

Wednesday, July 3, 2024 10:22 AM



docshare.ti
ps_sap-fio...

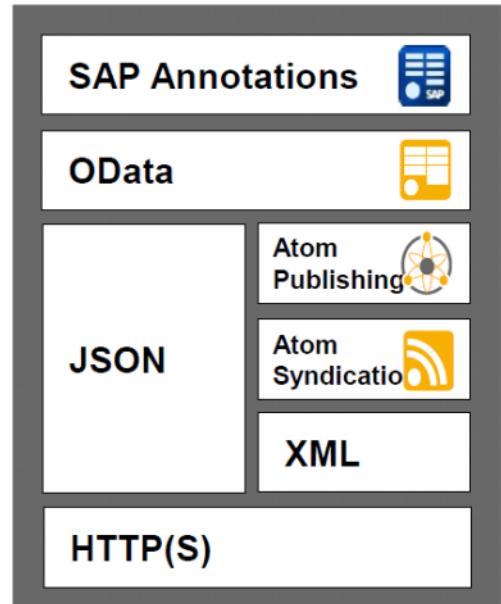
What is the Open Data Protocol (OData)?

OData is an open standard originally developed by Microsoft, but now managed by the [Oasis Organisation](#).

It is based on the Atom Publishing and Atom Syndication standards, which in turn, are based on XML and HTTP(S).

It was designed to provide a standardised implementation of a RESTful API. In doing so, it offers database-like access to server-side resources. Hence, OData has been described as:
“ODBC for the Web”

OData is also extensible. This allows SAP to supplement the data types used by OData with extra information from the ABAP Data Dictionary.



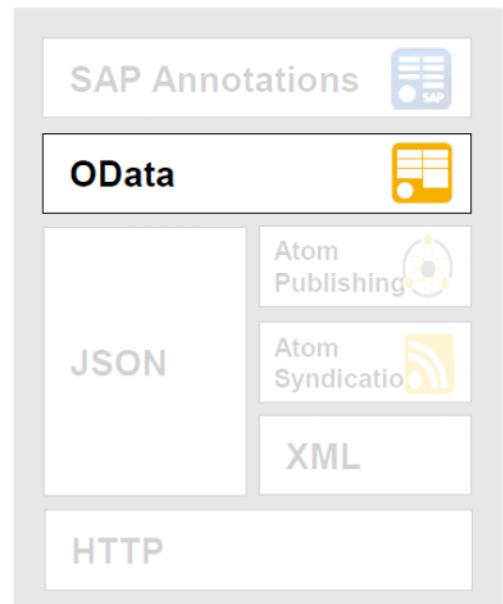
What Does OData Add to Atom?

The Atom Publishing Format does not specify **how** data should be encoded within a Feed; therefore, it fails to be fully RESTful because its messages are not self-describing.

OData extends Atom by providing a metadata description of the message.

OData provides definitions for:

- Simple Types
- Complex Types
- Associations between entries
- Navigation Paths between entries
- Custom behaviour (known as function imports) beyond the standard QUERY, CREATE, READ, UPDATE, DELETE (QCRUD) operations



Output Formats Supported by OData

```
{  
  d: {  
    results: [  
      {  
        __metadata: {  
          uri: "http://usphlrig21.phl.sap.corp:8000/sap/op/odata/v1/Workflows/WorkflowTasks(4500017386)",  
          type: "GBAPP_POAPPROVAL.WorkflowTask"  
        },  
        SAP__Origin: "IDM_800_RIG",  
        WorkitemID: "000001105838",  
        TaskType: "TS20000166",  
        PoNumber: "4500017386",  
        PoNumberFormatted: "4500017386",  
        WiCreatedAt: "/Date(1370422390000)/",  
        ForwardedByID: "",  
        ForwardedByName: "",  
        SubstitutingForID: "",  
        SubstitutingForName: "",  
        CreatedByID: "I003866",  
        CreatedByName: "I003866",  
        Value: "1023.00",  
        Currency: "EUR",  
        SupplierID: "1000",  
        SupplierName: "C.E.B. BERLIN",  
        ItemDescriptions: "Actuation",  
        SearchForText: "",  
        HeaderDetails: {  
          __deferred: {  
            href: "http://usphlrig21.phl.sap.corp:8000/sap/op/odata/v1/Workflows/HeaderDetails(HeaderDetails(1))",  
            type: "application/json"  
          }  
        }  
      }  
    ]  
  }  
}
```

JSON

```
<feed xmlns="http://www.w3.org/2005/Atom" XML  
xml:base="http://usphlrig21.phl.sap.corp:8000/sap/op/odata/v1/Workflows/WorkflowTasks(4500017386)"  
<id>http://usphlrig21.phl.sap.corp:8000/sap/op/odata/v1/Workflows/WorkflowTasks(4500017386)</id>  
<title type="text">WorkflowTaskCollection</title>  
<updated>2013-06-19T07:03:52Z</updated>  
<author>  
<name/>  
</author>  
<link href="WorkflowTaskCollection" rel="self" type="application/atom+xml"></link>  
<entry>  
<id>http://usphlrig21.phl.sap.corp:8000/sap/op/odata/v1/Workflows/WorkflowTasks(4500017386)</id>  
<title type="text">WorkflowTaskCollection(SAP__Origin:IDM_800_RIG, WorkitemID:000001105838, TaskType:TS20000166, PoNumber:4500017386, PoNumberFormatted:4500017386, WiCreatedAt:/Date(1370422390000)/, ForwardedByID:, ForwardedByName:, SubstitutingForID:, SubstitutingForName:, CreatedByID:I003866, CreatedByName:I003866, Value:1023.00, Currency:EUR, SupplierID:1000, SupplierName:C.E.B. BERLIN, ItemDescriptions:Actuation, SearchForText:, HeaderDetails:HeaderDetails(HeaderDetails(1)))</title>  
<category term="GBAPP_POAPPROVAL.WorkflowTask" type="text"/>  
<link href="WorkflowTaskCollection(SAP__Origin:IDM_800_RIG&WorkitemID:000001105838&TaskType:TS20000166&PoNumber:4500017386&PoNumberFormatted:4500017386&WiCreatedAt:/Date(1370422390000)/&ForwardedByID:&ForwardedByName:&SubstitutingForID:&SubstitutingForName:)" type="application/atom+xml"/>  
<content type="application/xml">  
<m:properties>  
<d:SAP__Origin>IDM_800_RIG</d:SAP__Origin>  
<d:WorkitemID>000001105838</d:WorkitemID>  
<d:TaskType>TS20000166</d:TaskType>  
<d:PoNumber>4500017386</d:PoNumber>  
<d:PoNumberFormatted>4500017386</d:PoNumberFormatted>  
<d:WiCreatedAt>2013-06-05T08:53:10Z</d:WiCreatedAt>  
<d:ForwardedByID></d:ForwardedByID>  
<d:ForwardedByName></d:ForwardedByName>  
<d:SubstitutingForID></d:SubstitutingForID>  
<d:SubstitutingForName></d:SubstitutingForName>
```

```
SearchForText: "",  
HeaderDetails: {  
    _deferred: {  
        uri: "http://usphlrig21.phl.sap.corp:"
```

```
<d:ForwardedByID/>  
<d:ForwardedByName/>  
<d:SubstitutingForID/>  
<d:SubstitutingForName/>  
<d:CreatedByID>T003866</d:CreatedByID>
```



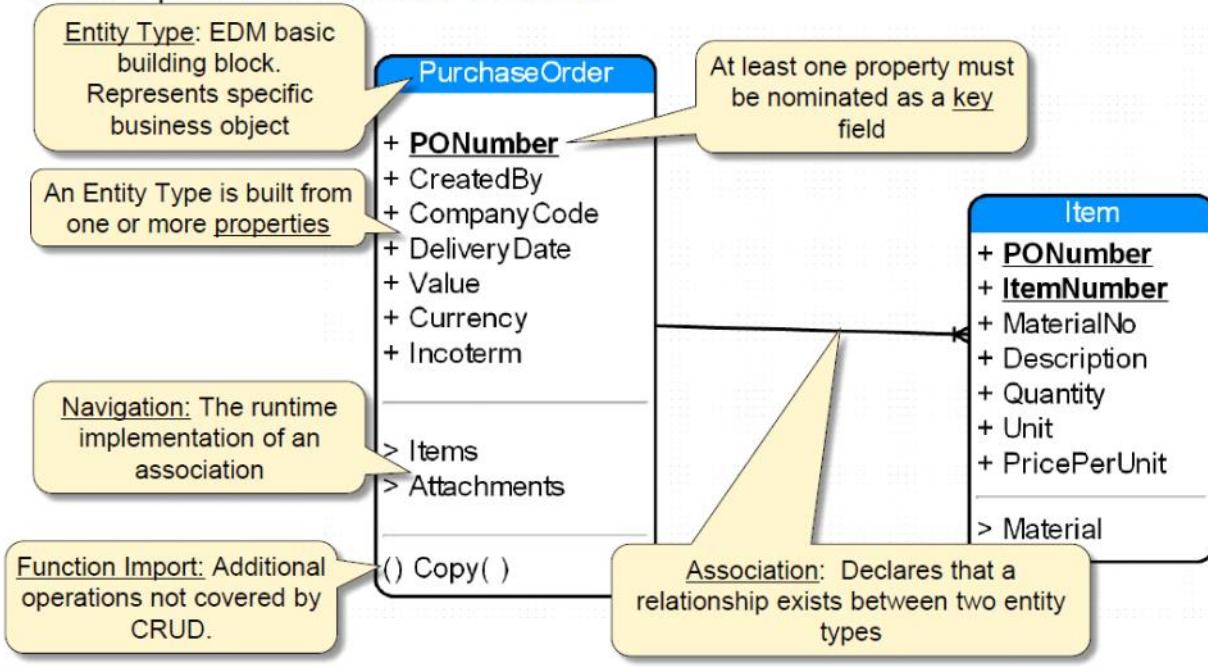
SAP NetWeaver Gateway

Exposing Entity Data Models Using the OData Protocol



Entity Data Model – Overview

An **Entity Data Model** (EDM) is the starting point when designing an OData service. The EDM describes the organisation and relationship of the resources within a particular business scenario.



OData Service Documents – The Consumption Starting Point

After an Entity Data Model has been turned into an OData Service, the Gateway system will provide you with a URL to access this service.

When this URL is requested, the Gateway service will respond by sending you the Service Document.

The Service Document is a high-level description of the resources exposed by the OData service.

```
<app:service xmlns:app="http://www.w3.org/2007/app"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData" xml:lang="en"
  xml:base="http://usphlrig21.phl.sap.corp:8000/sap/opu/odata/sap/gbapp_poapproval/">
  <app:workspace>
    <atom:title type="text">Data</atom:title>
    <app:collection sap:creatable="false" sap:updatable="false"
      sap:deletable="false" sap:addressable="false" sap:content-version="1"
      href="ItemDetailCollection">
      <atom:title type="text">ItemDetailCollection</atom:title>
      <sap:member-title>ItemDetail</sap:member-title>
    </app:collection>
    <app:collection sap:requires-filter="true" sap:content-version="1"
      href="UserDetailsCollection">
      <atom:title type="text">UserDetailsCollection</atom:title>
```

Obtaining OData Metadata

In order to consume an OData service, you should retrieve the metadata that describe the service. This is done by adding the suffix **\$metadata** to the Service Document URL. (OData metadata is only available in XML format, not JSON)

E.G. [http://usphlrig21.phl.sap.corp:8000/sap/\[...\]/\\$metadata](http://usphlrig21.phl.sap.corp:8000/sap/[...]/$metadata)

This will now return an **Entity Data Model** (edmx) XML description of the service:

```
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" xmlns:m="http:  
xmlns:sap="http://www.sap.com/Protocols/SAPData" Version="1.0">  
- <edmx:DataServices m:DataServiceVersion="2.0">  
- <Schema xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="gbapp_p  
+ <EntityType Name="Subscription" sap:semantics="subscriptions" sap:content-ve:  
+ <EntityType Name="Notification" sap:semantics="notifications" sap:content-ve:  
+ <EntityType Name="WorkflowTask" sap:content-version="1">...</EntityType>  
+ <EntityType Name="HeaderDetail" sap:content-version="1">...</EntityType>  
+ <EntityType Name="ItemDetail" sap:content-version="1">...</EntityType>  
+ <EntityType Name="UserDetails" m:HasStream="true" sap:content-version="1">..  
+ <EntityType Name="SupplierDetail" m:HasStream="true" sap:content-version="1"  
+ <EntityType Name="ServiceLine" sap:content-version="1">...</EntityType>  
+ <EntityType Name="Limit" sap:content-version="1">...</EntityType>  
+ <EntityType Name="PricingCondition" sap:content-version="1">...</EntityType>  
+ <EntityType Name="ForwardingAgent" m:HasStream="true" sap:content-version="1"  
+ <EntityType Name="Attachment" m:HasStream="true" sap:content-version="1">...</EntityType>
```



SAP NetWeaver Gateway

OData In Practice – Retrieving Data from an OData Service



Retrieving Data from an OData service

To start with, we will cover the simplest two operations that can be performed using OData: ReadEntitySet and ReadEntity (also referred to as QUERY and READ).

ReadEntitySet (QUERY) : Returns 0..n entries from a collection

ReadEntity (READ) : Returns 0..1 entries from a collection

Both operations are read-only and therefore use the HTTP GET method.

REST Operation	HTTP Method
Create a resource	POST
Retrieve one or more resources	GET
Update a resource	PUT
Delete a resource	DELETE

Retrieving a Collection – Constructing the URL

The URL can be constructed with the help of the Service Document:

```
<app:service
  xmlns:app="http://www.w3.org/2007/app"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData"
  xml:base="http://usphlrig21.phl.sap.corp:8000/sap/opu/odata/sap/gbapp_poapproval/"
  xml:lang="en">
  <app:workspace>
    <atom:title type="text">Data</atom:title>
    <app:collection
      sap:creatable="false" sap:updatable="false" sap:deletable="false,,"
      sap:addressable="false" sap:content-version="1" href='WorkflowTaskCollection'>
      <atom:title type="text">WorkflowTaskCollection</atom:title>
      <sap:member-title>WorkflowTask</sap:member-title>
    </app:collection>
  </app:workspace>
  <atom:link rel="self" href="[...]" />
  <atom:link rel="latest-version" href=" [...]" />
</app:service>
```

The diagram illustrates the mapping of XML elements from the Service Document to the final URL. It shows three main arrows originating from specific XML nodes and pointing to their equivalents in the URL:

- An arrow points from the `xml:base` attribute of the root element to the base URL in the resulting URL.
- An arrow points from the `href` attribute of the `app:collection` element to the collection name in the resulting URL.
- An arrow points from the `atom:link` element with `rel="self"` to the self-link in the resulting URL.

The resulting URL is highlighted in a blue box at the bottom: `http://usphlrig21.phl.sap.corp:8000/sap/[...]/WorkflowTaskCollection`.

Retrieving a Collection – Understanding the Results

Issuing the previously constructed URL from the browser's address line will cause the Gateway server to return an OData XML message containing the requested collection.

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://schemas.microsoft.com/ado/2007/08/metadata">
  ...
  <link href="WorkflowTaskCollection" rel="self" title="WorkflowTaskCollection"/>
  <entry>
    <link href="WorkflowTaskCollection('000001106057')" rel="self" title="WorkflowTaskCollection('000001106057')"/>
    <link href="WorkflowTaskCollection('000001106057')/HeaderDetails" rel="related" title="HeaderDetails"/>
    <content type="application/xml">
      <m:properties>
        <d:WorkitemID>000001106057</d:WorkitemID>
        <d:TaskType>TS20000166</d:TaskType>
        ...
      </m:properties>
    </content>
  </entry>
  <entry>
    ...
  </entry>
</feed>
```

Retrieving a Collection – Filtering

You can filter the `<feed>` contents by using the `$filter` query string parameter.

The `$filter` parameter can contain a complex filter structure consisting of logical, arithmetic, and grouping operators. Also available are string, date, math, and type-related functions.

`/WorkflowTaskCollection?$filter=Value gt 2000 and Currency eq 'USD'`
would only return Workflow Tasks whose PO's total value exceeded \$2000 USD.

IMPORTANT!

1. You must implement the ABAP coding that responds to this parameter
2. A space character (or %20) **must** be included on either side of the operator in the `$filter` condition.
3. Enclose non-numeric values within single quotes.

Retrieving a Collection – Paging

If a collection has a high number of entries, then you should use the `$top` and `$skip` parameters together to implement paging.

`$top` specifies the maximum number of entries that should be returned;
`$skip` specifies how many entries should be ignored when selecting the results

`WorkflowTaskCollection?$top=5` would reduce the `<feed>` down to the first 5 entries.

`WorkflowTaskCollection?$skip=5` would omit the first 5 entries from the `<feed>`.

Paging is achieved by setting `$top` to the number of entries per page, and then incrementing `$skip` by this number every time the next page is required.

Retrieving a Collection – Expanding

When requesting data for an entity type that is associated with other entity types, the Gateway server will provide you with a navigation URL from which you can retrieve the associated data.

If you are certain that you will also need data coming from associated entity types, you can request associated data in advance by using the `$expand` parameter. The Gateway server will then include the data from the specified associations in the feed entries.

`/WorkflowTaskCollection?$expand=HeaderDetails` will return a feed of purchase orders, extended with each Purchase Order's "header details"

`/WorkflowTaskCollection?$expand=HeaderDetails,HeaderDetails/ItemDetails` will return a feed of purchase orders with header details, in turn extended with all the items on every PO.

Retrieving a Single Entry

OData employs a special resource path syntax to reference a single entity from a collection. Essentially, you specify the collection name followed by all keys and their respective values that identify the entity in question. Multiple keys are separated with a comma.

`/WorkflowTaskCollection(WorkitemID='000001105834')` will return a single `<entry>` from the workflow task collection with the specified ID.

You will also need this special resource path syntax when navigating between associated entities. However, the appropriate syntax is already applied in an `<entry>`'s `<link>` elements:

