxAmount>4.00</d:TaxAmount></mproperties></content></entry>	

HTTP Response - Processing Time = 366 ms

Response in Browser Error Log HTTP Header Use as Request

Header Name	Value
-status_code	500
-status_reason	Internal Server Error

```
<?xml version='1.0' encoding='UTF-8'?>  
- <error  
xmlns='http://schemas.microsoft.com/ado/2007/08/dataservices/metad  
  <code>MESSAGE_TYPE_X</code>  
  <message> Laufzeitfehler: ' MESSAGE_TYPE_X '. Die Bearbeitung  
    des OData-Requests wurde vorzeitig beendet. Ist  
    "Laufzeitfehler" nicht initial, starten Sie die Transaktion ST22,  
    um Details und eine Analyse anzuzeigen. Starten Sie  
    anderenfalls die Transaktion SM21, um eine  
    Systemprotokollanalyse durchzuführen. </message>  
  <timestamptempstamp : 20190306165737 </timestamp>  
</error>
```

when I register the cabezera I have no problem.

The CDS is the following

```
@AbapCatalog.sqlViewName: 'ZDDL_I_S016'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'ZDEMO_I_SALESORDER'
@Search.searchable: true

@ObjectModel:{
  compositionRoot: true,
  transactionalProcessingEnabled: true,
  writeActivePersistence: 'ZPROTO_SO_A',
  semanticKey: ['SalesOrderID'],
  representativeKey: 'SalesOrderID',
  createEnabled: true,
  updateEnabled: true,
  deleteEnabled: true
}

define view ZDEMO_I_SALESORDER
as select from zproto_so_a as SalesOrder
association [0..*] to ZDEMO_I_SALESORDER_ITEM as _Item on $projection.SalesOrder

{

  @Search.defaultSearchElement: true

  key SalesOrder.salesorderid      as SalesOrderID,

  @Search.defaultSearchElement: true

  SalesOrder.businesspartnerid as BusinessPartnerID,

  SalesOrder.currencycode       as CurrencyCode,

  SalesOrder.grossamount        as GrossAmount,
```

```
SalesOrder.netamount      as NetAmount,

SalesOrder.taxamount      as TaxaMount,

SalesOrder.lifecyclestatus as LifecycleStutus,

@ObjectModel.association.type: #TO_COMPOSITION_CHILD

_Item

}
```

Like 0 | Share

Kai Sicker

June 28, 2020 at 10:53 pm

Thanks a lot!

Like 0 | Share

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support