```
<entry>
  <link href="WorkflowTaskCollection('000001106057')" rel="self" title="WorkflowTaskCollection('000001106057')"
  <link href="WorkflowTaskCollection('000001106057')/HeaderDetails" title="HeaderDetails"
  <content type="application/xml">
    ...
  </content>
</entry>
```

Retrieving a Single Value

Based on an entity read, you can limit your request so it only returns a single property. The OData service will then return the property's value in plain text.

This can be achieved using the `$value` navigation:

Based on the resource path of a single entry, append the property's name as well as `$value` as if they were navigation paths.

`/WorkflowTaskCollection('000001106057')/SupplierName/$value` would only return the Purchase Order's supplier name, not the entire Purchase Order.

In order to limit the resulting entries to several properties, you may use the `$select` parameter. `$select` should contain a comma-separated list of property names that are to be retrieved.

`/WorkflowTaskCollection?$select=Value,Currency` would only list total value and currency for every Purchase Order.

[Show TOC](#)

Front-End Server: Activate OData Services

There are several ways to activate the relevant OData services:

- You can activate the OData services for each app individually, as given below.
- You can activate OData services for several apps at once, by using a task list. For more information, see [Activate OData Services for Several SAP Fiori Apps](#).
- You can activate OData services and ICF nodes at once, by using an SAP Fiori app. For more information, see [Activate Apps with SAP Fiori App Implementation Foundation](#).

Procedure

1. Run transaction Activate and maintain services (/IWFND/MAINT_SERVICE) on the front-end server.
2. Choose Add Service.
3. Enter the system alias of your back-end system.
4. In the External Service Name field, enter the technical name of the OData service for your app without the version number. For more information on the OData service per app, see the app-specific documentation in the section *SAP Fiori Apps*.
5. In the Version field, enter the version number.
6. Choose Get Services.
7. Choose Add Selected Services.
8. Enter a technical name for the service in your customer namespace.
9. Assign a package or choose Local Object.
10. Choose Execute to save the service.
11. On the Activate and maintain services screen, check if the system alias is maintained correctly. If not, delete the alias and add the correct one.

Inserted from <[O DATA services Page 14](https://help.sap.com/fiori_bs2013/helpdata/en/55/b840530a3ee447e1000000a441470/content.htm?frameset=/en/11/e168533271a548e1000000a423f68/frameset.htm¤t_toc=/en/11/e168533271a548e1000000a423f68/plain.htm&node_id=10></p></div><div data-bbox=)

SAP (Fiori) OData Service Examples

Friday, April 13, 2018 10:18 PM

SAP (Fiori) OData Service Examples

April 10, 2018 | 437 Views |



[Dilip Kumar KrishnaDeo Pandey](#)

Overview

- Main objective of this blog is to put different oData Service blogs (*of mine*) at one place with respect to their business case summary.
- And to put brief info of different terms used in oData Service creation/consumption.
- *This blogs will be updated time to time as soon as new business case blog is ready..*

OData Service Examples w.r.t. Business cases:

1. Odata Service to get one table records via RFC

- A business case where we want to fetch material master table records. Following blogs can be referred for same:
 - To create similar OData Service:
 - [Create OData Service in SAP Fiori Server](#)
 - To register similar OData Service:
 - [Register OData Service in SAP Fiori Server](#)
 - To consume similar oData Service in Fiori App:
 - [SAP Fiori – Consume OData Service, CSS, i18n properties in UI5 Application](#)
 - *Please Note:*
 - OData service Registration happens once for one service. Registration Steps are common.

2. Odata Service to post multiple table as a request in a single call

- A business case where, in a single service call, we want to send multiple tabular requests and in response we want to get results in separate table structure. Following blogs can be referred for same:
 - To create similar OData Service:
 - [SAP oData Service \[POST\]: Multiple Table Input](#)
 - To consume similar oData Service in Fiori-App
 - [Call oData Service in Fiori App \(Eclipse\): 'Create' operation](#)

3. Odata Service to get multiple table response in a single call

- A business case where, in a single service call, we want to get multiple tabular results, which data may be used for drop-downlist.
 - To create similar OData Service:
 - [SAP oData Service \[GET\]: Multiple Table Output](#)
 - To consume similar oData Service in Fiori-App
 - [Call oData Service in Fiori App \(Eclipse\): 'Read' operation](#)

There are many different terms/objects been used during creation/implementation of above oData Service examples, which are been summarized as follows: (*Please note: Below definitions are been extracted from different links on internet and been related to my business-case blogs for better & brief understanding.*)

OData

- The Open Data Protocol (OData) is a data access protocol built on core protocols like HTTP and commonly accepted methodologies like REST for the web.
- OData interface is an open standard that can be consumed by any application, program, software or device of the Non-SAP World that can connect with SAP using the HTTP(s) protocol and can manipulate (read, write, modify and understand i.e. parse and construct) an XML document.
- OData is REST-inspired technology for reading, writing, and modifying information on the Web (not just SAP).
- Advantages of OData
 - Since the protocol is HTTP based, any programming language with HTTP stack can be used for consuming OData services
 - The OData interface is implemented using XML or JSON. Both of these formats are well known, plain text protocols for the transmission of information over the Web.
 - Data message is self-describing. So any non-SAP Web developer can understand the content of the OData message.
 - With the advent of OData, the communication barrier between SAP and Non-SAP Web Developers is removed. It is an Open book now. There is no cost or license agreement needed for the use of OData.

HTTP:

- HTTP (Hyper Text Transfer Protocol) is based on Client-Server Architecture.
- The Browser is the Client which sends HTTP request and Web Server is the Server which sends the response back to the browser.
- HTTP defines
 - “WHAT” can be transferred between Client and Server.
 - “HOW” the data packets are transferred via HTTP is handled by TCP/IP protocols.

Stateless

- Every single HTTP request that is received by the Web Server is forgotten after a response has been sent across. Web Servers do not remember or recall the previous request. This is called stateless.
- Odata services have this nature.

REST (REpresenational State Transfer)

- REST is an architectural style that uses simple and lightweight mechanism for inter-machine communication.
- It is an alternative to the RPC (Remote Procedure Calls) and Web Services.
- REST is ‘Resource-based’, unlike RPC or SOAP which are Action-based.
- REST Services are working with ‘Resources’ instead of ‘Operations’.

URI (Unified Resource Identifier)

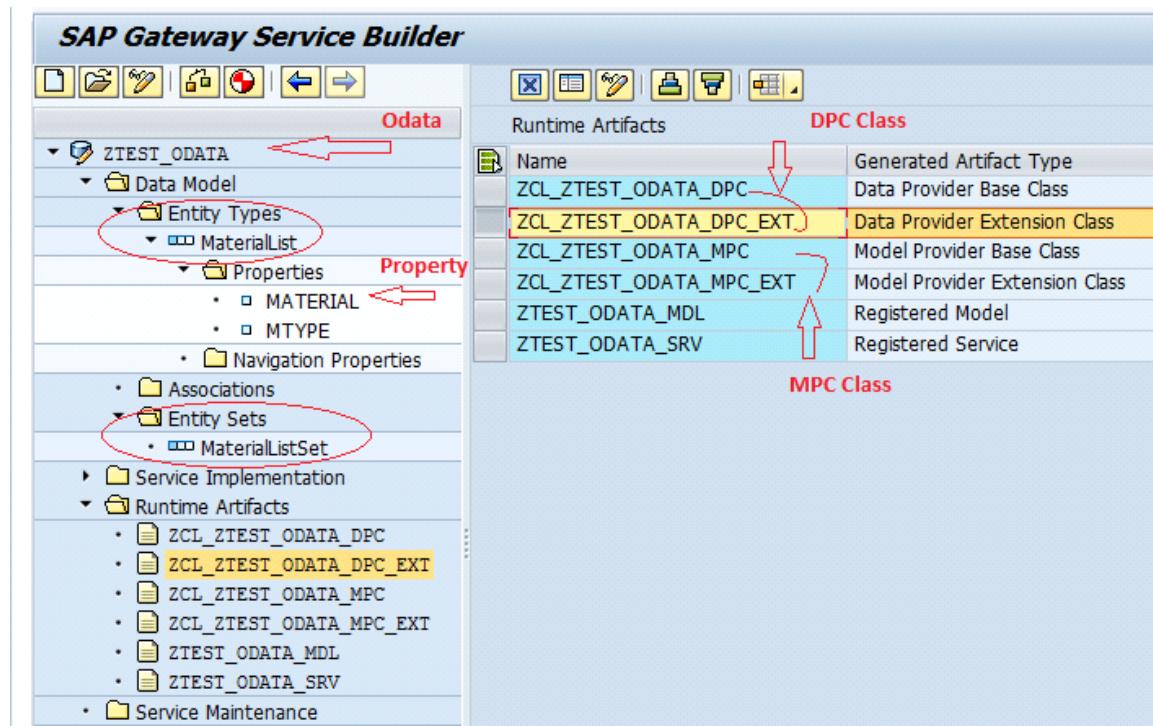
- In REST services, any communication between client and services are using URI over HTTP protocol using HTTP method.
- The URI is really the representation of the Resources (EntitySets like MaterialList, DocType,

VendorList etc).

- In RESTful service, once Resource been identified, we work with a uniform interface because it uses HTTP methods to work with the resource.
- HTTP methods are as below:
 - GET is to get the representation of an existing resource.
 - POST is to add new resource into the system.
 - PUT is to modify the existing resource
 - DELETE is to remove the resource from the system.
- So, the client does not need to know what the exact operation name defined in the service contract to call that method.
- For example:
 - consider above business case "[1. Odata Service to get one table records via RFC](#)" where odata service's
 - Request URI is "/sap/opu/odata/sap/ZTEST_ODATA_SRV/MaterialListSet"
 - Resource is "MaterialListSet"
 - Operation is "GET"
 - Protocol is "HTTP"

Creations of OData Service:

We create oData service in SAP-Fiori (front-end) system using t-coed 'SEGW'. Below screen displays different components of oData



EntityType & EntitySet

- Entity-Type is a structure (or a work area which holds just one row)
- Entity-Set is an internal table (holds more than one entity/rows).

Property:

- This is to define the fields of the structure/work area and internal table Service Implementation

Service Implementation

- Service Implementation folder has MaterialListSet Operations auto generated.
- These are ABAP Methods which would be triggered when the relevant endpoints would be called.
- Auto generated operations are:
 1. Create
 2. Delete
 3. GetEntity (Read)
 4. GetEntitySet (Query)
 5. Update

MPC & DPC

- Two classes, MPC and DPC are also generated along with Base and Extended Class.
- DPC and MPC are not connected by any coding. They talk to each other via Configuration.
- MPC (Model Provider class)
 - This defines the Gateway Service interface.
 - This is used to define model.
 - We can use the method Define to create entity, properties etc using code based implementation.
 - We rarely use MPC extension class.
- DPC (Data provider class)
 - This provides the Gateway Service functionalities.
 - It is used to code our 'CRUDQ' methods as well as function import methods.
 - We write all our logic in redefined methods of DPC extension class.
- The 'CRUDQ' methods:
 - This is nothing but Create, Read, Update, Delete and Query operations which we can do in oData Service.

Association & Navigation:

- These are two important properties available in SAP Netweaver Gateway to associate two/multiple entity types.
- Example:
 - If we want to push Header and Items data at a time in one call and in acknowledgement we want output in Result entity. This is can be achieved using association and navigation property
 - Above business case can be referred for same:
 - [“1. Odata Service to get one table records via RFC”](#)

Query Operation (GET_ENTITYSET):

- If we want to get all records of a table via oData Service, then we re-define method 'GET_ENTITYSET' of DPC.
- It returns an array or internal table in defined Entity structure format.
- Example:
 - Suppose our Fiori app wants to display all records from MARA table in some DropDownList (F4 help box).
 - Above business case examples can be referred i.e.

- “[2. Odata Service to post multiple table as a request in a single call](#)“
- “[3. Odata Service to get multiple table response in a single call](#)“

Read Operation (GET_ENTITY):

- If we want to fetch only one record based on input, then we redefine method ‘GET_ENTITY’ of DPC.
- Example:
 - Suppose our Fiori application wants to connect to SAP using OData service by providing the Material number and pull only that Material numbers details from MARA table.

The \$expand Query (GET_EXPANDED_ENTITYSET):

- The \$expand query option is very powerful and allows us to provide multiple entities and/or entity sets in one single service call, instead of performing several calls subsequently.
- For \$expand query to work, an association or navigation property should be created.
- And we implement ‘GET_EXPANDED_ENTITYSET’ method.
- Example:
 - Above business case example can be referred for same i.e. “[3. Odata Service to get multiple table response in a single call](#)“

Deep Insert (Create_Deep_Entity):

- To post/push Header and line items together to the back-end RFC via oData Service, we follow Create_Deep_Entity approach.
- The ‘Create_Deep_Entity’ approach is also called as Deep Insert in SAP OData service
- Deep insert is used to POST the nested structure of feed/collections to the back-end system.
- By implementing this we can reduce the no.of OData calls made to the SAP Netweaver Gateway server.
- Entity sets (Header, Item and Result) which are used should be associated, means while calling one Entity set (for e.g. say Header), we should use Item and Result Entity Sets as well, thus in a single call we are using three Entity Sets thus avoiding no.of OData calls.
- This can be achieved by the concept of Association and Navigation properties.
- Example:
 - Above business case example can be referred for same i.e. “[2. Odata Service to post multiple table as a request in a single call](#)“

Inserted from <<https://blogs.sap.com/2018/04/10/sap-fiori-odata-service-examples/>>

How to Guide: SAP Fiori Extensibility-Adding Custom Fields in OData

Thursday, February 22, 2018

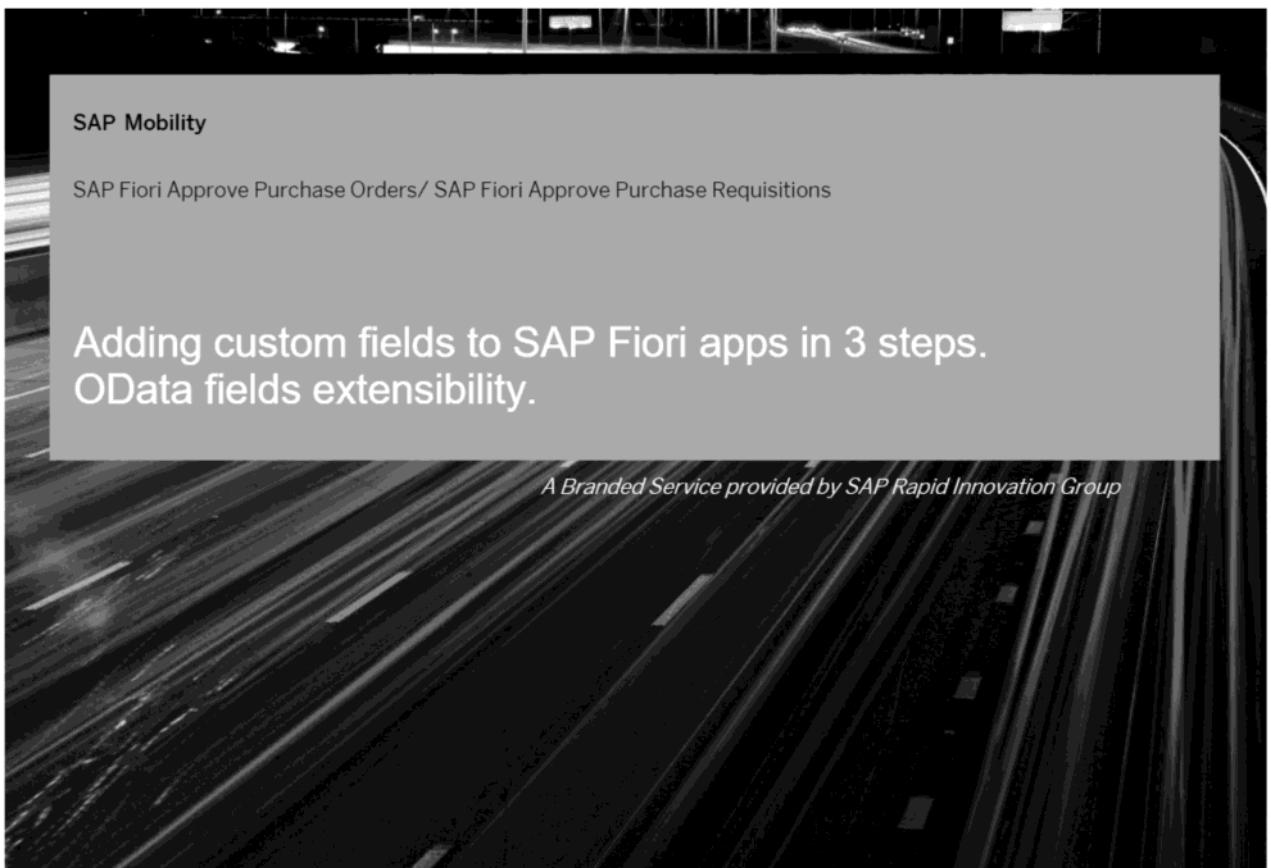
How to Guide: SAP Fiori Extensibility-Adding Custom Fields in OData

Adding custom fields in SAP Fiori apps UI is a common requirement from customers. Customers have custom fields in ERP backend applications. You are a consultant to enable those customer fields in SAP Fiori apps. This how to guide give you detail steps for adding custom fields in OData. [Download the Document](#)

Inserted from <<https://www.sap.com/documents/2015/08/9ccbe927-5a7c-0010-82c7-eda71af511fa.html>>



How to
Guide_S...



Applicable Releases:

SAP Fiori Approve Purchase Orders 1.0
SAP Fiori Approve Purchase Requisitions 1.0
SAP NetWeaver Gateway 2.0 SP 06

Version 1

June 2013



© Copyright 2013 SAP AG. All rights reserved.

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 Software Ltd. Business Objects is an SAP company.

SAP "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.



Document History

Document Version	Description
------------------	-------------

1	The first release version



Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip



The Best-Run Businesses Run SAP™

Table of Contents

1.	Business scenario	1
1.1	Sample scenario	1
1.2	Prerequisites	1
2.	Architecture	2
2.1	Overview	2
2.2	Test OData services [Gateway]	3
3.	Implementation	6
3.1	Step 1 . Extend the OData Entity in the GW Model Provider by implementing BAdI [ERP]	6
3.2	Step 2 . Extend the corresponding ABAP DDIC Structure by Append Structure [ERP]	6
3.3	Step 3. Fill data to the custom fields in the GW Data Provider by implementing BADI [ERP]	12
4.	Test the OData service	17
4.1	Check the OData service - Before	17
4.2	Clear the Cache	17
4.3	Test the OData service - after	17
4.4	UI extensibility	18



1. Business scenario

Adding custom fields in SAP Fiori apps UI is common requirement from customers. Customers have custom fields in ERP backend applications. You are a consultant to enable those customer fields in SAP Fiori apps.

There are 2 how to guides for adding custom fields.

- Part 1: Adding custom fields to SAP Fiori apps in 3 steps. OData fields extensibility.
- Part 2: Adding custom fields to SAP Fiori apps. UI fields extensibility.

1.1 Sample scenario

In this How to Guide we will show you how to add the "Purchasing Group" field in SAP Fiori Approve Purchase Orders.

The screenshot shows the SAP Fiori Approve Purchase Orders application. On the left, there is a list of purchase orders with details like amount and date. On the right, a detailed view of a specific purchase order is shown. The detailed view includes fields for Purchasing Group (ZOO), Purchase Order number (4500017996), Account Assignment (Cost Center 1110 Executive Board), G/L Account (400000 Consumption, raw material 1), Delivery Date (August 17, 2013), Plant (Werk Hamburg), Payment Terms (14 days 3%, 30/2%, 45 net), Incoterms (EXW from plant; Wallendorf), and Company Code (IDES AG (1000)). At the bottom, there is a table for 'Items (1)' with columns for Description, Quantity, and Subtotal. The description column shows 'LAT Test: casing'. At the bottom right, there are 'Approve' and 'Reject' buttons.

Implementation steps are in 3 steps.

1. Extend the corresponding ABAP DDIC Structure by Append Structure
2. Extend the OData Entity in the GW Model Provider by implementing BADI
3. Fill data to the custom fields in the GW Data Provider by implementing BADI

Steps are also same for SAP Fiori Approve Purchase Requisitions.

Note: The standard delivered BADI is different in each app. Please look at the Installation and Configuration Guide in the <http://help.sap.com/fiori> and also look at details in the transaction SE18 and SE80

1.2 Prerequisites

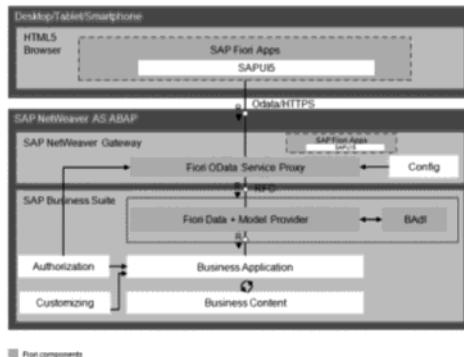
- SAP Fiori Approve Purchase Order is up and running.
- You have ABAP programming knowledge.

2. Architecture

2.1 Overview

The OData components are installed on SAP Business Suite backend.

Architecture – SAP Fiori components

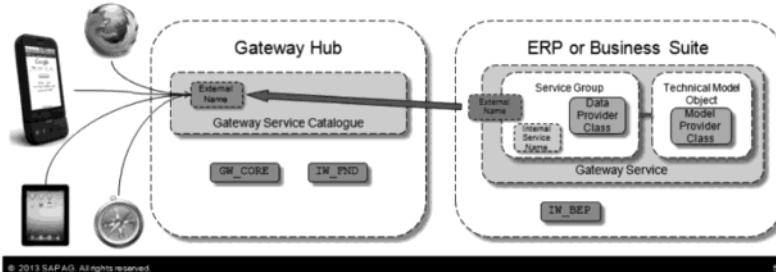


It has Model Provider Class and Data Provider Class and methods have enhancement spots which call BAdI (Business Add In) implementations.

Gateway Service Catalogue

A Gateway Service is always developed in whichever server has component IW_BEP installed. (This will typically be your backend server).

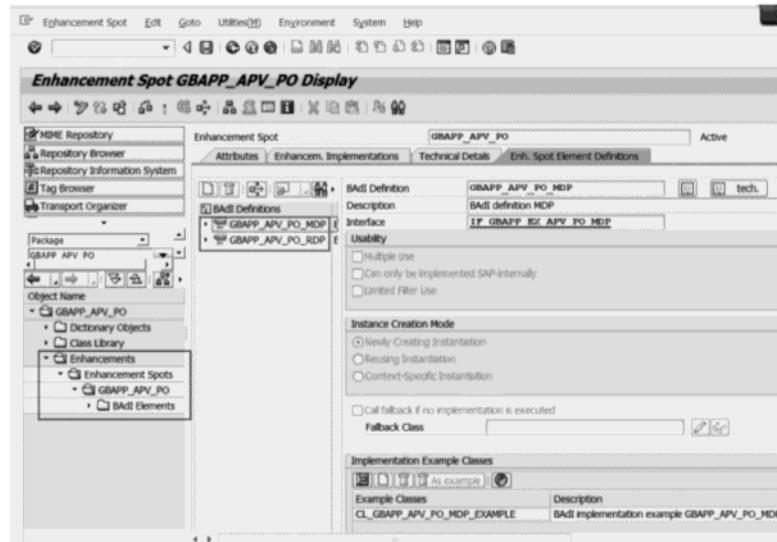
In order to expose our Gateway Service to the outside world, we must create an entry in the Service Catalogue on whichever server is acting as the Gateway Hub.



You can check the enhancement details in SE80 or SE18.

Transaction: SE80

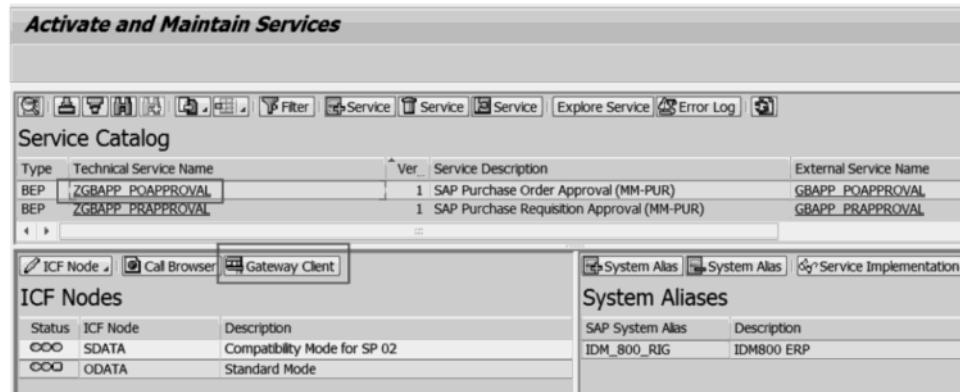
Package: GBAPP_APV_PO



2.2 Test OData services [Gateway]

1. Logon to NetWeaver Gateway system and call transaction /IWFM/MAINT_SERVICE. Logon user should be the approver who has workitems for approving Purchase Orders. Test OData using Gateway Client tool.

Transaction: /IWFM/MAINT_SERVICE



2. Test the \$metadata.

/sap/opu/odata/SAP/GBAPP_POAPPROVAL/\$metadata



SAP NetWeaver Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/SAP/GBAPP_POAPPROVAL/\$metadata

HTTP Response

```

Header Name Value
~status_code 200
~status_reason OK

<?xml version="1.0" encoding="utf-8" ?>
- <edmx:Edmx Version="1.0" xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData">
- <edmx:DataService m:DataServiceVersion="2.0">
- <Schema Namespace="GBAPP_POAPPROVAL" xml:lang="en"
  xmlns="http://schemas.microsoft.com/ado/2008/09/edmx">
+ <EntityType Name="Subscription" sap:semantics="subscriptions" sap:content-version="1">
+ <EntityType Name="Notification" sap:semantics="notifications" sap:content-version="1">
+ <EntityType Name="WorkflowTask" sap:content-version="1">
+ <EntityType Name="HeaderDetail" sap:content-version="1">
+ <Key>
  <Property Name="HeaderInfo" Type="GBAPP_POAPPROVAL.HeaderInfo" Nullable="false" />
<Property Name="WorkitemID" Type="Edm.String" MaxLength="12" sap:label="ID" />
<Property Name="PoNumber" Type="Edm.String" Nullable="false" MaxLength="10" sap:label="Doc." />

```

There is no Property for "Purchasing Group" in Header detail.

3. Test the OData service and check collections.

/sap/opu/odata/SAP/GBAPP_POAPPROVAL?\$format=xml

SAP NetWeaver Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/SAP/GBAPP_POAPPROVAL?\$format=xml

HTTP Response

```

Header Name Value
~status_code 200
~status_reason OK

+ <app:collection sap:createable="false" sap:updatable="false" sap:deletable="false" sap:addressable="false"
  version="1" href="LimitCollection">
+ <app:collection sap:createable="false" sap:updatable="false" sap:deletable="false" sap:addressable="false"
  version="1" href="PricingConditionCollection">
+ <app:collection sap:content-version="1" href="ForwardingAgentCollection">
+ <app:collection sap:requires-filter="true" sap:content-version="1" href="AttachmentCollection">
+ <app:collection sap:content-version="1" href="NoteCollection">
+ <app:collection sap:content-version="1" href="SupplierContactCollection">
+ <app:collection sap:content-version="1" href="AccountingCollection">
+ <app:collection sap:content-version="1" href="SubscriptionCollection">
+ <app:collection sap:createable="false" sap:updatable="false" sap:deletable="false" sap:addressable="false"
  version="1" href="NotificationCollection">
+ <app:collection sap:createable="false" sap:updatable="false" sap:deletable="false" sap:pageable="false"
  sap:addressable="false" sap:content-version="1" href="WorkflowTaskCollection">
+ <app:collection sap:createable="false" sap:updatable="false" sap:deletable="false" sap:addressable="false"
  version="1" href="HeaderDetailCollection">
</app:workspace>

```

4. Get a workitem ID

/sap/opu/odata/SAP/GBAPP_POAPPROVAL/WorkflowTaskCollection?\$filter=TaskType eq 'TS20000166'&\$top=1

Adding custom fields to SAP Fiori apps in 3 steps. OData fields extensibility.

SAP NetWeaver Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/SAP/GBAPP_POAPPROVAL/WorkflowTaskCollection?\$filter=TaskType eq 'TS20000166'&\$top=1

Header Name Value
~status_code 200
~status_reason OK

```
<id>http://usphirig21.phl.sap.corp:8000/sap/opu/odata/SAP/GBAPP_POAPPROVAL/WorkflowTaskC('000001105834')</id>
<title type='text'>WorkflowTaskCollection('000001105834')</title>
<updated>2013-06-22T03:12:18Z</updated>
<category term="GBAPP_POAPPROVAL.WorkflowTask">
  scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
  <link href="WorkflowTaskCollection('000001105834')" rel="self" title="WorkflowTask" />
  <link href="WorkflowTaskCollection('000001105834')/HeaderDetails" rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/HeaderDetails" type="application/atom+xml;type=entry" title="HeaderDetails" />
<content type="application/xml">
  - <m:properties>
    <d:WorkItemID>000001105834</d:WorkItemID>
    <d:TaskType>TS20000166</d:TaskType>
    <d:PoNumber>4500017385</d:PoNumber>
    <d:PoNumberFormatted>4500017385</d:PoNumberFormatted>
```

5. Get the Purchase Order detail
`/sap/opu/odata/sap/GBAPP_POAPPROVAL/WorkflowTaskCollection('000001105834')/HeaderD
etails?$expand=ItemDetails,Notes,Attachments`

SAP NetWeaver Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/sap/GBAPP_POAPPROVAL/WorkflowTaskCollection('000001105834')/HeaderDetails?\$expand=ItemDetails,Notes,Attachments

Header Name Value
~status_code 200
~status_reason OK

```
<d:CreatedBy>I003866</d:CreatedBy>
<d:CreatedByName>I003866</d:CreatedByName>
<d:NumberOfNotes>0</d:NumberOfNotes>
<d:NumberOfAttachments>0</d:NumberOfAttachments>
<d:Value>1023.00</d:Value>
<d:Currency>EUR</d:Currency>
<d:SupplierID>1000</d:SupplierID>
<d:SupplierName>C.E.B. BERLIN</d:SupplierName>
<d:NumberOfItems>1</d:NumberOfItems>
<d:Plant>1000</d:Plant>
<d:PlantName>Werk Hamburg</d:PlantName>
<d:DeliveryDate>2013-08-31T00:00:00</d:DeliveryDate>
<d:DeliveryDateAlsoLater />
```

No Purchasing Group

At the end of this How to Guide, you see Purchasing Group data in here. It means the OData service provides Purchasing Group value and HTML5 UI can consume it.

3. Implementation

3.1 Step 1 . Extend the corresponding ABAP DDIC Structure by Append Structure [ERP]

In the step 1 , you are going to add the field „Purchasing Group“ in the structure of entity.

1. Open the structure

Transaction: SE80

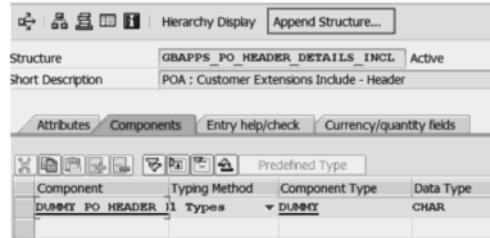
Structure: GBAPPS_PO_HEADER_DETAILS and GBAPPS_PO_WORKFLOW_TASK

The screenshot shows the SAP SE80 Object Navigator. In the left pane, under 'Object Name', there is a tree view of objects. Under 'GBAPP_APV_PO', the 'Structures' node is expanded, showing several entries. One entry, 'GBAPPS_PO_HEADER_DETAILS', is highlighted with a red box. The right pane displays detailed information about this structure, including its description ('GW Service: Purchas...'), components, and attributes.

2. Double click the include GBAPPS_PO_HEADER_DETAILS_INCL.

The screenshot shows the SAP SE80 structure editor for the structure 'GBAPPS_PO_HEADER_DETAILS'. The 'Components' tab is active. A table lists components and their types. The 'INCLUDE' component is highlighted with a red box and is listed as having a type of 'GBAPPS_PO_HEADER_DETAILS_INCL'.

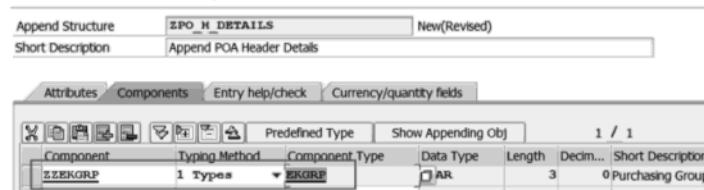
3. Select the „Append Structure...“ button.



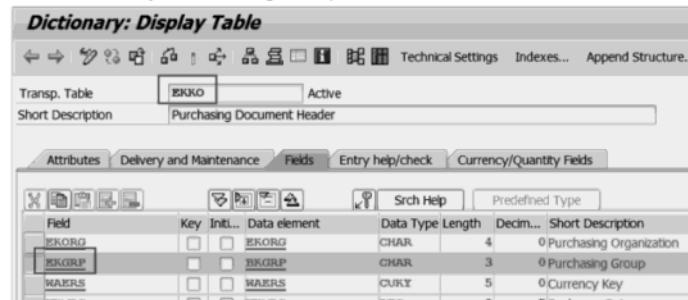
4. Enter the Append name. For example ZPO_H_DETAILS..



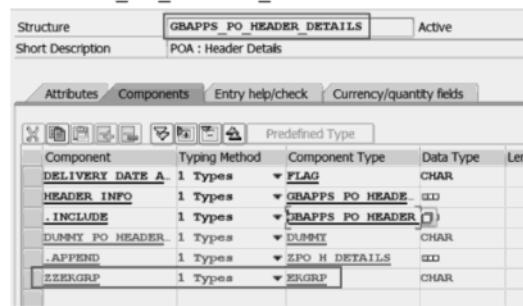
5. Add field. For example ZZEKGRP.



Information only: Purchasing Group field „EKGRP“ is in the table „EKKO“.



6. Activate the Append Structure. You should see the ZZEKGRP in the Structure GBAPPS_PO_HEADER_DETAILS.



7. Repeat same steps for GBAPPS_PO_WORKFLOW_TASK structure. There is include GBAPPS_PO_WORKFLOW_TASK_INCL.

Structure			
POA : WorkFlow Task List			
Active			
Attributes / Components / Entry help/check / Currency/quantity fields			
Component	Typing Method	Component Type	Data Type
SUPPLIER_ID	1 Types	LIFNR	CHAR
SUPPLIER_NAME	1 Types	NAME1_GP	CHAR
AGG_ITEM_DESC	1 Types	STRINGVAL	STRING
SEARCH_FOR_TEXT	1 Types	STRINGVAL	STRING
.INCLUDE	1 Types	GBAPPS_PO_WORKF...	UD
DUMMY_PO_WORKFL	1 Types	DUMMY	CHAR
.APPEND	1 Types	ZPO_WF_TASK	UD
ZZEKGRP	1 Types	EKGRP	CHAR

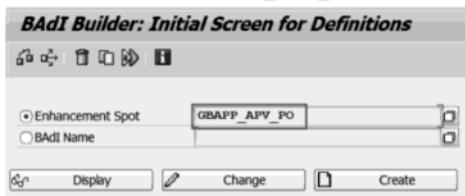
3.2 Step 2 . Extend the OData Entity in the GW Model Provider by implementing BAdI [ERP]

In the step 2, you are going to implement a BAdI method which adds “Purchasing Group” property in the Model Provider class.

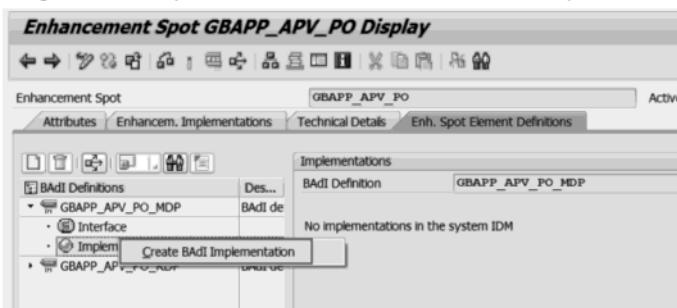
8. Display Enhancement Spot

Transaction: SE18

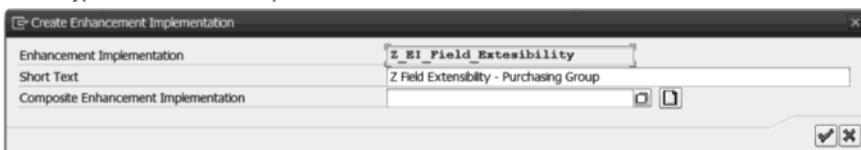
Enhancement Spot: GBAPP_APV_PO



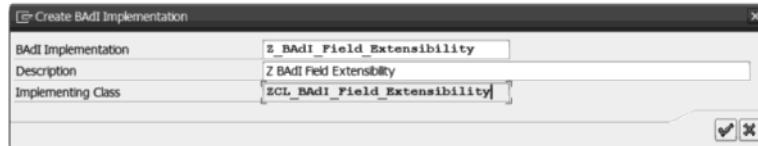
9. Right click on Implementations and select “Create BAdI Implementation”.



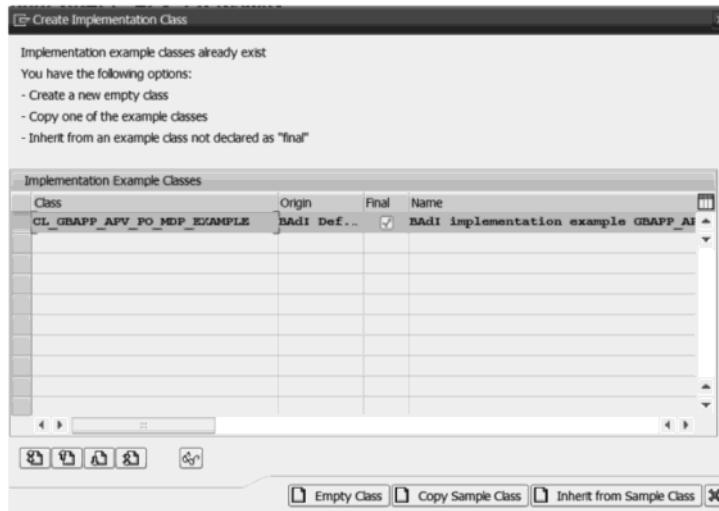
10. Type Enhancement Implementation name



11. Type BAdI Implementation name

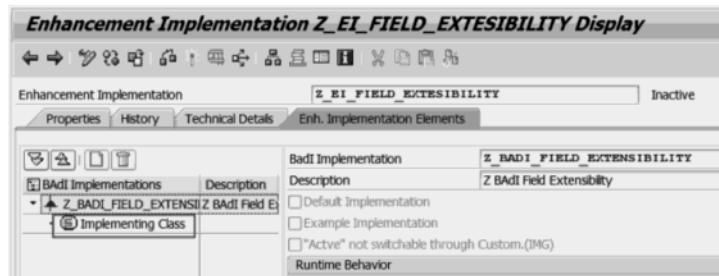


You have option for copying the sample class. In this guide, we copy the sample class.

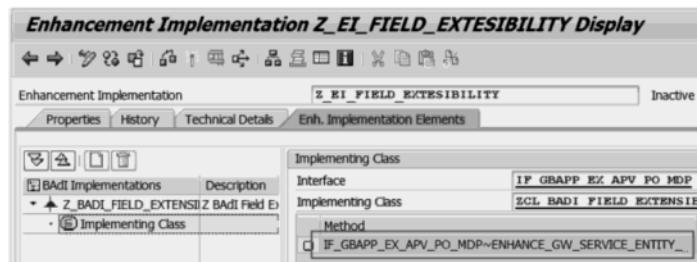


12. Implement the DAdI

Double click the Implementing Class.



Double click the method.



You see the sample code with comments.

Class Builder: Class ZCL_BADI_FIELD_EXTENSIBILITY Display

```

Method IF_GBAPP_EX_APV_PO_MDP-ENHANCE_GW_SERVICE_ENTITY_MDP [Inactive]
32: * DATA: lo_property TYPE REF TO /iwbep/if_mgw_odata_property.
33: *
34: *
35: *
36: * IF iv_entity_type_name = 'WorkflowTask'.
37: *   lo_property = io_entity_type->create_property( iv_property_name = 'CashDiscount1'    iv_abap_fieldname = 'CashDiscount1' ). 
38: *   lo_property->set_nullable( abap_true ). 
39: ** ... etc
40: *
41: *   lo_property = io_entity_type->create_property( iv_property_name = 'ProcCardCompany' iv_abap_fieldname = 'ProcCardCompany' ). 
42: *   lo_property->set_nullable( abap_true ). 
43: ** ... etc
44: *
45: *   lo_property = io_entity_type->create_property( iv_property_name = 'ProcCardNr'      iv_abap_fieldname = 'ProcCardNr' ). 
46: *   lo_property->set_nullable( abap_true ). 
47: ** ... etc
48: *
49: * ELSEIF iv_entity_type_name = 'Header'.
50: ** ... etc
51: *
52: *
53: *
54: **** You need also to extend the related bind structure with the new defined fields.
55: * E.g. for Entity 'WorkflowTask' -> the bind structure 'GBAPPS_PO_WORKFLOW_TASK'
56: 
```

13. Go to change mode and add "ZPurchasingGroup" property in both entity "WorkflowTask" and "HeaderDetail".

The UI screenshot helps you to understand why you handle 2 entity types.



Remember that you have looked at the meta data definition in the section 2.2.



SAP NetWeaver Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/sap/GBAPP_POAPPROVAL/\$metadata?sap-documentation=all

Test Group Test Case

HTTP Response

Header Name	Value
status_code	200
status_reason	OK

```

1 <?xml version="1.0" encoding="utf-8" ?>
- <edmx:Edmx Version="1.0" xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData">
- <edmx:DataServices m:DataServiceVersion="2.0">
- <Schema Namespace="GBAPP_POAPPROVAL" xml:lang="en" xmlns="http://schemas.microsoft.com/ado
  + <EntityType Name="Subscription" sap:semantics="subscriptions" sap:content-version="1">
  + <EntityType Name="Notification" sap:semantics="notifications" sap:content-version="1">
  + <EntityType Name="WorkflowTask" sap:content-version="1">
  □ <EntityType Name="HeaderDetail" sap:content-version="1">
    - <Key>
      <PropertyRef Name="PoNumber" />
    </Key>
    <Property Name="HeaderInfo" Type="GBAPP_POAPPROVAL.HeaderInfo" Nullable="false" />
    <Property Name="WorkitemId" Type="Edm.String" MaxLength="12" sap:label="ID" />
    <Property Name="PoNumber" Type="Edm.String" Nullable="false" MaxLength="10" sap:label="Purchasi

```

Program Code:

```

METHOD if_gbapp_ex_apv_po_mdp~enhance_gw_service_entity_mdp.

DATA: lo_property  TYPE REF TO /iwbep/if_mgw_OData_property.

IF iv_entity_type_name = 'HeaderDetail' OR
   iv_entity_type_name = 'WorkflowTask'.

  lo_property = io_entity_type-
  >create_property( iv_property_name = 'ZPurchasingGroup' iv_abap_fieldname = 'ZZEKGRP' ).
  lo_property->set_nullable( abap_true ).

ENDIF.

ENDMETHOD.

```

Class Builder: Class ZCL_BADI_FIELD_EXTENSIBILITY Change

Method: IF_GBAPP_EX_APV_PO_MDP~ENHANCE_GW_SERVICE_ENTITY_MDP Inactive (revised)

```

1 METHOD if_gbapp_ex_apv_po_mdp~enhance_gw_service_entity_mdp.
2
3   DATA: lo_property  TYPE REF TO /iwbep/if_mgw_o_data_property.
4
5   IF iv_entity_type_name = 'HeaderDetail' OR
6     iv_entity_type_name = 'WorkflowTask'.
7
8     lo_property = io_entity_type->create_property( iv_property_name = 'ZPurchasingGroup' iv_abap_fieldname
9     lo_property->set_nullable( abap_true ).
10
11   ENDIF.
12
13 ENDMETHOD.

```

14. Activate the method.

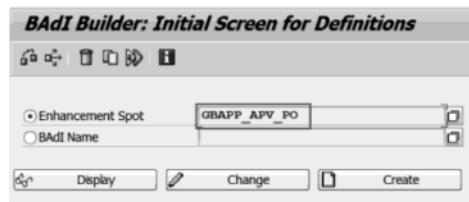
3.3 Step 3. Fill data to the custom fields in the GW Data Provider by implementing BADI [ERP]

In the step 3, you are going to implement a BADI method which fills "Purchasing Group" value in the Data Provider class.

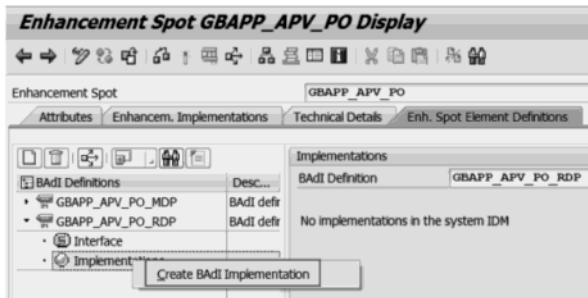
1. Open the Enhancement Spot.

Transaction: SE18

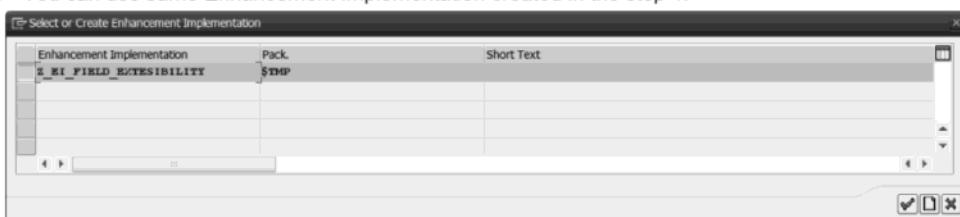
Enhancement Spot: GBAPP_APV_PO



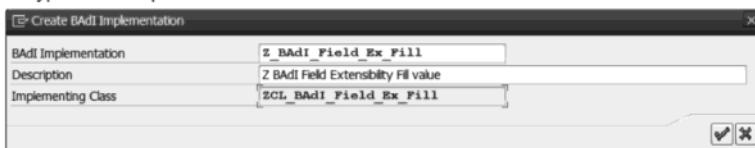
2. Right click on Implementations and select "Create BADI Implementation".



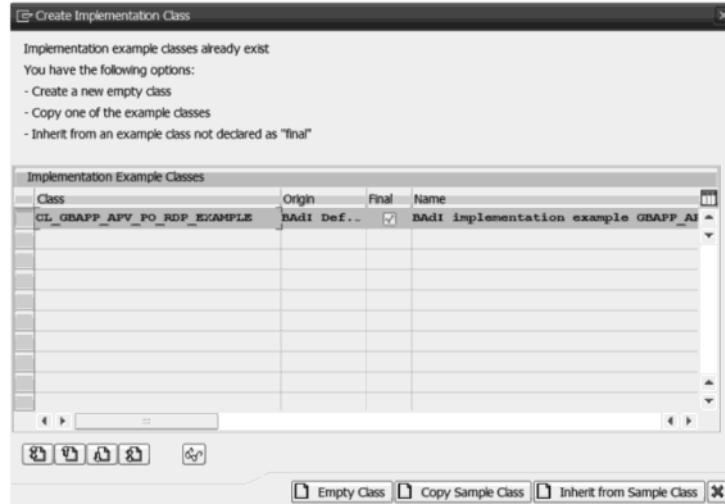
3. You can use same Enhancement Implementation created in the step 1.



4. Type BADI Implementation name

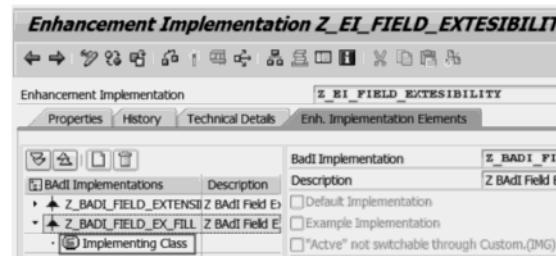


You have option for copying the sample class. In this guide, we copy the sample class.



5. Implement the BAdI

Double click the Implementing Class.



Double click the method CHANGE_HEADER_DETAILS_INFO_API.



You see the sample code with comments.

Class Builder: Class ZCL_BADI_FIELD_EX_FILL Display

```

Method IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API Inactive
1  METHOD IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API.
2
3  * Intended use:
4  * Customer uses the BAdI Enhancement Framework for own content / logik
5
6  * Here you can change/enhance the data, which is retrieved for the GET call for
7  * the Purchase Order Details.
8
9  * Note: If you need to process additional PO data, you can do the changes here
10
11 ENDMETHOD.

```

- Check the Method definition and find the field you have to fill the value.

Select the menu Goto -> Method Definition.

The screenshot shows the SAP Class Builder interface. A context menu is open over the method definition code, with the option 'Method Definition' selected. The main window displays the class definition for 'EX_FILL Display' with the method 'IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API'. The method code is as follows:

```

Method IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API Inactive
1  METHOD IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API.
2
3  * Intended use:
4  * Customer uses the BAdI Enhancement Framework for own content / logik
5
6  * Here you can change/enhance the data, which is retrieved for the GET call for
7  * the Purchase Order Details.
8
9  * Note: If you need to process additional PO data, you can do the changes here
10
11 ENDMETHOD.

```

The 'Attributes' tab of the method definition is selected, showing parameters:

Parameter	Type	Pa... O...	Typing M...	Associated Type
IV_WORKITEM_ID	Importn...	<input type="checkbox"/>	<input type="checkbox"/>	SW_WIID
IV_USER_NAME	Importn...	<input type="checkbox"/>	<input type="checkbox"/>	SYUNAME
IV_PO_NUMBER	Importn...	<input type="checkbox"/>	<input type="checkbox"/>	EBELN
CS_HEADER_DETAILS_X	Changing	<input type="checkbox"/>	<input type="checkbox"/>	GBAPPS_APV_PO_IT_X_HNI

- Open a new session and check the structure GBAPPS_APV_PO_IT_X

ABAP Dictionary: Initial Screen

The screenshot shows the ABAP Dictionary initial screen. The 'Data type' radio button is selected, and the text 'GBAPPS_APV_PO_IT_X_HNI' is entered into the associated input field. Other options like 'Database table', 'View', 'Domain', 'Search help', and 'Type Group' are available but not selected.

ZZEKGRP is there.

Dictionary: Display Structure																
Hierarchy Display Append Structure...																
Structure	GBAPPS_APV_PO_IT_X_HNI		Active													
Short Description	POA : Header Details Nested Structure															
Attributes Components Entry help/check Currency/quantity fields																
31 / 39																
Component	Typing Method	Component Type	Data Type	Length	Decim...	Short Description										
DELIVERY_DATE_ALL	Types	▼ FLAG	CHAR	1	0	General Flag										
HEADER_INFO	1 Types	▼ GBAPPS_PO_HEADER	XXD	0	0	POA : Header Account Details										
.INCLUDE	1 Types	▼ GBAPPS_PO_HEADER	XXD	0	0	POA : Customer Extensions Inc										
DUMMY_PO_HEADER	1 Types	▼ DUMMY	CHAR	1	0	Dummy function in length 1										
.APPEND	1 Types	▼ ZPO_H_DETAILS	XXD	0	0	Append POA Header Details										
ZZEKGRP	1 Types	▼ EKGRP	CHAR	3	0	Purchasing Group										
ITEMDETAILS	1 Types	▼ GBAPPT_PO_ITEM	ITI	0	0	POA : Item Details										
NOTES	1 Types	▼ GBAPPT_NOTES	ITI	0	0	Common : Notes										
ATTACHMENTS	1 Types	▼ GBAPPT_ATTACHMEN	ITI	0	0	Common : Attachments										

You have to fill the Purchasing Group value to the cs_header_details_x-zzekgrp.

8. Go back to the original session.

Class Builder: Class ZCL_BADI_FIELD_EX_FILL Display	
Pattern Pretty Printer Signature Public Section Protected	
Method	IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API
1	METHOD IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API.
2	
3	/* Intended use:
4	/* Customer uses the BAdI Enhancement Framework for own content / logik
5	
6	/* Here you can change/enhance the data, which is retrieved for the GET call for
7	/* the Purchase Order Details.
8	
9	/* Note: If you need to process additional PO data, you can do the changes here
10	
11	ENDMETHOD.

9. Switch to change mode and implement the code.

Program Code:

```
METHOD if_gbapp_ex_apv_po_rdp~change_header_details_info_api.
```

```
DATA: lv_po_number TYPE ebeln.
```

```
lv_po_number = cs_header_details_x-po_number.
```

```
SELECT SINGLE ekgrp FROM ekko INTO cs_header_details_x-
zzekgrp WHERE ebeln = lv_po_number .
```

```
ENDMETHOD..
```

Class Builder: Class ZCL_BADI_FIELD_EX_FILL Change

```

Method IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API Inactive (revised)
1 METHOD IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_API.
2
3 DATA: lv_po_number TYPE ebeln.
4
5 lv_po_number = cs_header_details_x-po_number.
6
7 SELECT SINGLE ekgrp FROM ekko INTO cs_header_details_x-zzekgrp WHERE ebeln = lv_po_number .
8
9 ENDMETHOD.

```

10. Activate the method. Status should be Active.
11. Repeat same step for CHANGE_WORKFLOW_TASK_API method.

Enhancement Implementation Z_EI_FIELD_EXTESIBILITY Display

Enhancement Implementation		Z_EI_FIELD_EXTESIBILITY	Active
		Properties	History
		Technical Details	Enh. Implementation Elements
<input checked="" type="checkbox"/> BADI Implementations	Description		
<input checked="" type="checkbox"/> Z_BADI_FIELD_EXTENSIBILITY Z BADI Field E			
<input checked="" type="checkbox"/> Z_BADI_FIELD_EX_FILL Z BADI Field E			
<input checked="" type="checkbox"/> Implementing Class			
		Implementing Class	IF_GBAPP_EX_APV_PO_RDP
		Interface	ZCL_BADI_FIELD_EX_FILL
		Method	
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_AGENT_LIST_API	Short description Change Data for Agent list
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_ATTACHMENT_CONTENT...	Change Data for Attachment
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_AUTH_CHECK	Change Data for Authority check
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_HEADER_DETAILS_INFO_...	Change Data for Header details
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_INBOX_COUNTER_API	Change Data for GET of Inbox Counter Da...
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_ITEM_DETAILS_INFO_API	Change Data for Item details
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_NOTES_API	Change Data for Notes
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_SUPPLIER_INFO_API	Change Data for Supplier Info
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_USER_DETAILS_API	Change Data for GET of PO Details
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_WORKFLOW_TASKFILE	Change Task Type filter for GET of Workt...
		<input checked="" type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~CHANGE_WORKFLOW_TASK_API	Change Data of Workitem List Data
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~GET_IMAGE_BINARY_FILE	Get binary data for image retrieval
		<input type="checkbox"/> IF_GBAPP_EX_APV_PO_RDP~SET_ATTRIBUTES	Set initial values to Note types, Quantity for

Program Code:

```
METHOD IF_GBAPP_EX_APV_PO_RDP~CHANGE_WORKFLOW_TASK_API.
```

```
DATA: lv_po_number TYPE ebeln.
```

```
FIELD-SYMBOLS: <ls_workflow_list> TYPE GBAPPS_PO_WORKFLOW_TASK.
```

```
LOOP AT CT_WORKFLOW_LIST ASSIGNING <ls_workflow_list>.
```

```
lv_po_number = <ls_workflow_list>-po_number.
```

```
SELECT SINGLE ekgrp FROM ekko INTO <ls_workflow_list>-zzekgrp WHERE ebeln = lv_po_number .
```

```
ENDLOOP.
```

```
ENDMETHOD..
```

4. Test the OData service

Before implementing 3 steps, there was no OData for Purchasing Group.

4.1 Check the OData service - Before

Get the Purchase Order detail

```
/sap/opu/OData/sap/GBAPP_POAPPROVAL/WorkflowTaskCollection('000001105834')/HeaderDetails?$expand=ItemDetails,Notes,Attachments
```

The screenshot shows the SAP NetWeaver Gateway Client interface. The request URL is: /sap/opu/odata/sap/GBAPP_POAPPROVAL/WorkflowTaskCollection('000001105834')/HeaderDetails?\$expand=ItemDetails,Notes,Attachments. The response XML is displayed in the main pane:

```

<d:HeaderDetails>
  <d:SupplierName>C.E.B. BERLIN</d:SupplierName>
  <d:Plant>1000</d:Plant>
</d:HeaderDetails>

```

A callout box with the text "No Purchasing Group" points to the XML entry where a purchasing group would normally be listed.

There was no Purchasing Group in the end of xml.

4.2 Clear the Cache

When you changed the Model, you have to clear Gateway cache in both Gateway and ERP backend.

Transaction: /IWEND/CACHE_CLEANUP on Gateway server

The screenshot shows the 'Cleanup of Model Cache' dialog box. The 'Model Identifier' dropdown is set to 'ZGBAPP_POAPPROVAL'. The 'Cleanup Cache for all Models' checkbox is checked.

Transaction: /IWBEPCACHE_CLEANUP on ERP Server

The screenshot shows the 'Cleanup of Model Cache' dialog box. The 'Cleanup Cache for all Models' checkbox is checked.

4.3 Test the OData service - after

Get the Purchase Order detail

```
/sap/opu/OData/sap/GBAPP_POAPPROVAL/WorkflowTaskCollection('000001105834')/HeaderDetails?$expand=ItemDetails,Notes,Attachments
```

SAP NetWeaver Gateway Client

HTTP Method: GET POST PUT PATCH DELETE HEAD OPTIONS

Request URI: /sap/opu/odata/SAP/GBAPP_POAPPROVAL/WorkflowTaskCollection('000001105834')/HeaderDetails?\$expand=ItemDetails,Notes,Attachments

Test Group: Test Case

HTTP Response

```

Header Name Value
~status_code 200
~status_reason OK

<d:NumberOfNotes>0</d:NumberOfNotes>
<d:NumberOfAttachments>0</d:NumberOfAttachments>
<d:Value>1023.00</d:Value>
<d:Currency>EUR</d:Currency>
<d:SupplierID>1000</d:SupplierID>
<d:SupplierName>C.E.B. BERLIN</d:SupplierName>
<d:NumberOfItems>1</d:NumberOfItems>
<d:Plant>1000</d:Plant>
<d:PlantName>Werk Hamburg</d:PlantName>
<d:DeliveryDate>2013-08-31T00:00:00</d:DeliveryDate>
<d:DeliveryDateAlsoLater />
<d:ZPurchasingGroup>008</d:ZPurchasingGroup>
</m:properties>
</content>
</entry>

```

The Workflow Task entity has also Purchasing Group value.
 /sap/opu/odata/SAP/GBAPP_POAPPROVAL/WorkflowTaskCollection?\$filter=TaskType eq 'TS20000166'&\$top=1

SAP NetWeaver Gateway Client

HTTP Method: GET POST PUT PATCH DELETE HEAD OPTIONS

Request URI: /sap/opu/odata/SAP/GBAPP_POAPPROVAL/WorkflowTaskCollection?\$filter=TaskType eq 'TS20000166'&\$top=1

Test Group: POA Test Case ID: WorkflowTaskCollection \$top and \$skip

HTTP Response

```

Header Name Value
~status_code 200
~status_reason OK

<d:ForwardedByID />
<d:ForwardedByName />
<d:SubstitutingForID />
<d:SubstitutingForName />
<d:CreatedByID>1003866</d:CreatedByID>
<d:CreatedByName>1003866</d:CreatedByName>
<d:Value>1023.00</d:Value>
<d:Currency>EUR</d:Currency>
<d:SupplierID>1000</d:SupplierID>
<d:SupplierName>C.E.B. BERLIN</d:SupplierName>
<d:ItemDescriptions>Actuation</d:ItemDescriptions>
<d:SearchForText />
<d:ZPurchasingGroup>008</d:ZPurchasingGroup>
</m:properties>
</content>
</entry>

```

4.4 UI extensibility

The next step is you are going to display the Purchasing Group field in the UI. Please refer to the separate How to Guide "Adding custom fields to SAP Fiori apps. UI fields extensibility".

Adding custom fields to SAP Fiori apps in 3 steps. OData fields extensibility.



Purchase Orders (9)

PURCHASE ORDERS

REFRESH

Search:

Supplier	Amount	Created
C.E.B. BERLIN	9,990.00 EUR	1 day ago
ZOO	2222	
Olga KREINDLINA		
C.E.B. BERLIN	550.00 EUR	2 days ago
ZOO	2 items	
Albert Manager		
C.E.B. BERLIN	750.00 EUR	2 days ago
ZOO	LAT Test: casing	
Olga KREINDLINA		
C.E.B. BERLIN	5,000.00 EUR	

PURCHASE ORDER

C.E.B. BERLIN 750.00 EUR 2 days ago

Olga KREINDLINA

i

Purchasing Group ZOO

Purchase Order 4500017996

Account Assignment: Cost Center 1110 Executive Board
GL Account 4000000 Consumption, raw material 1

Delivery Date: August 17, 2013

Plant: Werk Hamburg

Payment Terms: 14 days 5%, 30/2%, 45 net

Invoicing: EXW From plant, Walldorf

Company Code: IDES AG (1000)

Items (1)

Description	Quantity	Subtotal
LAT Test: casing	15 PCE	750.00 EUR

Approve Reject



www.sap.com/contactsap

www.sdn.sap.com/irj/sdn/howtогuides



The Best-Run Businesses Run SAP™

[Show TOC](#)

Front-End Server: Activate OData Services

There are several ways to activate the relevant OData services:

- You can activate the OData services for each app individually, as given below.
- You can activate OData services for several apps at once, by using a task list. For more information, see [Activate OData Services for Several SAP Fiori Apps](#).
- You can activate OData services and ICF nodes at once, by using an SAP Fiori app. For more information, see [Activate Apps with SAP Fiori App Implementation Foundation](#).

Procedure

1. Run transaction Activate and maintain services (/IWFND/MAINT_SERVICE) on the front-end server.
2. Choose Add Service.
3. Enter the system alias of your back-end system.
4. In the External Service Name field, enter the technical name of the OData service for your app without the version number. For more information on the OData service per app, see the app-specific documentation in the section *SAP Fiori Apps*.
5. In the Version field, enter the version number.
6. Choose Get Services.
7. Choose Add Selected Services.
8. Enter a technical name for the service in your customer namespace.
9. Assign a package or choose Local Object.
10. Choose Execute to save the service.
11. On the Activate and maintain services screen, check if the system alias is maintained correctly. If not, delete the alias and add the correct one.

Inserted from <[O DATA services Page 46](https://help.sap.com/fiori_bs2013/helpdata/en/55/b840530a3ee447e1000000a441470/content.htm?frameset=/en/11/e168533271a548e1000000a423f68/frameset.htm¤t_toc=/en/11/e168533271a548e1000000a423f68/plain.htm&node_id=10></p></div><div data-bbox=)

Step by Step Guide on SAP Netweaver Gateway and OData |

Tuesday, March 31, 2020 4:05 PM

Sign in

Welcome! Log into your account
your username
your password

[Forgot your password? Get help](#)

[Create an account](#)

[Privacy Policy](#)

Create an account

Welcome! Register for an account

your email

your username

A password will be e-mailed to you.

[Privacy Policy](#)

Password recovery

Recover your password

your email

A password will be e-mailed to you.



- [Home](#)
- [All](#)
- [Tuto](#)
- [Adb](#)
- [A4F](#)
- [CDS](#)
- [VDM](#)
- [UI5](#)
- [Fiori](#)
- [OData](#)
- [HANA-ABAP](#)
- [4m](#)
- [V-Co](#)

- [SoS](#)
- [PAID](#)

[Home](#) Tutorials on OData & SAP Netweaver Gateway

Tutorials on OData & SAP Netweaver Gateway

Video Tutorial on [OData and SAP Netweaver Gateway – Absolutely Free!!](#)



SAP Netweaver Gateway
and OData for Beginners

Step by Step Tutorials on SAP Netweaver Gateway and OData

- [OData and SAP Netweaver Gateway. Part I. Introduction](#)
- [OData and SAP Netweaver Gateway. Part II. Create your first OData Service](#)
- [OData and SAP Netweaver Gateway. Part III. Query Options in OData Service URI](#)
- [OData and SAP Netweaver Gateway. Part IV. Association and Navigation in OData Service](#)
- [OData and SAP Netweaver Gateway. Part V. CRUD Operations in OData Services](#)
- [OData and SAP Netweaver Gateway. Part VI. Frequently Asked Questions](#)
- [OData and SAP Netweaver Gateway. Part VII. Debugging, Trace, Cache Cleanup and F4 Help](#)
- [OData and SAP Netweaver Gateway. Part VIII. SAP's Love for OData – a Tale of the Friendly ABAPer](#)
- [CDS Part 3. Expose CDS Views as OData Service through Annotation](#)
- [OData and SAP Netweaver Gateway. Part IX. How to Add Multiple Entities in One Operation in OData Service](#)
- [OData and SAP Netweaver Gateway. Part X. How to Delete OData Service from Service Catalog?](#)
- [OData and SAP Netweaver Gateway. Part XI. Query Options & HTTP Status Code Summary](#)
- [SAP Netweaver Gateway and OData. Part XII. Media Handling using OData Gateways](#)
- [SAP Netweaver Gateway and OData. Part XIII. Entity Tags in SAP OData Gateways](#)
- [SAP Netweaver Gateway and OData. Part XIV. OData Service using RFC](#)
- [Free Video Course – Introduction to SAP Netweaver Gateway & OData](#)

- [SAPUI5 for ABAPers – Consuming OData Service from SAPUI5 Application – CRUD Operations](#)
 - [Advance SAPUI5 – 2- Push Notification in SAP – ABAP Push Channel, ABAP Messaging Channel in SAPUI5 – a Real Time Interaction](#)
 - [CDS Part 11. How to Consume CDS View in Smart Business Service KPI Fiori Apps?](#)
 - [XSOData and Hana Database Views for Beginners](#)
 - [OData Service from CDS Annotation Not Working in Browser Mode](#)
 - [SAPUI5 – Custom Control in UI5](#)
 - [Using Postman to “POST” without modifying ~CHECK_CSRF_TOKEN in ICF settings.](#)
 - [ABAP Programming Model for SAP Fiori – 4 – OData Service Creation](#)
 - [ABAP Programming Model for SAP Fiori – 5 – OData Service Generation](#)
- SAPYard's YouTube Channel

Video Player

https://youtu.be/9-QSXJz_Qpw

00:00

03:45

[Use Up/Down Arrow keys to increase or decrease volume.](#)

Donate & Support SAPYard

[Donate](#)



Recent Comments

- Vignesh on [CDS Part 18 – Bar Chart & Donut Chart using CDS Views](#)
- tohid786 Shaikh on [Introduction to JavaScript & SAPUI5 for ABAP Developers Training – Details](#)
- Pavan on [Introduction to JavaScript & SAPUI5 for ABAP Developers Training – Details](#)
- Vignesh on [ABAP Programming Model for SAP Fiori – 9 – Use of Determinations in BOPF](#)
- Vignesh on [ABAP Programming Model for SAP Fiori – 8 – Developing Transactional Application – CRUD Operations using BOPF](#)

EDITOR PICKS



[ABAP Programming Model for SAP Fiori – 12 – Scenario of...](#)

March 28, 2020



[ABAP Programming Model for SAP Fiori – 11 – Enabling Draft...](#)

March 28, 2020



[ABAP Programming Model for SAP Fiori – 9 – Use of...](#)

March 28, 2020

POPULAR POSTS



[OData and SAP Netweaver Gateway. Part II. Create your first OData...](#)

January 2, 2017



[ABAP on SAP HANA. Part I. First Program in ABAP HANA](#)

May 19, 2016



[OData and SAP Netweaver Gateway. Part I. Introduction](#)

December 29, 2016

POPULAR CATEGORY

- [Tutorial313](#)
- [ABAP122](#)
- [Tweaks84](#)
- [SAPUI565](#)
- [HANA47](#)
- [HANA-ABAP45](#)
- [Code Snippets41](#)
- [Tips41](#)
- [SAP ABAP on HANA35](#)



ABOUT US

Contact us: mail@sapyard.com

FOLLOW US

- [Forums](#)
- [All SAP Tutorials](#)
- [Tutorials on SAP ABAP on HANA](#)
- [Write & Earn](#)
- [SAP Interview Q&A](#)
- [SAPYard YouTube Channel](#)
- [Shop & Support](#)
- [Video Tutorials](#)
- [Donate](#)

© 2020 TAC Global Business Services LLC

[Sumo](#)

Shares



1

2

3

4

5

Inserted from <<https://sapyard.com/tutorials-on-odata-sap-netweaver-gateway/>>

[Sign in](#)

Welcome! Log into your account

your username

your password

[Forgot your password? Get help](#)

[Create an account](#)

[Privacy Policy](#)

Create an account

Welcome! Register for an account

your email

your username

A password will be e-mailed to you.

[Privacy Policy](#)

Password recovery

Recover your password

your email

A password will be e-mailed to you.



SAPYard's Partners



- [Home](#)
- [All](#)
- [Tuto](#)
- [Adb](#)
- [A4E](#)
- [CDS](#)
- [VDM](#)
- [UI5](#)
- [Fiori](#)
- [OData](#)
- [HANA-ABAP](#)
- [4m](#)
- [V-Co](#)
- [SoS](#)
- [PAID](#)

[Home](#) [Tutorial](#) [OData & SAP Netweaver Gateway](#) OData and SAP Netweaver Gateway. Part I. Introduction

[Tutorial](#)

[OData & SAP Netweaver Gateway](#)

OData and SAP Netweaver Gateway. Part I. Introduction

By

[SAP Yard](#)

-
December 29, 2016

[15](#)

15846

[Facebook](#)

[Twitter](#)

[Pinterest](#)

[WhatsApp](#)



SAP HANA as the database, S/4 HANA and then **SAP ABAP on HANA** for technical folks like us is one area which we need to keep ourselves abreast with. But there is another cluster of **OData, SAP NetWeaver Gateway, SAPUI5 and SAP Fiori** which every ABAPer will stumble upon in his/her projects sooner if not later. After the relatively exhaustive "[**Fundamental Tutorial Series on SAP ABAP on HANA**](#)" (hopefully, soon we will have Advance Tutorial Series on ABAP on SAP HANA), we received numerous requests from our readers to put a series on **OData and NetWeaver Gateway**.

As there are some really good tutorials on this subject in SCN and other blogs, initially we deferred the idea to write on the same topic. *But now, with the repetitive request from our beloved subscribers, we would try our best to put the series on OData and NetWeaver Gateway in a simple and systematic way so that all the beginners on this topic can learn and benefit from our page.*

BREAKING NEWS!!! – [End To End Video Course on SAP OData – Absolutely FREE!!!!](#)



SAP Netweaver Gateway
and OData for Beginners

Let's start.

OData

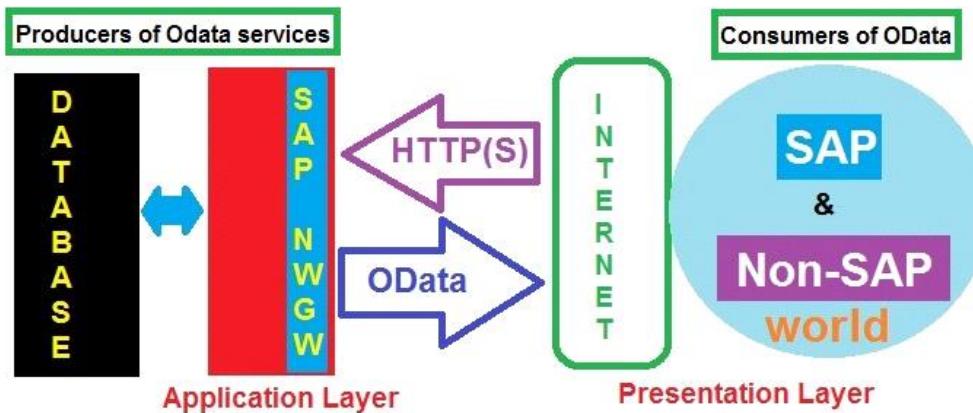
As everyone says, **OData** is the acronym for "**Open Data Protocol**". The protocol is a set of rules which every player has to follow. To put it straight, *OData interface is an open standard that can be consumed by any application, program, software or device of the Non-SAP World that can connect with SAP using the **HTTP(s) protocol** and can manipulate (read, write, modify and understand i.e. parse and construct) an XML document*. Since the protocol is HTTP based, *any programming language with HTTP stack can be used for consuming OData services*.

In other words, **OData** is a Web protocol for unlocking and sharing data; freeing it from silos that exist in some software applications. The OData protocol supports serialization in multiple popular formats, including JSON and Atom/XML. With OData, developers are able to build cross-platform Web and mobile Applications.

With OData, organizations can develop services with the high levels of *data integration and cross-platform interoperability* that are required by the modern day complex business.

SAP Netweaver Gateway

The below figure shows that **SAP NetWeaver Gateway** sits in SAP Application Layer. It is the Window for outside world to peep into SAP and transfer data to/from SAP. **Outer world** can send **HTTP(s)** message and **SAP** would provide them with **OData**. Also note, OData is an open source to exchange data over the Internet. **SAP Netweaver Gateway** (blue box within Application Layer in the below picture) **is a technology that seamlessly connects devices, platforms and environments to SAP Enterprise Data using the OData services**. SAP Netweaver Gateway offers connectivity to SAP Business data using any programming language and without the need of strong SAP development knowledge.



Both Client side (Outer World) and Server side (SAP) development can be in completely different programming languages as long as both are able to communicate with each other via HTTP.

Clients that consume the service to query and manipulate the data from OData Services are also called as "**Consumers**" of OData Service. Similarly, **Servers** that expose the OData services via endpoints are known as "**Producers**" of OData services.

Prior to OData, external non-SAP developers have connected to SAP using RFC/BAPI or web services. In those cases, the non-SAP developers (Web developers) needed to know beforehand the structures of the data passed from SAP. The non-SAP developers needed to have at least a basic knowledge of the internal workings of an SAP system. But the scenario has changed with OData.

Advantages of OData for Programmers and Developers.

i. The OData interface is implemented using XML or JSON. Both of these formats are well known, plain text protocols for the transmission of information over the Web.

ii. OData message is self-describing. So any non-SAP Web developer can understand the content of the OData message without the knowledge of ABAP or how SAP works.

It is understood by now that **Server hosts the Data** and **Clients can call the Service** to retrieve the resources and manipulate them. Servers expose one or more endpoints which are Services that refers to the Resources. Clients or non-SAP World developers only need to know the Server side endpoints to call the service to query or manipulate the data.

With the advent of OData, the communication barrier between SAP and Non-SAP Web Developers is removed. It is an Open book now. There is no cost or license agreement needed for the use of OData.

FYI: Microsoft originally developed and introduced **OData**. Not SAP. Surprise for an ABAPer. ☺ . Citrix Systems Inc., IBM Corp., Microsoft Corp., Progress Software, SAP AG, WSO2 etc collaborated together to standardize the **OData** for implementation of a

RESTful API. Now OData is managed by the [Oasis Organisation](#).

Why is OData called ODBC (Open Database Connectivity) of the Web?

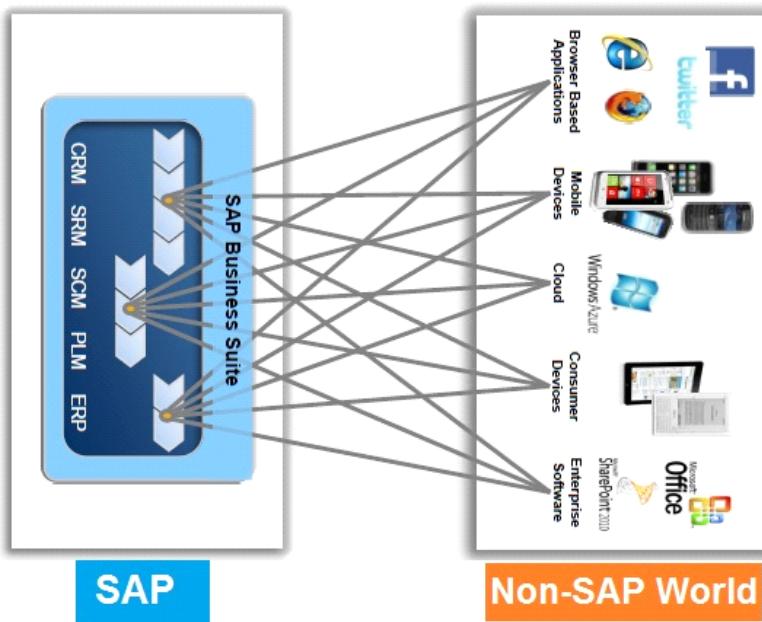
Ans: OData offers database-like access to server-side resources. Hence, OData is also described as "ODBC for the Web". Confused? Let's simplify it. ☺

ODBC is a standard API to access the DBMS, independent of the database management systems or operating systems. ODBC achieves this by adding drivers between the Application layer and the DBMS to translate the queries requested by the applications into instructions which DBMS can understand. Similarly, OData acts like middleware between producers and consumers to communicate data. There is a uniform way to consume data and is independent of the producer (SAP or Non-SAP OData) much like ODBC.

What was the need for OData in SAP?

Ans: Before the introduction of OData, there was the "**Point-to-Point**" solution for SAP to Non-SAP integration. One application for two different organizations or platforms needed two different design in SAP. *This led to duplication of work, effort, time and money.*

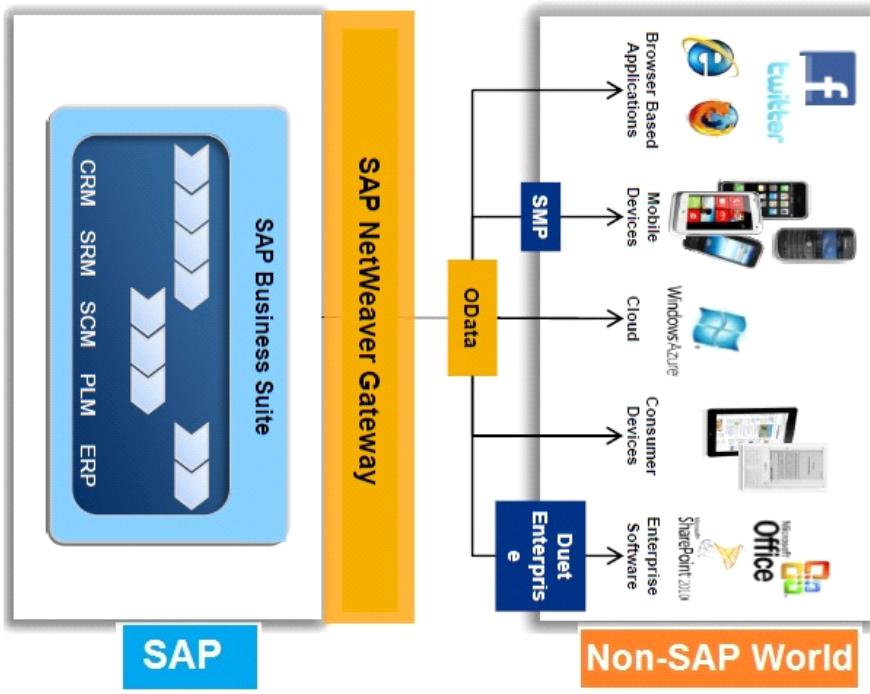
Check the below image. For pulling the same data from SAP, there are multiple interfaces created to cater the need of multiple end users (browsers/mobile/cloud/custom devices/software etc).



Point-to-point Solutions

This means there was **poor scalability, increased system landscape complexity and increased administration effort.**

The alternative to Point-to-Point solution is "**One Data Model -> One API -> Multiple End-User Experiences**". Please check the image below. One OData service along with SAP NetWeaver Gateway suffices all the needs of multiple end applications.



1 Data Model → 1 API → Multiple End-User Experiences

This approach provides one solution to any environment, any platform and any experience. Moreover, no SAP knowledge required for consumption of OData.

What is the nearest competitor of OData?

Ans: **GData** from Google.

OData's **extensibility feature** was one of the most important reasons why SAP chose OData over GData. This is particularly useful in cases where *SAP specific values need to be described in this protocol*; for instance, currency fields. A currency field contains two separate values, the *currency amount and the currency code*. These two values must always be treated as a linked pair and OData's extensibility allows for this.

HTTP (Hyper Text Transfer Protocol) is an integral part of OData. So it deserves a small mention. ☺

HTTP is based on *Client-Server Architecture*. The *Browser is the Client* which sends HTTP request and *Web Server is the Server* which sends the response back to the browser. **HTTP** defines "WHAT" can be transferred between Client and Server. "HOW" the data packets are transferred via HTTP is handled by **TCP/IP** protocols.

What is stateless?

Ans: Every single HTTP request that is received by the Web Server is forgotten after a response has been sent across. Web Servers do not remember or recall the previous request. This is called stateless.

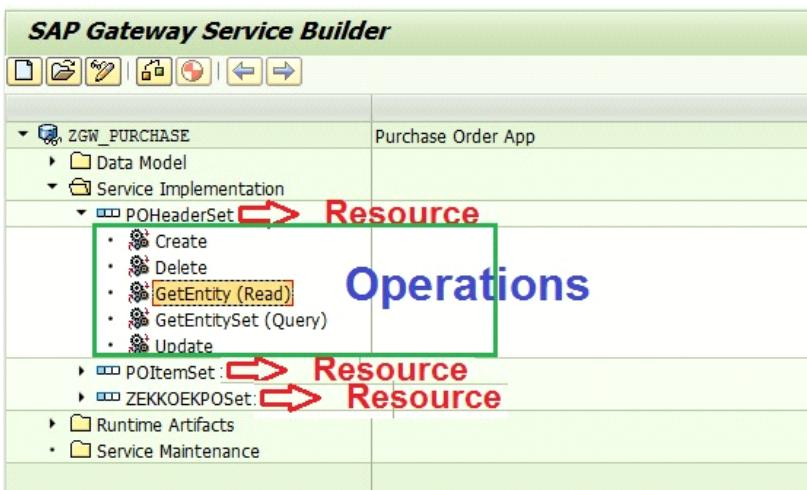
What is RESTful?

Ans: OData is REST-inspired technology for reading, writing, and modifying information on the Web (not just SAP). **REST = REpresentational State Transfer**. *REST is an architectural style that uses simple and lightweight mechanism for inter-machine communication*. It is an alternative to the RPC (Remote Procedure Calls) and Web Services. REST is **Resource-based**, unlike RPC or SOAP which are **Action-based**.

REST services are called as REST services because the Services are really working with **Resources** instead of **Operations**. Any communication between client and services are using **URI** (Unified Resource Identifier) over HTTP protocol using HTTP method. The URI is really the representation of the Resources (like POHeader, POItem, Customer, Vendor etc). Also, in RESTful service, once you identified the Resource, you will be working with a uniform interface, because it uses HTTP methods (GET, PUT, POST and DELETE) to work with the resource. *So, the client does not need to know what the exact operation name defined in the service contract to call that method*. GET method is

used whenever we need to get the representation of an existing resource. POST is used to add new resource into the system. PUT is to modify the existing resource and DELETE is to remove the resource from the system. *No matter what is the Service in whatever Platform, GET, PUT, POST, DELETE remains the same.*

Still confusing? Let us try to understand with the below real example (*Do not worry, how we created it. We will learn how to create them in our next post*).



In the above OData Service, POHeaderSet, POItemSet, ZEKKOEKPOSet are "**Resource**" names. Create, Delete, GetEntity, GetEntitySet etc are the "**Operation**" names. While calling this OData services, the outside world (Client) needs to call the **URI** with the Resource details. Also, outside world only needs to use GET, POST, PUT etc verbs to pull data, create data and update data etc at the server side. These verbs are uniform and standardized for all OData. Clients do not need to know the server side actual methods or operations like Create / GetEntity / GetEntitySet etc which are called behind the scene.

```

<id>http://txaixegd01.../sap/opu/odata/sap/ZGW_PURCHASE_SRV/POHeaderSet('6000000254')</id>
<title type="text">POHeaderSet('6000000254')</title>
<updated>2016-12-28T21:38:02Z</updated>
<category term="ZGW_PURCHASE_SRV.POHeader">
  ...
<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:Ebeln>6000000254</d:Ebeln>
    <d:Bukrs>0208</d:Bukrs>
    <d:Bstyp>A</d:Bstyp>
    <d:Bsart>AN</d:Bsart>
    <d:Loekz />
    <d:Statu />
  </m:properties>
</content>

```

It is also important to note here that it is possible to have **multiple URIs pointing to the same Resource** with the same output. Let us check it with the below example.

URI 1:

/sap/opu/odata/sap/ZGW_PURCHASE_SRV/POItemSet(Ebeln='4500004723',Ebelp='0010')

SAP Gateway Client

HTTP Method: GET

Request URI: /sap/opu/odata/sap/ZGW_PURCHASE_SRV/POItemSet(Ebeln='4500004723',Ebelp='00010') **URI 1**

Protocol: HTTP

HTTP Request

HTTP Response - Processing Time = 254 ms

Header Name	Value
~status_code	200
~status_reason	OK

```

<entry>
  <id>http://schemas.microsoft.com/ado/2007/08/dataservices/related/POItemSet('4500004723',00010)</id>
  <content type="application/atom+xml;type=entry" title="POItemToTableUpdNav" />
    - <content type="application/xml">
      <m:properties>
        <d:Ebeln>4500004723</d:Ebeln>
        <d:Ebelp>00010</d:Ebelp>
        <d:Loekz />
        <d:Statu />
        <d:Txz01>SRINITEST 7</d:Txz01>
        <d:Matnr />
        <d:Werks>4030</d:Werks>
        <d:Lgort />
        <d:Ktmng>0.000</d:Ktmng>
        <d:Menge>1.000</d:Menge>
        <d:Meins>SU</d:Meins>
      </m:properties>
    </content>
  </entry>

```

Same data

URI 2:

/sap/opu/odata/sap/ZGW_PURCHASE_SRV/POHeaderSet('4500004723')/HeaderToItemNav

SAP Gateway Client

HTTP Method: GET

Request URI: /sap/opu/odata/sap/ZGW_PURCHASE_SRV/POHeaderSet('4500004723')/HeaderToItemNav **URI 2**

Protocol: HTTP

HTTP Request

HTTP Response - Processing Time = 412 ms

Header Name	Value
~status_code	200
~status_reason	OK

```

<entry>
  <id>http://schemas.microsoft.com/ado/2007/08/dataservices/metadata</id>
  <content type="application/atom+xml;type=entry" title="HeaderToItemNav" />
    - <content type="application/xml">
      <m:properties>
        <ns0:Ebeln>4500004723</ns0:Ebeln>
        <ns0:Ebelp>00010</ns0:Ebelp>
        <ns0:Loekz />
        <ns0:Statu />
        <ns0:Txz01>SRINITEST 7</ns0:Txz01>
        <ns0:Matnr />
        <ns0:Werks>4030</ns0:Werks>
        <ns0:Lgort />
        <ns0:Ktmng>0.000</ns0:Ktmng>
        <ns0:Menge>1.000</ns0:Menge>
        <ns0:Meins>SU</ns0:Meins>
      </m:properties>
    </content>
  </entry>

```

Same data

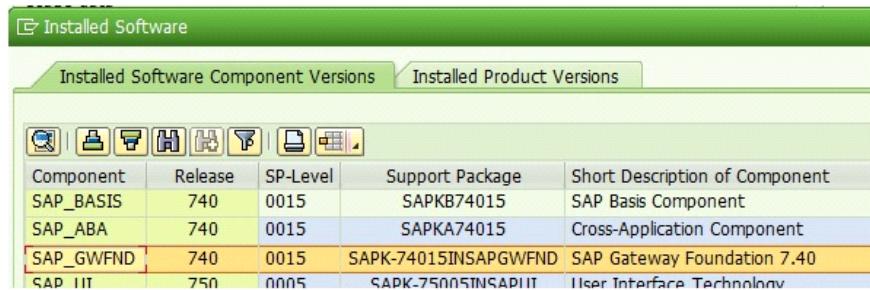
The above two URIs point to the same Resource.

How to check if your system has SAP Netweaver Gateway?

Ans: At NW 7.3 and prior to NW 7.3, SAP Netweaver Gateway had three add-ons namely **GW_CORE**, **IW_FND** and **IW_BEP**. The **GW_CORE** and **IW_FND** components were required for Gateway Server functionalities, whereas **IW_BEP** was used for

Gateway Backend functionalities.

After 7.4 release, all the three components are bundled into a single component **SAP_GWFND** called Gateway Foundation.



Component	Release	SP-Level	Support Package	Short Description of Component
SAP_BASIS	740	0015	SAPKB74015	SAP Basis Component
SAP_ABA	740	0015	SAPKA74015	Cross-Application Component
SAP_GWFND	740	0015	SAPK-74015INSAPGWFND	SAP Gateway Foundation 7.40
SAP_IIT	750	0005	SAPK-75005INSAPII	User Interface Technology

This is just the introduction of OData and SAP Netweaver Gateway along with some related terminologies. In the subsequent parts, we will dive deeper in the actual practical scenarios of OData. We will learn how to create OData services, play with the GET, POST, PUT etc operations along with the backend methods which need to be re-defined to meet our requirement. We will also check **Navigations** and **Associations** of Data Models and learn how to handle them. At the end, we will build SAPUI5 Applications to consume our OData service.

Next Post: [SAP Netweaver Gateway and OData Tutorial Part II: Create your first OData Service](#)

This would be an interesting and hands-on learning series on OData and SAP Netweaver Gateway. So get hold of your SAP systems and invest 20 minutes behind each Tutorial Part and learn the fundamentals along with us.

If you want to get such useful articles directly to your inbox, please **SUBSCRIBE**. We respect your privacy and take protecting it seriously.

*If you liked this post, please hit the **share** buttons and like us on [facebook](#).*

Do you have anything to add to this article? Have you faced any issue understanding OData or SAP Netweaver Gateway? Do you want to share any real project requirement or solutions? Please do not hold back. **Please leave your thoughts in the comment section.**

Thank you very much for your time!!

[Call for Guest Authors and Contributors to write SAP Articles on our page and get noticed.](#)

*Do you have any tips or tricks to share? **Do you want to write some articles at SAPYard?** Please [REGISTER](#) and start posting and sharing your knowledge to the SAP world and get connected to your readers.*

Step by Step Tutorials on SAP Netweaver Gateway and OData

- [OData and SAP Netweaver Gateway. Part I. Introduction](#)
- [OData and SAP Netweaver Gateway. Part II. Create your first OData Service](#)
- [OData and SAP Netweaver Gateway. Part III. Query Options in OData Service URI](#)
- [OData and SAP Netweaver Gateway. Part IV. Association and Navigation in OData Service](#)
- [OData and SAP Netweaver Gateway. Part V. CRUD Operations in OData Services](#)
- [OData and SAP Netweaver Gateway. Part VI. Frequently Asked Questions](#)
- [OData and SAP Netweaver Gateway. Part VII. Debugging, Trace, Cache Cleanup and F4 Help](#)
- [OData and SAP Netweaver Gateway. Part VIII. SAP's Love for OData – a Tale of the Friendly ABAPer](#)
- [CDS Part 3. Expose CDS Views as OData Service through Annotation](#)
- [OData and SAP Netweaver Gateway. Part IX. How to Add Multiple Entities in One Operation in OData Service](#)

- [OData and SAP Netweaver Gateway. Part X. How to Delete OData Service from Service Catalog?](#)
- [OData and SAP Netweaver Gateway. Part XI. Query Options & HTTP Status Code Summary](#)
- [SAP Netweaver Gateway and OData. Part XII. Media Handling using OData Gateways](#)
- [SAP Netweaver Gateway and OData. Part XIII. Entity Tags in SAP OData Gateways](#)
- [SAP Netweaver Gateway and OData. Part XIV. OData Service using RFC](#)
- [Free Video Course – Introduction to SAP Netweaver Gateway & OData](#)
- [SAPUI5 for ABAPers – Consuming OData Service from SAPUI5 Application – CRUD Operations](#)
- [Advance SAPUI5 – 2- Push Notification in SAP – ABAP Push Channel, ABAP Messaging Channel in SAPUI5 – a Real Time Interaction](#)
- [CDS Part 11. How to Consume CDS View in Smart Business Service KPI Fiori Apps?](#)
- [XSOData and Hana Database Views for Beginners](#)
- [OData Service from CDS Annotation Not Working in Browser Mode](#)
- [SAPUI5 – Custom Control in UI5](#)
- [Using Postman to “POST” without modifying ~CHECK_CSRF_TOKEN in ICF settings.](#)
- [ABAP Programming Model for SAP Fiori – 4 – OData Service Creation](#)
- [ABAP Programming Model for SAP Fiori – 5 – OData Service Generation](#)

Also, Check Step by Step Tutorials on SAPUI5

- [Journey to SAPUI5](#)
- [SAPUI5 Tutorial with WebIDE. Part I. How to Consume Custom OData in SAPUI5 Application](#)
- [SAPUI5 Tutorial with WebIDE. Part II. Routing and Navigation in SAPUI5 Application](#)
- [SAPUI5 Tutorial with WebIDE. Part III. Drop Down in SAPUI5 Applications \(2 Methods\)](#)
- [SAPUI5 Tutorial. Part IV with WebIDE. Routers and Routing in SAPUI5](#)
- [SAPUI5 Tutorial with WebIDE. Part V. Navigation in SAPUI5 without Routers](#)
- [SAPUI5 Tutorial with WebIDE. Part VI. Using Fragments in SAPUI5 Fiori Applications](#)
- [SAPUI5 Tutorial with WebIDE. Part VII. An ABAPer’s First SAPUI5 App in SAP WebIDE](#)
- [SAPUI5 Tutorial with WebIDE. Part VIII. Deploy my First SAPUI5 App in WebIDE](#)
- [SAPUI5 Tutorial with WebIDE. Part IX. Alternative to oModel.setSizeLimit\(\)](#)
- [SAPUI5 Tutorial with WebIDE. Part X. Using Media Queries in UI5 Application](#)
- [SAPUI5 Tutorial with WebIDE. Part XI. An ABAPer’s Second SAPUI5 App](#)
- [SAPUI5 Tutorial with WebIDE. Part XII. SAPUI5 Basic Debugging for Beginners](#)
- [Routing and Navigation in SAP UI5 – Theoretical Explanation Part 1](#)
- [Routing and Navigation in SAP UI5 – Theoretical Explanation Part 2](#)
- [Add Delete & Save Multiple Records in SAPUI5. Part 1 – ADD](#)
- [Add Delete & Save Multiple Records in SAPUI5. Part 2 – DELETE](#)
- [Add Delete & Save Multiple Records in SAPUI5. Part 3 – SAVE](#)
- [Modularization and Large Scale Architecture in SAPUI5](#)
- [Use of Third Party \(or\) External Resources in SAPUI5. Part I – Overview](#)
- [Use of Third Party \(or\) External Resources in SAPUI5. Part II – Practical with Gauges](#)
- [How to Deploy UI5 App without LPD_CUST?](#)
- [SAPUI5 For ABAPers – Component Reuse with Real Time Example](#)
- [SAPUI5 for ABAPers – Consuming OData Service from SAPUI5 Application – CRUD Operations](#)
- [Advance SAPUI5 – 1- Trick to Send QR code or Barcode Data Remotely from Android to PC for SAPUI5 App](#)
- [Advance SAPUI5 – 2- Push Notification in SAP – ABAP Push Channel, ABAP Messaging Channel in SAPUI5 – a Real Time Interaction](#)
- [Advance SAPUI5 – 3 – How to send e-mail in SAPUI5 Hybrid App?](#)

- [Advance SAPUI5 – 4 – How to Get Weight from the Weigh Scale/Weigh Bridge Bluetooth Device using SAPUI5 Hybrid App?](#)
- [SAPUI5 – How to Change the Master List Item Selection Based on Changes in Hash Tag URL?](#)
- [SAPUI5 – How to Load a Full Screen and Navigate to Master-Detail Application?](#)

Approach 1

We have a very active Telegram SAP Technical Group. Please join it using below link.

Telegram SAP Technical Discuss Group.

WhatsApp had the limitation of 256 members per group and we were finding a hard time syncing and maintaining information in multiple WhatsApp group. Therefore we have moved to Telegram as it can accommodate more than 10,000 users. [Please join.](#)

- TAGS
- [Introduction about SAP Netweaver Gateway](#)
- [Introduction to OData](#)
- [OData in SAP NetWeaver Gateway](#)
- [RESTful](#)
- [SAP OData Tutorial](#)
- [Stateless](#)
- [Step-by-step guide to build an OData Service](#)

[Facebook](#)

[Twitter](#)

[Pinterest](#)

[WhatsApp](#)

Previous article [ABAP on SAP HANA. Part XII. Open SQL, CDS or AMDP, which Code to Data Technique to use?](#)

Next article [Data Lake and Data Warehouse](#)



[SAP Yard](#)

<https://sapyard.com/>

SAPYard is one stop page for all Technical Folks in SAP. You would find un-conventional explanations, tutorials, tricks and end to end Free SAP Video Courses and Training. Please like our [Facebook Page](#), follow us at [Twitter](#), [Instagram](#) and also join our [LinkedIn Group](#). Please Subscribe to our [Youtube Channel](#) for Free SAP Video Trainings.

[RELATED ARTICLES](#)[MORE FROM AUTHOR](#)



[ABAP Programming Model for SAP Fiori – 12 – Scenario of Transactional App with Draft Enabled Functionality](#)



[ABAP Programming Model for SAP Fiori – 11 – Enabling Draft Functionality for Transactional Apps](#)



[ABAP Programming Model for SAP Fiori – 10 – Use of Actions in BOPF](#)

15 COMMENTS



Abhi [January 18, 2020 At 2:15 pm](#)

Hi, How to pass value to the blank field in a table?

[Reply](#)



Nilesh [September 12, 2018 At 5:35 am](#)

Dear SAPYard,

Thanks for posting OData tutorials, its very easy to understand. I am having one question.

How to check our system i.e(DEV or QAS) is Enabled or disabled for OData. Do we need to make any configuration changes.

[Reply](#)



Khan [March 28, 2018 At 10:54 pm](#)

Hi,

Good work!!! Really liked this website. It really helped me a lot.

Any possibility to include Implementation of Workflow in ODATA service?

Thanks & Regards,

Khan

[Reply](#)



[SAP Yard](#) March 29, 2018 At 9:49 am

Thank you Mam854. We are glad it helped you.

Once you re-define the Methods, implementing Workflow is same as you would do in normal ABAP. If you could reveal what you are trying to do, we might suggest something.

Also, if you want to share any article which you have recently learned, please write us at mail@sapyard.com.

Regards,

Team SAPYard.

[Reply](#)



[YD](#) January 12, 2018 At 1:05 am

thanks guy, Very Nice!

[Reply](#)



[SAP Yard](#) January 12, 2018 At 10:10 am

Thank you YD. Please keep visiting. Please share our link with your colleagues, team members and friends.

Regards,

Team SAPYard.

[Reply](#)



[Ashish](#) October 29, 2017 At 1:25 am

Awesome blog. Thanks a ton.

[Reply](#)



[Sam](#) October 16, 2017 At 7:49 am

Hi, I have been going through the ODATA series in SAP Yard and have a couple of questions in mind so far.

Could you please clarify.

- 1) What is extensibility feature in odata.
- 2) Why not write code in dpc class instead of extension??
- 3) How to pass multiple values in URI – for e.g. 2 PO numbers
- 4) Why not have a create entity set, we should be able to create multiple records as well.

[Reply](#)



[Venkat](#) January 10, 2017 At 6:43 am

HI Team,

Can you please explain like, when do we use HTTP/HTTPS protocols?

Regards,

Venkat

[Reply](#)



[sapyard](#) January 10, 2017 At 3:33 pm

Dear Venkat – Browser is the Client and the computer which has information about the site you are accessing is the Server. It is Client-Server Architecture. HTTP & HTTPS are protocols. It cares about how information is presented to the user of the computer. It monitors your clicks and other activities on the browser. But it does not tell how data is moved from Client to Server and vice-versa.

HTTP and HTTPS are same till now.

But there is a risk in HTTP. You are accessing a public data in HTTP and others can track and monitor your activity. People with malicious intention can disguise as you and send/receive information from the server of your behalf. So whenever doing banking transaction or passing any sensitive info, always check the protocol is HTTPS.

HTTPS differentiates one sender and receiver from another. So a third person cannot see your data or pretend to be you. How HTTPS can do it? It takes help of a third party which acts like Police Department. HTTPS works along with another protocol, called Secure Sockets Layer (SSL), to transport data safely. SSL encrypts all incoming and outgoing data using some complex mathematical algorithm to hide the actual meaning of the data coming and going from client and server. So theoretically, it is not possible for hackers to see what you typed.

In short, HTTPS website owner has security clearance from the Department of Defense (SSL) in the form of Certificates which are time-sensitive. When the browser sees the secure Website, it uses the information in the certificate to verify that the site is what it claims to be and the information passed to server is not from other phishing sites.

Hope it is clear.

HTTP(s) = Hypertext Transfer Protocol (secure)

Regards,

Team SAPYard.

[Reply](#)



[Marco van de Sande](#) January 9, 2017 At 7:27 am

Hi Team SAPYard,

Thanks for the answer.

Where can I find the navigation, in the transaction SGEW ?

Maybe you have an example picture ?

Regards,

Marco

[Reply](#)



[Marco van de Sande](#) January 6, 2017 At 7:47 pm

Great blog!

We face a Real project requirement.

We use OData in combination with SAP BW Queries. In the query you can mark it as an odata query. It works fine. However we have 1 issue.

There is an hierarchy in the query, which parameter/ code must the non sap consumer

use, to see/ rebuild the hierarchy on the non sap consumer side ?

Hopefully you can help us.

[Reply](#)



SAP Yard [January 9, 2017 At 4:41 am](#)

Dear Marco – Do you not have navigation/association for the hierarchy. If you look closely, you should find the navigation set to view the hierarchy of data.

Regards,

Team SAPYard.

[Reply](#)



Joy Claxino [January 3, 2017 At 10:36 am](#)

Dear SAP Yard,

Thanks for posting oData, i just started learning all these topics. nice blog 😊

[Reply](#)



sapyard [January 3, 2017 At 4:00 pm](#)

Dear Joy – Thank you very much for your feedback. Glad you like our blog. We will cover OData, Netweaver Gateway and then user them in SAPUI5 Applications.

Please stay tuned.

Regards,

Team SAPYard.

[Reply](#)

Please help us improve. Please Shout Often & Loud!!

This site uses Akismet to reduce spam. [Learn how your comment data is processed](#).

SAPYard's YouTube Channel

Video Player

https://youtu.be/9-QSXJz_Qpw

00:00

03:45

[Use Up/Down Arrow keys to increase or decrease volume.](#)

Donate & Support SAPYard

[Donate](#)



Recent Comments

- Vignesh on [CDS Part 18 – Bar Chart & Donut Chart using CDS Views](#)
- tohid786 Shaikh on [Introduction to JavaScript & SAPUI5 for ABAP Developers Training – Details](#)
- Pavan on [Introduction to JavaScript & SAPUI5 for ABAP Developers Training – Details](#)
- Vignesh on [ABAP Programming Model for SAP Fiori – 9 – Use of Determinations in BOPF](#)
- Vignesh on [ABAP Programming Model for SAP Fiori – 8 – Developing Transactional Application – CRUD Operations using BOPF](#)

EDITOR PICKS



[ABAP Programming Model for SAP Fiori – 12 – Scenario of...](#)

March 28, 2020



[ABAP Programming Model for SAP Fiori – 11 – Enabling Draft...](#)

March 28, 2020



[ABAP Programming Model for SAP Fiori – 9 – Use of...](#)

March 28, 2020

POPULAR POSTS



[OData and SAP Netweaver Gateway. Part II. Create your first OData...](#)

January 2, 2017



[ABAP on SAP HANA. Part I. First Program in ABAP HANA](#)

May 19, 2016



[OData and SAP Netweaver Gateway. Part I. Introduction](#)

December 29, 2016

POPULAR CATEGORY

- [Tutorial313](#)
- [ABAP122](#)
- [Tweaks84](#)
- [SAPUI565](#)
- [HANA47](#)

- [HANA-ABAP45](#)
- [Code Snippets41](#)
- [Tips41](#)
- [SAP ABAP on HANA35](#)



ABOUT US

Contact us: mail@sapyard.com

FOLLOW US

- [Forums](#)
- [All SAP Tutorials](#)
- [Tutorials on SAP ABAP on HANA](#)
- [Write & Earn](#)
- [SAP Interview Q&A](#)
- [SAPYard YouTube Channel](#)
- [Shop & Support](#)
- [Video Tutorials](#)
- [Donate](#)

© 2020 TAC Global Business Services LLC

[Sumo](#)

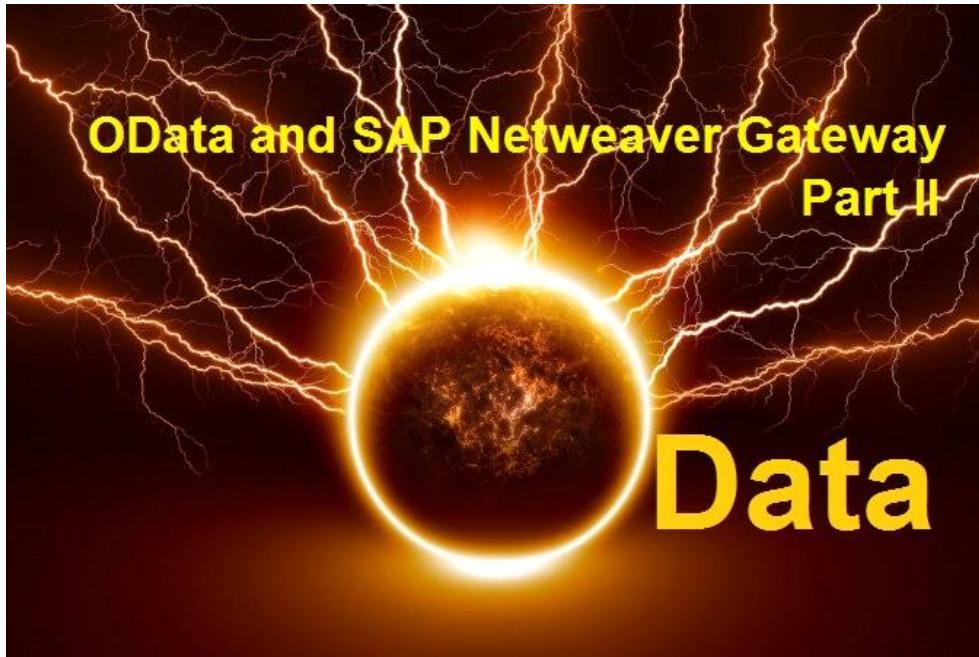
Shares



Inserted from <<https://sapyard.com/odata-and-sap-netweaver-gateway-part-i-introduction/>>

OData and SAP Netweaver Gateway. Part II. Create your first OData Service |

Tuesday, March 31, 2020 4:07 PM



In the Part I post [Introduction to SAP Netweaver Gateway and OData](#), we learned about the definition, terminologies and concept of SAP Netweaver Gateway, OData and HTTPs. Now we have a fair idea of the **RESTful** and **STATEless** concept. In this article, we will do a hands-on exercise and generate our first OData Service in SAP and play around with the different URIs of the service. *These URIs are the endpoints of Server side of OData which the outside world (Client side) would need to call using HTTP(s) in order to update/extract data to/from SAP.* In SAPUI5 applications, we need to consume these URIs to connect to SAP and play with its data. We will re-define the standard methods and add our custom ABAP logic to meet our requirements.

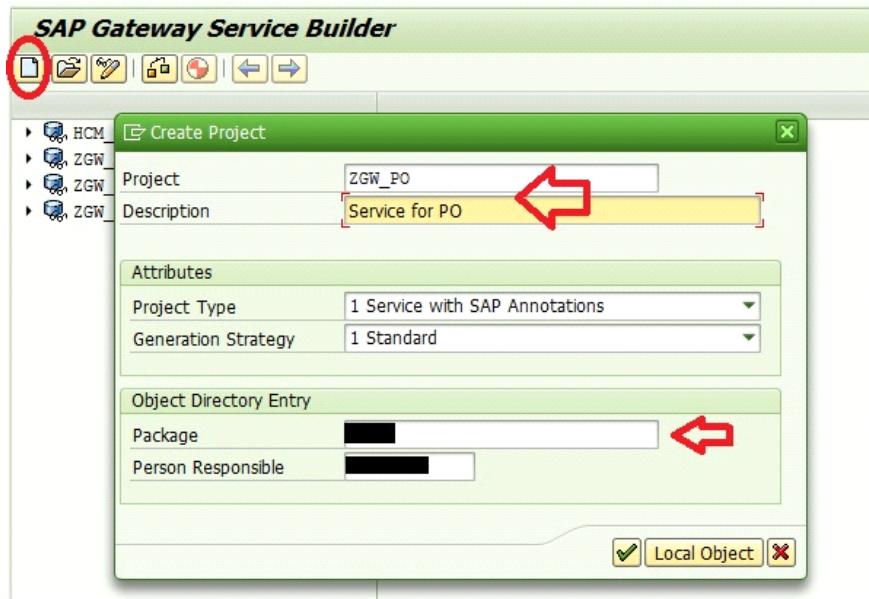
BREAKING NEWS!!! – [End To End Video Course on SAP OData – Absolutely FREE!!!!](#)



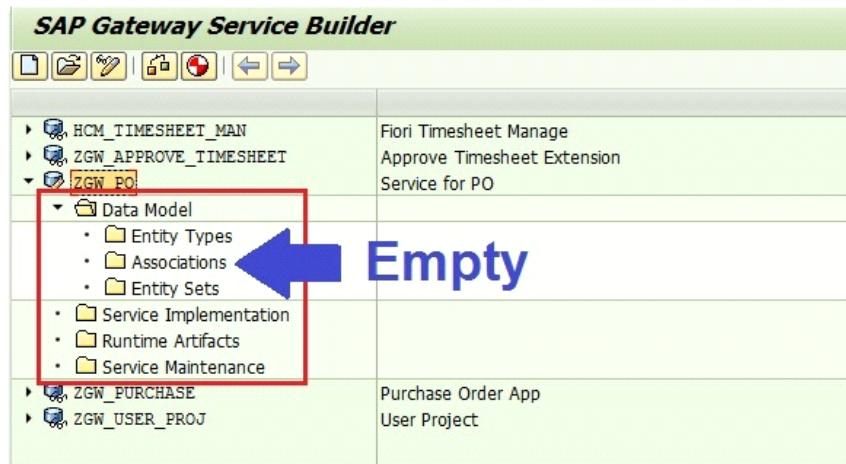
SAP Netweaver Gateway
and OData for Beginners

A. Define Data Model

Go to t-code **SEGW** (SAP Gateway Service Builder). Remember it like SE38. Instead of 38, add **GW** for **GateWay**. So our friendly t-code for today's topic is SEGW. Hit the Create Icon and provide the name of the Project, description and package (or local) and save it.



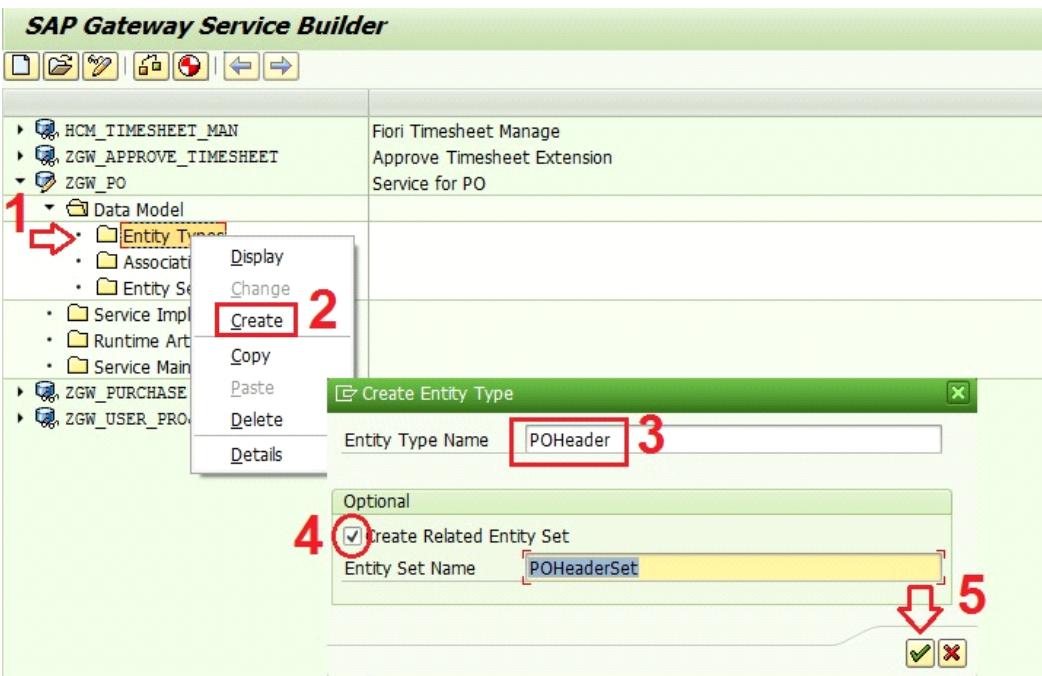
The project gets created with four folders, namely **Data Model, Service Implementation, Runtime Artifacts and Service Maintenance**. Please take note that Data Model further has three sub-folders viz **Entity Types, Associations and Entity Sets**. All the folders are empty by default.



Before you ask yourself, what is Entity Types and Entity Sets, let me answer it quickly.
After all, SAPYard is here to hand hold you and help you cross this SAP Gateway learning curve smoothly. ☺

Entity Type is our very own **structure** (or a work area (holds just one row)). And you guessed it right, **Entity Set** is an **internal table** (holds more than one entity/rows).
ABAPers, are you gaining confidence? You just learned two new terms. ☺

Let us create our first structure; oops!! work area; oops Entity Type (right term in Gateway terminology).



Right click on Entity Types folder and select "Create", provide the name you like and do not forget to tick the checkbox "Create Related Entity Set". For our example, POHeader is the structure(work area) while POHeaderSet is our internal table.

Name	ABAP Structure Name	Base Type	Abs...	Label	L...	Semantics	Thi...	Me...	Author
POHeader									

Check the Service Implementation folder has POHeaderSet Operations auto generated. These are ABAP Methods which would be triggered when the relevant endpoints would be called.

Now we need to define the fields of the structure/work area and internal table.

SAP Gateway Service Builder

Properties

Name	Key	Edm Type	Prec.	Scale	Max...	Unit Prop.	Cre...	Up...	Sor...	Null...	Filt.	Label
Ebeln	<input type="checkbox"/>	Edm.String	0	0	10		<input type="checkbox"/>					
Bukrs	<input type="checkbox"/>	Edm.String	0	0	4		<input type="checkbox"/>					

Doble click on the Properties folder, hit the create icon and then start adding the field names and the type and length. In the example, we have added Ebeln and Bukrs. So on and so forth, we need to build our Entity Type.

Also Read: [SAP HANA from Space Level](#)

Doesn't it look too tiresome to add one field at a time in the Entity Type? There has to be a better way. The above steps are just to let you know such thing exists. *In real projects, you would like to use the below method.* Delete the POHeader Entity Type and POHeaderSet Entity Set by right clicking and hitting delete.

Let us create them again.

SAP Gateway Service Builder

Project Edit Goto Extras System Help

1

2

3

4

5

6

7

Wizard Step 1 of 3: Import from DDIC Structure

Create an Entity Type or Complex Type

Name: POHeader

Entity Type Complex Type

Import from ABAP Structure

ABAP Structure: EKKO

Create Default Entity Set

Next Cancel

Right click on the Data Model folder and Import the DDIC Structure. Give the Entity Type name and ABAP structure whose fields you want to import to your Entity Type. Do not forget to click the checkbox for Entity Set (unless you are sure you will not need an internal table).

Hit Next and choose the fields from EKKO (our example) structures which you want to add to your Entity Type.

Wizard Step 2 of 3: Import from DDIC Structure

Select Parameter(s)

Data Source Parameter	Assign Structure	Description	Type	Length	Decimals	Import Search Help	Search Help
• <input checked="" type="checkbox"/> EKKO	<input type="checkbox"/>	EKKO				<input type="checkbox"/>	
• <input type="checkbox"/> MANDT	<input type="checkbox"/>	Client	CLNT	3		<input type="checkbox"/>	H_T000
• <input type="checkbox"/> EBELN	<input type="checkbox"/>	Purchasing Doc.	CHAR	10		<input type="checkbox"/>	
• <input checked="" type="checkbox"/> BUKRS	<input type="checkbox"/>	Company Code	CHAR	4		<input type="checkbox"/>	
• <input type="checkbox"/> BSTYP	<input type="checkbox"/>	Doc. Category	CHAR	1		<input type="checkbox"/>	
• <input checked="" type="checkbox"/> BSART	<input type="checkbox"/>	Document Type	CHAR	4		<input type="checkbox"/>	H_T161
• <input type="checkbox"/> BSAKZ	<input type="checkbox"/>	Control	CHAR	1		<input type="checkbox"/>	
• <input checked="" type="checkbox"/> LOEKZ	<input type="checkbox"/>	Deletion Ind.	CHAR	1		<input type="checkbox"/>	
• <input checked="" type="checkbox"/> STATU	<input type="checkbox"/>	Status	CHAR	1		<input type="checkbox"/>	
• <input type="checkbox"/> AEDAT	<input type="checkbox"/>	Created on	DATS	8		<input type="checkbox"/>	
• <input checked="" type="checkbox"/> ERNAM	<input type="checkbox"/>	Created by	CHAR	12		<input type="checkbox"/>	
• <input type="checkbox"/> PINCR	<input type="checkbox"/>	Item Interval	NUMC	5		<input type="checkbox"/>	
• <input type="checkbox"/> LPONR	<input type="checkbox"/>	Last Item	NUMC	5		<input type="checkbox"/>	
• <input checked="" type="checkbox"/> LIFNR	<input type="checkbox"/>	Vendor	CHAR	10		<input type="checkbox"/>	
• <input type="checkbox"/> SPRAS	<input type="checkbox"/>	Language	LANG	1		<input type="checkbox"/>	
• <input checked="" type="checkbox"/> ZTERM	<input type="checkbox"/>	Payt Terms	CHAR	4		<input type="checkbox"/>	H_T002
• <input type="checkbox"/> ZBD1T	<input type="checkbox"/>	Payment in	DEC	3		<input type="checkbox"/>	
• <input type="checkbox"/> ZBD2T	<input type="checkbox"/>	Payment in	DEC	3		<input type="checkbox"/>	
• <input type="checkbox"/> ZBD2T	<input type="checkbox"/>	Payment in	DEC	2		<input type="checkbox"/>	

Back Next Cancel

Next, choose the Key for your structure/entity type and entity set. For our example it is Ebeln.

Wizard Step 3 of 3: Import from DDIC Structure

Modify Entity Type

IsE...	Complex/Entity Type	ABAP Name	Is Key	Type	Name	Label
<input checked="" type="checkbox"/>	POHeader	EBELN	<input checked="" type="checkbox"/>	CHAR	Ebeln	Purchasing Doc.
<input checked="" type="checkbox"/>	POHeader	BUKRS	<input type="checkbox"/>	CHAR	Bukrs	Company Code
<input checked="" type="checkbox"/>	POHeader	BSTYP	<input type="checkbox"/>	CHAR	Bstyp	Doc. Category
<input checked="" type="checkbox"/>	POHeader	BSART	<input type="checkbox"/>	CHAR	Bsart	Document Type
<input checked="" type="checkbox"/>	POHeader	LOEKZ	<input type="checkbox"/>	CHAR	Loekz	Deletion Ind.
<input checked="" type="checkbox"/>	POHeader	STATU	<input type="checkbox"/>	CHAR	Statu	Status
<input checked="" type="checkbox"/>	POHeader	ERNAM	<input type="checkbox"/>	CHAR	Ernam	Created by
<input checked="" type="checkbox"/>	POHeader	LIFNR	<input type="checkbox"/>	CHAR	Lifnr	Vendor
<input checked="" type="checkbox"/>	POHeader	ZTERM	<input type="checkbox"/>	CHAR	Zterm	Payt Terms

Back Finish Cancel

After you hit finish hit the save button and check the Properties and Service Implementation folder. They would have the *structure and operations* respectively.

SAP Gateway Service Builder

The screenshot shows the SAP Gateway Service Builder interface. On the left, the project tree shows a 'Data Model' node expanded, with 'Entity Types' selected. Under 'Entity Types', 'POHeader' is selected. The main area displays the 'Entity Types' table:

Name	ABAP Structure Name	Base Type	Abs...	Label	L...	Semantics	Thi...	Me...	Author	ETag	Published
POHeader	EKKO		<input type="checkbox"/>		T						

Below the table, two sections are highlighted with red boxes:

- Properties:** A list of fields: Ebeln, Ebelp, Loekz, Statu, Txz01, Matnr, Werks, Lgort, Ktrmg, Menge.
- Operations:** A list of operations: Create, Delete, GetEntity (Read), GetEntitySet (Query), Update.

Structure and **Operations** are also labeled in large red text next to their respective sections.

At this point, let us import one more Entity Type called POItem from EKPO using the steps just mentioned above.

SAP Gateway Service Builder

The screenshot shows the SAP Gateway Service Builder interface after importing the POItem entity type. The project tree now includes 'POItem' under the 'Data Model' node. The main area displays the 'Entity Types' table:

Name	ABAP Structure Name	Base Type	Abs...	Label	L...	Semantics	Thi...	Me...	Author	ETag	Published	Title
POHeader	EKKO		<input type="checkbox"/>		T							
POItem	EKPO		<input type="checkbox"/>		T							

Below the table, two sections are highlighted with red boxes:

- Properties:** A list of fields: Ebeln, Ebelp, Loekz, Statu, Txz01, Matnr, Werks, Lgort, Ktrmg, Menge.
- Operations:** A list of operations: Create, Delete, GetEntity (Read), GetEntitySet (Query), Update.

Structure and **Operations** are also labeled in large red text next to their respective sections.

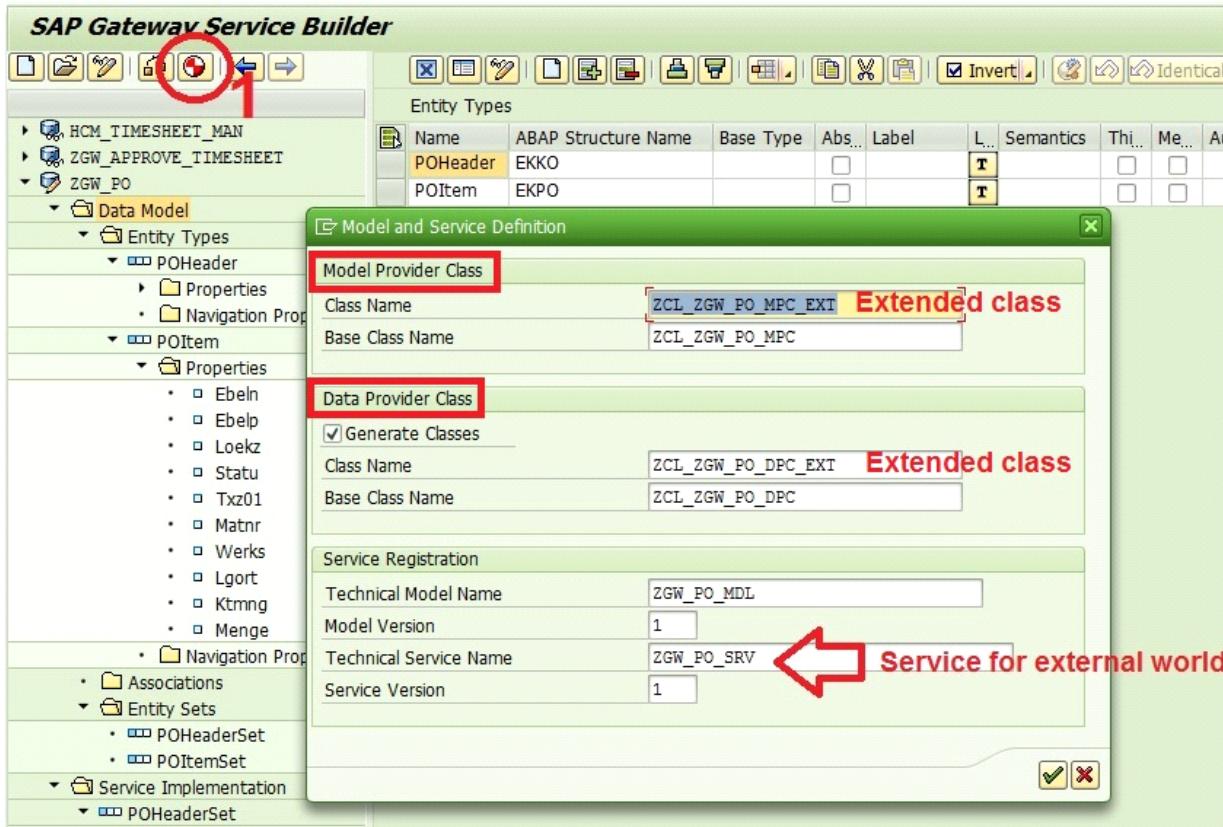
From the above image, it is clear that every Entity Types will have its own **Properties**

and **Navigation Properties** Folder. And every Service Implementation Entity Set will have its own **Operations** (Create, GetEntity, Update, Delete etc).

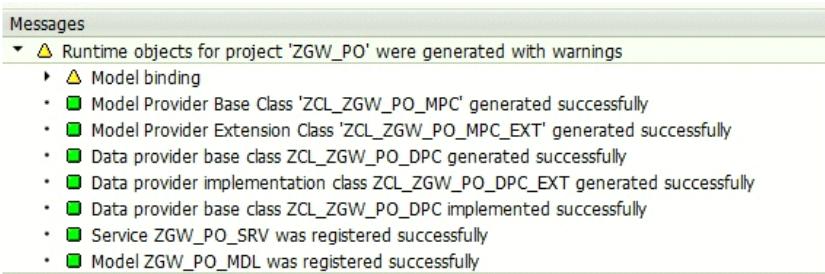
As of now, we have a Header Table and an Item/Detail Table. Let us assume we do not need any more Entity Types.

B. Implement/Register the Service

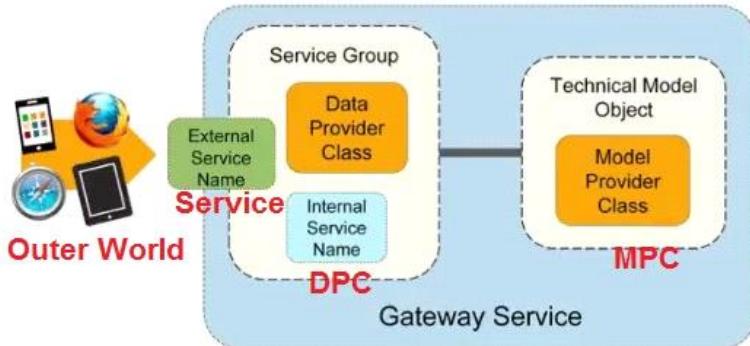
Let us go ahead and generate and register our service.



Hit the **Generate** icon and hit ok. Provide the package and transport number or save it as local. You should get the success message as shown below.



Please note the **Technical Service Name is the actual Service which the external system needs to call**. Two classes, **Model Provider Class (MPC)** and **Data Provider Class (DPC)** are also generated along with Base and Extended Class.



DPC and MPC are connected by Configuration (Not Coding)

Model Provider Class inherits from /IWBEPECL_MGW_ABS_MODEL and Data Provider Class inherits from /IWBEPECL_MGW_ABS_DATA. The below image shows the relationship between the generated classes and their superclasses (parents).

SAP Gateway Service Builder

Name	Generated Artifact Type	Program ID	Object Type
ZCL_ZGW_PO_DPC	Data Provider Base Class	R3TR	CLAS
ZCL_ZGW_PO_DPC_EXT	Data Provider Extension Class	R3TR	CLAS
ZCL_ZGW_PO_MPC	Model Provider Base Class	R3TR	CLAS
ZCL_ZGW_PO_MPC_EXT	Model Provider Extension Class	R3TR	CLAS
ZGW_PO_MDL	Registered Model	R3TR	IWMO
ZGW_PO_SRV	Registered Service	R3TR	IWSV

The screenshot shows the SAP Gateway Service Builder interface. On the left, the navigation tree highlights the 'Runtime Artifacts' section under 'ZGW_PO'. This section contains several artifacts, with the 'Data Provider Class' (ZCL_ZGW_PO_DPC) and 'Model Provider Class' (ZCL_ZGW_PO_MPC) being the most prominent. These two classes are highlighted with red boxes. Below them, the 'Model Provider Extension Class' (ZCL_ZGW_PO_MPC_EXT) and the 'Data Provider Extension Class' (ZCL_ZGW_PO_DPC_EXT) are also highlighted with red boxes. Red arrows point from the 'Superclass' fields of the ZCL_ZGW_PO_DPC and ZCL_ZGW_PO_MPC screens to the respective /IWBEPECL_MGW_ABS_DATA and /IWBEPECL_MGW_PUSH_ABS_MODEL screens. Blue arrows point from the 'Superclass' fields of the ZCL_ZGW_PO_MPC and ZCL_ZGW_PO_DPC screens to the /IWBEPECL_MGW_PUSH_ABS_MODEL screen.

FYI: Data Provider Class **provides** the **Gateway Service functionalities**. Model Provider Class **defines** the **Gateway Service interface**. DPC and MPC are not connected by any coding. They talk to each other via Configuration.

C. Add Service to Service Catalog (Register the Service to Gateway Hub)

We have implemented the Service, now we need to add the service to the **Service Catalog**.

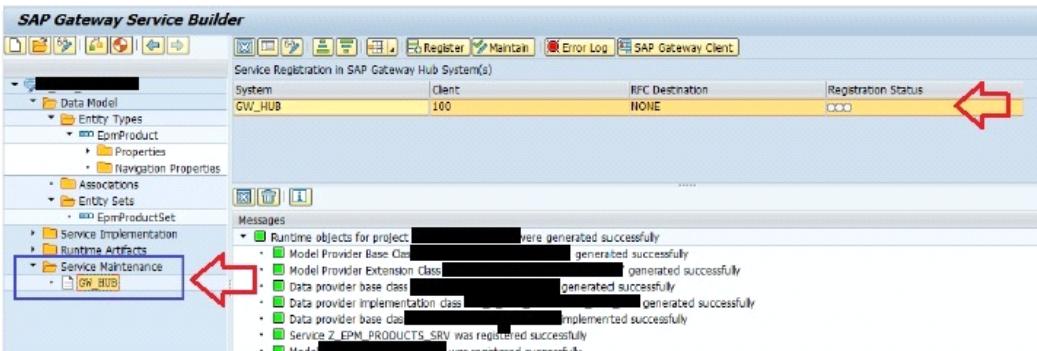


Image from Embedded Deployment

Most of the tutorial present on the internet shows that the *Service Maintenance folder* would have an entry with *System details* and the *Register/Maintain* buttons are active (as in above image). If you see an entry in Service Maintenance folder, it means your system has "**Embedded Deployment**".

SAP Netweaver gateway components are installed as *add-ons* on the *SAP backend system* itself in the *Embedded Deployment* option. Both the **OData modeling** and the **exposing of the Services** to the outer world is done from the *backend system*.

The advantage of Embedded Deployment Strategy:

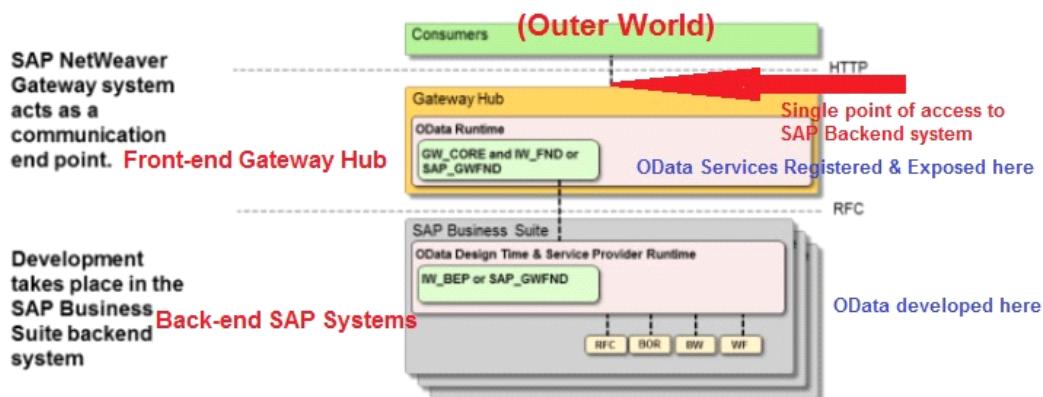
- Saves cost as there is no need of any dedicated Gateway Hub
- It is faster as the runtime overhead due to remote calls in case of two separate systems (separate front end and back end) is avoided

Disadvantages of Embedded Deployment Strategy:

- There is no single point connection to the backend via Gateway Hub; instead, back end system is directly exposed to the outer world (consumers), so better security measures need to be in place
- Back end system upgrade would follow a different cycle than Netweaver Gateway
- If the landscape has multiple SAP Business Suite backend, then each system needs to have its own installation and configuration of Netweaver Gateway components.

If you have Embedded System, hit the Register and Maintain buttons and you are good.

The above image is not from our sap system. Our system does not show any entry in Service Maintenance folder. Our architecture is "**Hub Deployment with Development in the Backend System**". In this strategy, the *SAP Netweaver Gateway* is *installed on separate SAP machine* called as *Gateway Hub*. **Gateway Hub** would be the window for the outer world to SAP. **Gateway Hub** would register and expose the OData Services, but the development would still happen in the SAP backend system.



Hub Deployment with Development in the Backend System Strategy

The advantage of Hub Deployment with Development in the Backend System Strategy:

- i. Better security as it has only one point of access to SAP Backend system for outer space via Gateway Hub
- ii. Although Gateway components are in different SAP system, it still has direct access to Backend DDIC and business data for OData Model/Services
- iii. Gateway Hub can be in advanced newer version NW 7.3/7.4 or up while Backend still can be in the lower release. Gateway on new release would support SAPUI5.

Disadvantage: More cost and maintenance effort along with the runtime overhead due to remote calls.

The top screenshot shows the SAP Gateway Service Builder interface. The left sidebar lists services: HCM_TIMESHEET_MAN, ZGW_APPROVE_TIMESHEET, ZGW_PO (with sub-options Data Model, Service Implementation, Runtime Artifacts, and Service Maintenance), ZGW_PURCHASE, and ZGW_USER_PROJ. The main area displays a message: "Either GW is not enabled, or your system has different front-end and back-end systems". The "Register" and "Maintain" buttons in the toolbar are circled in red.

The bottom screenshot shows the "Activate and Maintain Services" screen. It includes tabs for "Service Catalog", "ICF Nodes", and "System Aliases". The "Service Catalog" table lists BEP ZGW_PURCHASE_SRV (1 Purchase Order App) and BEP ZHCM_TIMESHEET_APPROVE_SRV_1 (1 Approve Timesheet Extension). The "ICF Nodes" table shows a single entry for ODATA. The "System Aliases" table shows a entry for UD1_130 (UD1 Backend). The SAP status bar at the bottom shows "SAP > UD1 (1) 130". A red circle highlights the SAP icon in the status bar.

Hub Deployment with Development in the Backend System

Coming back to our screen. We are in our Backend system, therefore by default, we do not get the option to maintain and register our service in our backend system. We need to go to our front end system where we have the Netweaver Gateway i.e. our Gateway Hub.

Go to your **Gateway Hub (Front-end system)** and execute t-code /n/IWFND/MAINT_SERVICE. Save this t-code in your favorite list or remember it by heart. ☺ You will need it for all your OData services.

Activate and Maintain Services

The screenshot shows the SAP Activate and Maintain Services interface. At the top, there is a toolbar with various icons and buttons, including 'Add Service' which is highlighted with a red circle. Below the toolbar, there are several status links: Request Statistics, Refresh Catalog, OAuth, Soft State, and Processing Mode.

Service Catalog

Type	Technical Service Name	V...	Service Description	External Service Name	Nsp.	O
BEP	C_DRAFTLIFECYCLEADMIN_CDS	1	Draft Lifecycle Administrative Data	C_DRAFTLIFECYCLEADMIN_CDS	/IWFND/	
	IWFND/SG_MED_CATALOG	1	Catalog Service	CATALOGSERVICE	/IWFND/	
	IWFND/SG_MED_CATALOG	2	Catalog Service Version 2	CATALOGSERVICE	/IWFND/	
BEP	DAAG_DTG_SRV	1	Odata for DTG app	DAAG_DTG_SRV		

ICF Nodes

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

System Aliases

SAP System Alias	Description	Defau

Our service is still not added to the **Service Catalog**. After we **add the service to the Service Catalog**, only then is **our service available for the outer world to access**.

Add Selected Services

Get Services 2

Filter

System Alias	UD1_130
Technical Service Name	
External Service Name	ZGW_PO_SRV 

 Add Selected Services

Select Backend Services

Type	Technical Service Name	Service Description	External Service Name
BEP	ZGW_PO_SRV	1 Service for PO	ZGW_PO_SRV

 3. Click here

Add Service

Service

Technical Service Name	ZGW_PO_SRV
Service Version	1
Description	Service for PO
External Service Name	ZGW_PO_SRV
Namespace	
External Mapping ID	
External Data Source Type	C

Model

Technical Model Name	ZGW_PO_MDL 
Model Version	1

Creation Information

Package Assignment	<input type="button" value="Local Object"/>
--------------------	---

ICF Node

<input checked="" type="radio"/> Standard Mode	<input type="radio"/> None
<input checked="" type="checkbox"/> Set Current Client as Default Client in ICF Node	

OAuth enablement

<input type="checkbox"/> Enable OAuth for Service



Hit the Add Service button, provide your backend system alias and external service name (our case ZGW_PO_SRV). You will get the service you created in the backend. Click on it and it would show the Service (technical/external) name along with the Technical name of the Model (ZGW_PO_MDL) and hit save. Go back to the Service Catalog screen.

Activate and Maintain Services

Type	Technical Service Name	V...	Service Description	External Service Name	Nsp.	OAu...	Soft State	Processing Mode
BEP	REPM_SALESCORDERDETAILSO_CDS	1	REPM: Sales Order Details Query View	REPM_SALESCORDERDETAILSO_CDS		<input type="checkbox"/>	Not Supported	Routing-based
BEP	REPM_SALESCORDERPRODAMTQUERY_CDS	1	REPM : Sales Order Product Query View	REPM_SALESCORDERPRODAMTQUERY_CDS		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZS_EPM_SADL_GW_DEV_SCEN_RO_SRV	1	EPM: SADL-based GW-Service 'Developer Scenario'	S_EPM_SADL_GW_DEV_SCEN_RO_SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	SADL_V_EXP_QUERY_CDS	1	SADL Generated Service	SADL_V_EXP_QUERY_CDS		<input type="checkbox"/>	Not Supported	Routing-based
BEP	SADI_V_SALESCORDER_BO_CDS	1	SADI Generated Service	SADI_V_SALESCORDER_BO_CDS		<input type="checkbox"/>	Not Supported	Routing-based
BEP	SEPM_HANA_EXT_PAL_ODATA_SRV	1	SEPM_HANA_EXT_PAL_ODATA	SEPM_HANA_EXT_PAL_ODATA_SRV		<input checked="" type="checkbox"/>	Not Supported	Routing-based
BEP	/IWFND/SUBSCRIPTIONMANAGEMENT	2	MOC enabled Subscription Management Service	SUBSCRIPTIONMANAGEMENT	/IWBEPEP/	<input type="checkbox"/>	Routing-based	
BEP	ZTRANSPORT	1	UI2: Transport Service	TRANSPORT	/UI2/	<input type="checkbox"/>	Routing-based	
	/IWFND/USAGEEXTRACTOR	1	Metering Usage Extractor	USAGEEXTRACTOR	/IWFND/	<input type="checkbox"/>	Routing-based	
	/IWFND/SG_USER_SERVICE	1	Information Worker - User Service	USERSERVICE	/IWFND/	<input type="checkbox"/>	Routing-based	
BEP	ZGW_APPROVE_TIMESHEET_SRV	1	Approve Timesheet Extension	ZGW_APPROVE_TIMESHEET_SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZGW_PO_SRV	1 Click	Service for PO	ZGW_PO_SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZGW_PURCHASE_SRV	1	Purchase Order App	ZGW_PURCHASE_SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZHOM_TIMESHEET_APPROVE_SRV	1	Approve Timesheet Extension	ZHOM_TIMESHEET_APPROVE_SRV		<input type="checkbox"/>	Not Supported	Routing-based

ICF Node Call Browser SAP Gateway Client 3

ICF Nodes

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

System Aliases 2 This shows up

SAP System Alias	Description	Default	
UD1_130	UD1 Backend	<input checked="" type="checkbox"/>	

Find your service, click on it. It would show up in System Aliases window at the lower right corner. Let us test it using **SAP Gateway Client**. Or use t-code /IWFND/GW_CLIENT (remember this t-code as well). You can also test by Call Browser option. For now, we will use SAP Gateway Client option.

SAP Gateway Client

Execute Select Service Administration Service Implementation EntitySets Add URI Option

HTTP Method: GET Request URI: /sap/opu/odata/sap/ZGW_PO_SRV/?\$format=xml Reuse HTTP Connection (e.g. for Soft State)

Protocol: HTTP Test Group

HTTP Request

Header Name Value

HTTP Response - Processing Time = 292 ms

Header Name Value

-status_code 403
 -status_reason Forbidden
 content-type application/xml; charset=utf-8
 ~server_protocol HTTP/1.0
 set-cookie sap-usercontext=sap-language=E&sap-client=130; path=/

```
<?xml version="1.0" encoding="utf-8" ?>
- <error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <code>/IWFND/CM_CONSUMER/101</code>
  <message xml:lang="en">No authorization to access Service 'ZGW_PO_SRV_0001'</message>
- <innererror>
  - <application>
    <component_id />
    <service_namespace>/SAP/</service_namespace>
    <service_id>ZGW_PO_SRV</service_id>
    <service_version>0001</service_version>
  </application>
  <transactionid>585FE63A574B0350E10080000A8C7D76</transactionid>
  <timestamp>20161230205654.6498930</timestamp>
- <Error_Resolution>
  <SAP_Transaction>Run transaction /IWFND/ERROR_LOG on SAP Gateway hub system (System Alias ) and search for entries with the timestamp above for more details</SAP_Transaction>
```

OOPS!! You get this error. Status 403, Forbidden with the message "No authorization to access Service". Let us check the Authorization Object issue in t-code **SU53** and ask your SAP Security Team to provide you access to this authorization.

•	•	Authorization check failed
•	Date 12/30/2016 Time 15:15:50 SAP Gateway: Service Groups Metadata ZGW_PO_SRV_0001	
•	Authorization Obj. S_SERVICE Check at Start of External Services	
• Authorization Field SRV_NAME Program, transaction or function module name		E3D470220DEDD730178D34
• Authorization Field SRV_TYPE Type of Check Flag and Authorization Default Values		HT
• Date 12/30/2016 Time 15:03:34 SAP Gateway: Service Groups Metadata ZGW_PO_SRV_0001		
• Date 12/30/2016 Time 14:56:54 SAP Gateway: Service Groups Metadata ZGW_PO_SRV_0001		
• Date 12/30/2016 Time 13:48:06 Transaction /IWFND/MAINT_SERVICE		
• Authorization Obj. S_CTS_SADM System-Specific Administration (Transport)		
• Authorization Field CTS_ADMINCT Administration Tasks for Change and Transport System		TABL
• Authorization Field DESTSYS Logical System		EGD
• Authorization Field DOMAIN TMS: Transport Domain		DOMAIN_EGD
• Date 12/30/2016 Time 13:48:06 Transaction /IWFND/MAINT_SERVICE		
• Authorization Obj. S_CTS_ADMIN Administration Functions in Change and Transport System		
• Authorization Field CTS_ADMINCT Administration Tasks for Change and Transport System		TABL
• Date 12/30/2016 Time 13:48:04 Transaction /IWFND/MAINT_SERVICE		
• Date 12/30/2016 Time 13:48:04 Transaction /IWFND/MAINT_SERVICE		
• Date 12/30/2016 Time 13:48:02 Transaction /IWFND/MAINT_SERVICE		
• Date 12/30/2016 Time 13:48:02 Transaction /IWFND/MAINT_SERVICE		
• Date 12/30/2016 Time 13:48:02 Transaction /IWFND/MAINT_SERVICE		
• Date 12/30/2016 Time 13:48:02 Transaction /IWFND/MAINT_SERVICE		
• User's Authorization Data 00951730		
• Authorization Object S_SERVICE Check at Start of External Services		
• Authorization Object S_CTS_ADMIN Administration Functions in Change and Transport System		

Just in case your security team asks what has to be done. The below screenshot might help him.

Display Role: Authorizations

Open Changed Maintained Organizational levels... Information Trace Versions

Maint. 0 unmaint. org. levels, 0 open fields, Status: Unchanged

BC: [REDACTED] SRVC OOC Temp Role- Do not Transport

- > OOC Manually Cross-application Authorization Objects
 - > OOC Manually Administration for Internet Communication Framework
 - > OOC Manually Check at Start of External Services
 - > OOC Manually Check at Start of External Services
 - > Program, transaction or functi [REDACTED] 0BE9A829CF9DB7B9154650EE39BCA4, A404F50234C726F8641D183BA5E7D0 <...>
 - > Type of Check Flag and Authori Hash Value for TADIR Object
- > OOC Manually Basis: Administratio A

Define Values

Type	TADIR Service		
[REDACTED]			
[REDACTED]			
[REDACTED]			
Name	Prog. ID	Obj.	Object Name
[REDACTED]	R3TR	IWSG	[REDACTED]_DEV_SCEN_RO_SRV_0001
[REDACTED]	R3TR	IWSG	[REDACTED]_SRV_0001
E3D470220DEDD730178D34146F2E...	R3TR	IWSG	ZGW_PO_SRV_0001

Where did the ZGW_PO_SRV_0001 come from? The below screen might help.

BEP	ZTRANSPORT	1	UI2: Transport Service	TRANSPORT	/UI2/	<input type="checkbox"/>	Routing-based
	/IWFND/USAGEEXTRACTOR	1	Metering Usage Extractor	USAGEEXTRACTOR	/IWFND/	<input type="checkbox"/>	Routing-based
	/IWFND/SG_USER_SERVICE	1	Information Worker - User Service	USERSERVICE	/IWFND/	<input type="checkbox"/>	Routing-based
BEP	ZGW_APPROVE_TIMESHEET_SRV	1	Approve Timesheet Extension	ZGW APPROVE_TIMESHEET_SRV		<input type="checkbox"/> Not Supported	Routing-based
BEP	ZGW_PO_SRV	1	Service for PO	ZGW PO_SRV		<input type="checkbox"/> Not Supported	Routing-based
BEP	ZGW_PURCHASE_SRV	1	Purchase Order App	ZGW PURCHASE_SRV		<input type="checkbox"/> Not Supported	Routing-based
BEP	ZHCM_TIMESHEET_APPROVE_SRV_1	1	Approve Timesheet Extension	ZHCM_TIMESHEET_APPROVE_SRV		<input type="checkbox"/> Not Supported	Routing-based

ICF Node Call Browser SAP Gateway Client

ICF Nodes

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

System Aliases

SAP System Alias	Description	Default	Metadata	User Role
UD1_130	UD1 Backend	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Display View "Assign SAP System Aliases to OData Service": Overview



Assign SAP System Aliases to OData Service

Service Doc. Identifier	User Role	Host Name	SAP System Alias	Default System	Metadata Default	Tech. Svc. Name	Ext. Service Nam
ZGW_PO_SRV_0001	<input type="checkbox"/> 2		UD1_130	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ZGW_PO_SRV	ZGW_PO_SRV

Also Check: [SAP ABAP for HANA Tutorials](#)

Assuming your authorization is in place. Let us try t-code

/n/IWFND/MAINT_SERVICE, select the service and hit SAP Gateway Client or directly use t-code /n/IWFND/GW_CLIENT. Hit the Execute button. This time the status code is 200 i.e success.

SAP Gateway Client

Execute Select Service Administration Service Implementation EntitySets Add URI Option

HTTP Method GET POST PUT PATCH MERGE DELETE Reuse HTTP Connection (e.g. for Soft State)

Request URI /sap/opu/odata/sap/ZGW_PO_SRV/?\$format=xml Multiple Rows

Protocol HTTP HTTPS Test Group Test Case

HTTP Request

Header Name	Value
Header Name	Value

HTTP Response - Processing Time = 137 ms

Header Name	Value
~status_code	200
status_reason	OK
sap-processing-time	Microhub=,crp=,st=,MedCacheHub=SHM,codeployed=,softstate=
last-modified	Thu, 29 Dec 2016 23:07:55 GMT

```

<?xml version="1.0" encoding="utf-8" ?>
- <app:service xml:lang="en"
  xmlns:base="http://txaixegd01[REDACTED]/sap/opu/odata/sap/ZGW_PO_SRV/"
  xmlns:app="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData">
- <app:workspace>
  <atom:title type="text">Data</atom:title>
- <app:collection sap:createable="false" sap:updatable="false" sap:deletable="false"
  sap:pageable="false" sap:content-version="1" href="POHeaderSet">
    <atom:title type="text">POHeaderSet</atom:title>
    <sap:member-title>POHeader</sap:member-title>
  </app:collection>
- <app:collection sap:createable="false" sap:updatable="false" sap:deletable="false"
  sap:pageable="false" sap:content-version="1" href="POItemSet">
    <atom:title type="text">POItemSet</atom:title>
    <sap:member-title>POItem</sap:member-title>
  </app:collection>
</app:workspace>
<atom:link rel="self"
  href="http://txaixegd01[REDACTED]/sap/opu/odata/sap/ZGW_PO_SRV/" />
<atom:link rel="latest-version"
  href="http://txaixegd01[REDACTED]/sap/opu/odata/sap/ZGW_PO_SRV/" />
</app:service>
```

Congratulations!! You just created, maintained and activated your first OData Service.

In the above image, our URI (/sap/opu/odata/sap/ZGW_PO_SRV/?\$format=xml) has format XML. Let us change it to JSON (/sap/opu/odata/sap/ZGW_PO_SRV/?\$format=json). Both xml and json informs that they have two Entity Sets.

SAP Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/sap/ZGW_PO_SRV/?\$format=json

Protocol: HTTP Test Group: Test Case

HTTP Request

Header Name	Value
~status_code	200
~status_reason	OK
sap-processing-info	microhub=,crp=,st=,MedCacheHub=SHM,codeployed=,softstate=
last-modified	Thu, 29 Dec 2016 23:07:55 GMT

HTTP Response - Processing Time = 132 ms

```

1
2   "d" : {
3     "EntitySets" : [
4       "POHeaderSet",
5       "POItemSet"
6     ]
7   }
8

```

Please note: The word json is case sensitive. Give lower case "json".

Now let us check the metadata of the OData. In layman's term metadata means the structure and properties or full skeleton (X-Ray view). ☺

SAP Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/sap/ZGW_PO_SRV/\$metadata

Protocol: HTTP Test Group: Test Case

HTTP Request

Header Name	Value
~status_code	200

HTTP Response - Processing Time = 288 ms

```

<?xml version="1.0" encoding="utf-8" ?>
- <edmx:Edmx Version="1.0" xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData">
- <edmx:DataServices m:DataServiceVersion="2.0">
- <Schema Namespace="ZGW_PO_SRV" xml:lang="en" sap:schema-version="1"
  sap:uri="http://txaixegd01.*****.sap.basis.tra.svc/sap/opu/odata/sap/ZGW_PO_SRV/>
+ <EntityType Name="POHeader" sap:content-version="1">
+ <EntityType Name="POItem" sap:content-version="1">
- <EntityContainer Name="ZGW_PO_SRV_Entities" m:isDefaultEntityContainer="true" sap:supported-
  formats="atom json vxml">
  1
  <EntitySet Name="POHeaderSet" EntityType="ZGW_PO_SRV.POHeader" sap:createable="false"
    sap:updatable="false" sap:deletable="false" sap:pageable="false" sap:content-version="1" />
  <EntitySet Name="POItemSet" EntityType="ZGW_PO_SRV.POItem" sap:createable="false"
    sap:updatable="false" sap:deletable="false" sap:pageable="false" sap:content-version="1" />
  <EntityContainer>
    <atom:link rel="self"
      2
      href="http://txaixegd01.*****.sap.basis.tra.svc/sap/opu/odata/sap/ZGW_PO_SRV/
      $metadata" xmlns:atom="http://www.w3.org/2005/Atom" />
    <atom:link rel="latest-version"
      3
      href="http://txaixegd01.*****.sap.basis.tra.svc/sap/opu/odata/sap/ZGW_PO_SRV/
      latest" xmlns:atom="http://www.w3.org/2005/Atom" />
  </EntityContainer>

```

It shows the Entity Type (work area/structure) Names. In our OData, we have two (POHeader and POItem). It also shows the Entity Set (Internal tables) Names which are also two for our case (POHeaderSet and POItemSet). At the bottom of the page, you would see a link. That is the link which opens your service in Web Browser. Copy the link and paste it in your browser (you need to give your SAP Front End credentials).

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData" Version="1.0">
  <edmx:DataServices m:DataServiceVersion="2.0">
    <Schema xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="ZGW_PO_SRV" xml:lang="en" sap:schema-version="1">
      <EntityType Name="POHeader" sap:content-version="1">
        <Key>
          <PropertyRef Name="Ebeln"/>
        </Key>
        <Property Name="Ebeln" Type="Edm.String" Nullable="false" MaxLength="10" sap:label="Purchasing Doc." sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Bukrs" Type="Edm.String" Nullable="false" MaxLength="4" sap:label="Company Code" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Bstyp" Type="Edm.String" Nullable="false" MaxLength="1" sap:label="Doc. Category" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Bsart" Type="Edm.String" Nullable="false" MaxLength="4" sap:label="Document Type" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Loekz" Type="Edm.String" Nullable="false" MaxLength="1" sap:label="Deletion Ind." sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Statu" Type="Edm.String" Nullable="false" MaxLength="1" sap:label="Status" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Ernam" Type="Edm.String" Nullable="false" MaxLength="12" sap:label="Created by" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Lifnr" Type="Edm.String" Nullable="false" MaxLength="10" sap:label="Vendor" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Zterm" Type="Edm.String" Nullable="false" MaxLength="4" sap:label="Payt Terms" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
      </EntityType>
      <EntityType Name="POItem" sap:content-version="1">
        <Key>
          <PropertyRef Name="Ebeln"/>
          <PropertyRef Name="Ehelp"/>
        </Key>
        <Property Name="Ebeln" Type="Edm.String" Nullable="false" MaxLength="10" sap:label="Purchasing Doc." sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Ehelp" Type="Edm.String" Nullable="false" MaxLength="5" sap:label="Item" sap:creatable="false" sap:updatable="false" sap:sortable="false" sap:filterable="false"
          sap:filterable="false"/>
        <Property Name="Loekz" Type="Edm.String" Nullable="false" MaxLength="1" sap:label="Deletion Ind." sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Statu" Type="Edm.String" Nullable="false" MaxLength="1" sap:label="RFQ status" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
        <Property Name="Txz01" Type="Edm.String" Nullable="false" MaxLength="40" sap:label="Short Text" sap:creatable="false" sap:updatable="false" sap:sortable="false"
          sap:filterable="false"/>
      </EntityType>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

If you were a non-sap web developer who would need to consume this SAP OData service and you just have the service name and URI. You would be interested in knowing the details of the structures so that you can make use of it in your non-sap application. The first thing you would do is use **?\$format** to check the details.

Below is two popular way to check the format.

1. ?\$format = json

[http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/?\\$format=json](http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/?$format=json)

2. ?\$format=xml

[http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/?\\$format=xml](http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/?$format=xml)

```
{"d":{"EntitySets":["POHeaderSet","POItemSet"]}}
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

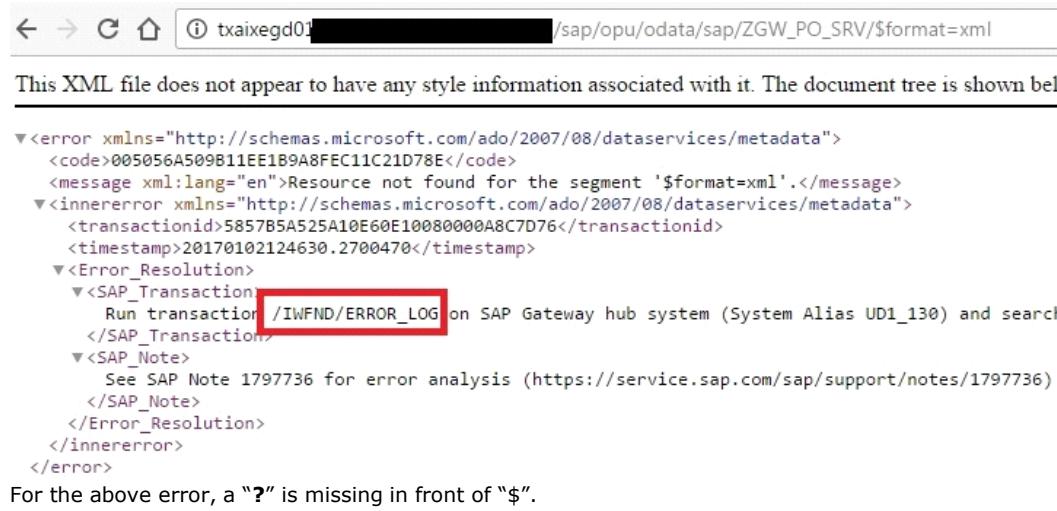
```

<app:service xmlns:app="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData" xml:lang="en" xml:base="http://txaixegd01 [REDACTED]/sap/opu/odata/sap/ZGW_PO_SRV/">
  <app:workspace>
    <atom:title type="text">Data</atom:title>
    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:pageable="false" sap:content-version="1" href="POHeaderSet">
      <atom:title type="text">POHeaderSet</atom:title>
      <app:member><atom:title>POHeader</atom:title></app:member>
    </app:collection>
    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:pageable="false" sap:content-version="1" href="POItemSet">
      <atom:title type="text">POItemSet</atom:title>
      <app:member><atom:title>POItem</atom:title></app:member>
    </app:collection>
  </app:workspace>
  <atom:link rel="self" href="http://txaixegd01 [REDACTED]/sap/opu/odata/sap/ZGW_PO_SRV/">
  <atom:link rel="latest-version" href="http://txaixegd01 [REDACTED]/sap/opu/odata/sap/ZGW_PO_SRV/">
</app:service>
```

How can ABAPers find the error of OData service if any?

In case you enter something incorrect in the URI or your URI is not working, you will get the message "Run transaction **/IWFND/ERROR_LOG** on SAP Gateway hub system" to find the issue.

T-code /n/IWFND/ERROR_LOG would be an ABAPers friend for stormy weather. Check the message for the URI.



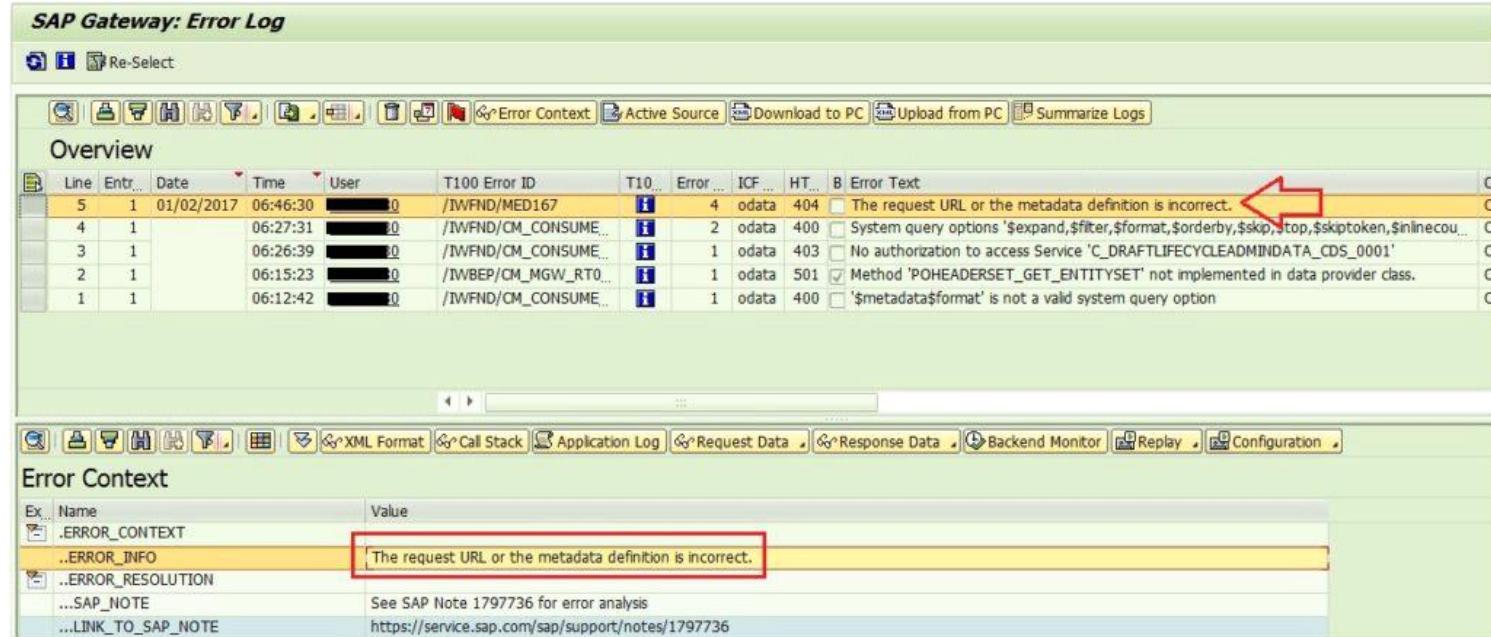
```
<error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <code>005056A509B11EE1B9A8FEC11C21D78E</code>
  <message xml:lang="en">Resource not found for the segment '$format=xml'.</message>
  <innererror xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
    <transactionid>5857B5A525A10F60E10080000A8C7D76</transactionid>
    <timestampl>20170102124630.2700470</timestampl>
    <Error_Resolution>
      <SAP_Transaction>
        Run transaction /IWFND/ERROR_LOG on SAP Gateway hub system (System Alias UD1_130) and search
      </SAP_Transaction>
      <SAP_Note>
        See SAP Note 1797736 for error analysis (https://service.sap.com/sap/support/notes/1797736)
      </SAP_Note>
    </Error_Resolution>
  </innererror>
</error>
```

For the above error, a "?" is missing in front of "\$".

wrong: [http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/\\$format=xml/](http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/$format=xml/)

right: [http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/?\\$format=xml](http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/?$format=xml)

Check the /n/IWFND/ERROR_LOG screen.



SAP Gateway: Error Log

Re-Select

Overview

Line	Entr...	Date	Time	User	T100 Error ID	T10...	Error ...	ICF ...	HT...	B	Error Text
5	1	01/02/2017	06:46:30	[REDACTED]	/IWFND/MED167	[REDACTED]	4	odata	404	<input type="checkbox"/>	The request URL or the metadata definition is incorrect.
4	1		06:27:31	[REDACTED]	/IWFND/CM_CONSUME...	[REDACTED]	2	odata	400	<input type="checkbox"/>	System query options '\$expand,\$filter,\$format,\$orderby,\$skip,\$top,\$skiptoken,\$inlinecou...
3	1		06:26:39	[REDACTED]	/IWFND/CM_CONSUME...	[REDACTED]	1	odata	403	<input type="checkbox"/>	No authorization to access Service 'C_DRAFTLIFECYCLEADMINDATA_CDS_0001'
2	1		06:15:23	[REDACTED]	/IWBEPM/CM_MGW_RT0...	[REDACTED]	1	odata	501	<input checked="" type="checkbox"/>	Method 'POHEADERSET_GET_ENTITYSET' not implemented in data provider class.
1	1		06:12:42	[REDACTED]	/IWFND/CM_CONSUME...	[REDACTED]	1	odata	400	<input type="checkbox"/>	'\$metadata\$format' is not a valid system query option

Error Context

Ex... Name	Value
.ERROR_CONTEXT	The request URL or the metadata definition is incorrect.
.ERROR_INFO	The request URL or the metadata definition is incorrect.
.ERROR_RESOLUTION	See SAP Note 1797736 for error analysis
...SAP_NOTE	See SAP Note 1797736 for error analysis
...LINK_TO_SAP_NOTE	https://service.sap.com/sap/support/notes/1797736

Now let us check if we can pull some data for POHeaderSet.

Enter URI

http://txaixegd01.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet in your browser or /sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet in your SAP t-code /IWFND/GW_CLIENT.

SAP Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet Protocol: HTTP

Test Group: Test Case Response in Browser: **HTTP Response - Processing Time = 436 ms**

Header Name Value
~status_code 501
~status_reason Not Implemented

content-type application/xml; charset=utf-8

HTTP Request

Header Name Value

```
<?xml version="1.0" encoding="utf-8" ?>
- <error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<code>/IWBEPCM_MGW_RT/021</code>
<message xml:lang="en">Method 'POHEADERSET_GET_ENTITYSET' not implemented in data provider class.</message>
- <innererror>
- <application>
<component_id>CA</component_id>
<service_namespace>/SAP/</service_namespace>
<service_id>ZGW_PO_SRV</service_id>
<service_version>0001</service_version>
</application>
<transactionid>5857B3C325A10E60E10080000A8C7D76</transactionid>
<timestamp>20170102125548.5592110</timestamp>
- <Error_Resolution>
<SAP_Transaction>Run transaction /IWFND/ERROR_LOG on SAP Gateway hub system (System Alias UD1_130) and search for entries with the timestamp above for more
```

OOPS, we get the error, *Not Implemented. Method 'POHEADERSET_GET_ENTITYSET' not implemented in data provider class.*

SAP Gateway Service Builder

Project: ZGW_PO Service for PO

1. Right Click on GetEntitySet (Query)

Information: Operation POHEADERSET_GET_ENTITYSET has not yet been implemented

In our backend system, in t-code SEGW, we created the Entity Types/Set and activated the service and registered it. The MPC and DPC were generated. **But the DPC**

methods were not re-defined. We need to write our ABAP code and logic to pull data from the backend system and populate the Entity Set (internal table). Once the entity set is populated, we can see the output in our OData service call.

Let us just re-define the method and write no code. What do you expect the OData service to return (give output)?

Go to change mode in the DPC Extension class and not in Base class (we work in the extension class since we are enhancing the inherited class).

Class Builder: Change Class ZCL_ZGW_PO_DPC_EXT

Repository Browser

Class / Interface: ZCL_ZGW_PO_DPC_EXT

Object Name: ZCL_ZGW_PO_DPC_EXT

Description: Data Provider S...

Properties | Interfaces | Friends | Attributes | Methods | Events | Types | **Aliases**

Method	Level	Visibility	M... Description	Redefine Method
/IWBEPIF_SB_DPC_COMM_SERVICES-GET	Instance Method Public		Get generation strategy	
/IWBEPIF_SB_DPC_COMM_SERVICES-RFC	Instance Method Public		RFCall exception handling	
/IWBEPIF_SB_DPC_COMM_SERVICES-RFC	Instance Method Public		Save the RFC call log in the application log	
/IWBEPIF_SB_DPC_COMM_SERVICES-SET	Instance Method Public		SET DPC injection	
/IWBEPIF_SB_DPC_COMM_SERVICES-LOG	Instance Method Public		Log message in the logger	
/IWBEPIF_SB_DPC_COMM_SERVICES-COM	Instance Method Public		Call RFC commit work	
CHECK_SUBSCRIPTION_AUTHORITY	Instance Method Protected			
POHEADERSET_CREATE_ENTITY	Instance Method Protected		Related EntitySet Name: POHeaderSet	
POHEADERSET_DELETE_ENTITY	Instance Method Protected		Related EntitySet Name: POHeaderSet	
POHEADERSET_GET_ENTITY	Instance Method Protected		Related EntitySet Name: POHeaderSet	
POHEADERSET_GET_ENTITYSET	Instance Method Protected		Related EntitySet Name: POHeaderSet	←
POHEADERSET_UPDATE_ENTITY	Instance Method Protected		Related EntitySet Name: POHeaderSet	
POITEMSET_CREATE_ENTITY	Instance Method Protected		Related EntitySet Name: POItemSet	
POITEMSET_DELETE_ENTITY	Instance Method Protected		Related EntitySet Name: POItemSet	
POITEMSET_GET_ENTITY	Instance Method Protected		Related EntitySet Name: POItemSet	

Put the cursor the method you want to re-define. Hit the re-define method icon and hit enter. Some auto generated code would show up.

Class Builder: Class ZCL_ZGW_PO_DPC_EXT Change

Repository Browser

Class Builder: Class ZCL_ZGW_PO_DPC_EXT C

Object Name: ZCL_ZGW_PO_DPC_EXT

Description: Data Provider S...

Properties | Pattern | Pretty Printer | Signature | Public Section | Protected Section | Private Section

Method	POHEADERSET_GET_ENTITYSET	Active
1	method POHEADERSET_GET_ENTITYSET.	
2	*TRY.	
3	*CALL METHOD SUPER->POHEADERSET_GET_ENTITYSET	
4	* EXPORTING	
5	IV_ENTITY_NAME =	
6	IV_ENTITY_SET_NAME =	
7	IV_SOURCE_NAME =	
8	IT_FILTER_SELECT_OPTIONS =	
9	IS_PAGING =	
10	IT_KEY_TAB =	
11	IT_NAVIGATION_PATH =	
12	IT_ORDER =	
13	IV_FILTER_STRING =	
14	IV_SEARCH_STRING =	
15	** io_tech_request_context =	
16	** IMPORTING	
17	** et_entityset =	
18	** es_response_context =	
19	*	
20	** CATCH /iwbeplcx_mgw_busi_exception .	
21	** CATCH /iwbeplcx_mgw_tech_exception .	
22	**ENDTRY.	
23	endmethod.	

We would learn about all the importing and exporting parameters of the methods in next article. For now save as it is and activate the class. Check the folder **Redefinition** has the method 'POHEADERSET_GET_ENTITYSET' listed. This means our method is successfully re-defined.

Let us check the output. Go to t-code /n/IWFND/MAINT_SERVICE or /n/IWFND/GW_CLIENT and give the URI: /sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet.

SAP Gateway Client

HTTP Method: GET Request URI: /sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet Protocol: HTTP Test Group: Test Case

Reuse HTTP Connection (e.g. for Soft State) Multiple Rows

HTTP Request

Header Name	Value
~status_code	200
~status_reason	OK

HTTP Response - Processing Time = 297 ms

Header Name	Value
sap-processing-info	microhub=,crp=,st=,MedCacheHub=SHM,codeployed=,softstate=

```

<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
      xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
      xml:base="http://txaixegd01. retail.nrgenergy.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/*">

  <id>http://txaixegd01. retail.nrgenergy.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet</id>
  <title type='text'>POHeaderSet</title>
  <updated>2017-01-02T13:41:49Z</updated>
  - <authors>
    <name />
    </authors>
    <link href="POHeaderSet" rel="self" title="POHeaderSet" />
</feed>

```

This time we do not get any error. The status code is 200 and it is OK. But, the body has no data yet. Quite obviously, we did not write any logic to populate the internal table (entityset). ☺

Now, let us write one small statement to pull 10 entries from Purchase Order table EKKO in the method and activate it.

```

1 METHOD poheaderset_get_entityset.
2
3
4 SELECT * UP TO 10 ROWS FROM ekko INTO CORRESPONDING FIELDS OF TABLE et_entityset.
5
6 ENDMETHOD.

```

Class Builder: Class ZCL_ZGW_PO_DPC_EXT Change

Repository Browser: ZCL_ZGW_PO_DPC_EXT

Type	Parameter	Type spec.
IV_ENTITY_NAME		TYPE STRING
IV_ENTITY_SET_NAME		TYPE STRING
IV_SOURCE_NAME		TYPE STRING
IT_FILTER_SELECT_OPTIONS		TYPE /WBEP/T_MGW_SELECT_OPTION
IS_PAGING		TYPE /WBEP/S_MGW_PAGING
IT_KEY_TAB		TYPE /WBEP/T_MGW_NAME_VALUE_PAIR
IT_NAVIGATION_PATH		TYPE /WBEP/T_MGW_NAVIGATION_PATH
IT_ORDER		TYPE /WBEP/T_MGW_SORTING_ORDER
IV_FILTER_STRING		TYPE STRING
IV_SEARCH_STRING		TYPE STRING
IO_TECH_REQUEST_CONTEXT		TYPE REF TO /WBEP/IF_MGW_REQ_ENTITYSET OPTIONAL
ET_ENTITYSET		TYPE ZCL_ZGW_PO_MPC->TT_POHEADER

Method: POHEADERSET_GET_ENTITYSET

```

1 METHOD poheaderset_get_entityset.
2
3   SELECT * UP TO 10 ROWS FROM ekko INTO CORRESPONDING FIELDS OF TABLE et_entityset.
4
5 ENDMETHOD.

```

Now, let us check the output again.

SAP Gateway Client

HTTP Method GET POST PUT PATCH MERGE DELETE Reuse HTTP Connection (e.g. for Soft State) Multiple Rows

Request URI /sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet

Protocol HTTP HTTPS Test Group

HTTP Request

Add File Remove File Data Explorer

HTTP Response - Processing Time = 1039 ms

Header Name	Value
~status_code	200
~status_reason	OK

```
<d><http://txaixegd01>
<title type="text">POHeaderSet</title>
<updated>2017-01-02T13:48:21Z</updated>
- <author>
  <name />
</author>
<link href="POHeaderSet" rel="self" title="POHeaderSet" />
+ <entry>
</feed>
```

10 entries from EKKO

10 entries are returned. If you expand the entries, you would see the details of each row.

SAP Gateway Client

HTTP Method GET POST PUT PATCH MERGE DELETE Reuse HTTP Connection (e.g. for Soft State) Multiple Rows

Request URI /sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet

Protocol HTTP HTTPS Test Group

HTTP Request

Add File Remove File Data Explorer

HTTP Response - Processing Time = 1039 ms

Header Name	Value
~status_code	200
~status_reason	OK

```
<link href="POHeaderSet('4500002013')" rel="self" title="POHeader" />
- <content type="application/xml">
- <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices"
  xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
<d:Ebeln>4500002013</d:Ebeln>
<d:Bukrs>0319</d:Bukrs>
<d:Bstyp>F</d:Bstyp>
<d:Bsart>NB</d:Bsart>
<d:Loekz />
<d:Statu />
<d:Ernam>T020728</d:Ernam>
<d:Lifnr>100644</d:Lifnr>
<d:Zterm>ZN30</d:Zterm>
</m:properties>
</content>
</entry>
+ <entry>
+ <entry>
+ <entry>
```

Now, we can see the same output in the browser as well.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:i="http://www.w3.org/2005/Atom" xml:base="http://txaixegd01.../sap/opu/odata/sap/ZGW_PO_SRV/">  

  <i:id>  

    http://txaixegd01.../sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet  

  </id>  

  <title type="text">POHeaderSet</title>  

  <updated>2017-01-02T13:55:00Z</updated>  

  <author>  

    <name/>  

  </author>  

  <link href="POHeaderSet" rel="self" title="POHeaderSet"/>  

  <entry>  

    <i:id>  

      http://txaixegd01.../sap/opu/odata/sap/ZGW_PO_SRV/POHeaderSet('4500002013')  

    </id>  

    <title type="text">POHeaderSet('4500002013')</title>  

    <updated>2017-01-02T13:55:00Z</updated>  

    <category term="ZGW_PO_SRV.POHeader" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>  

    <link href="POHeaderSet('4500002013')" rel="self" title="POHeader"/>  

    <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" xmlns:i="http://www.w3.org/2005/Atom">  

        <d:Ebeln>4500002013</d:Ebeln>  

        <d:Bukrs>0319</d:Bukrs>  

        <d:Bstyp>F</d:Bstyp>  

        <d:Bsalt>NB</d:Bsalt>  

        <d:Loekz/>  

        <d:Statu/>  

        <d:Ernam>T020728</d:Ernam>  

        <d:Lifnr>100644</d:Lifnr>  

        <d:Zterm>ZN30</d:Zterm>  

      </m:properties>  

    </content>  

  </entry>  

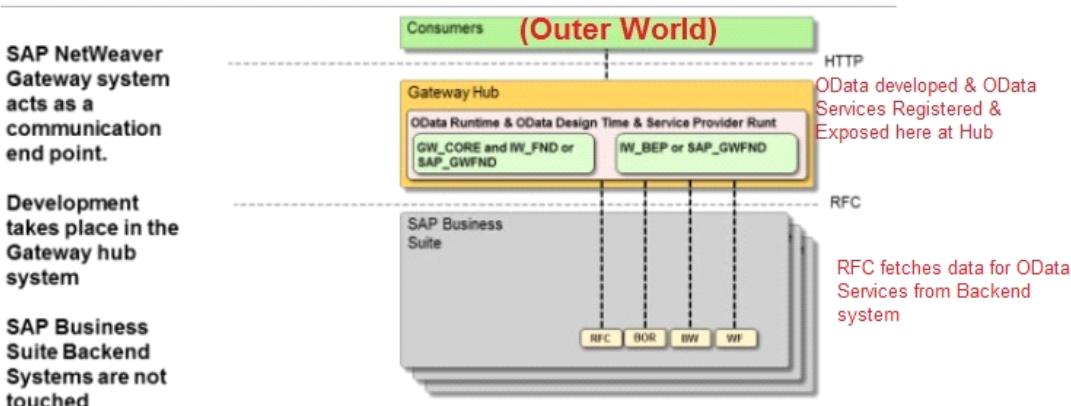
</entry>
```

Let us stop here for today. In the next post, we will implement all the other methods and manipulate the URIs to meet our requirement. We would show you how **Association and Navigation** links can be created to flow/pull/push data to and from different Entity (i.e. Navigation and Association between data models). The next article will be more hands on with lots of code and tips. So, please stay tuned. ☺

Bonus Contents for this Post

i. Hub Deployment with Development in the Gateway Hub

There is a third Deployment Strategy, in addition to the two mentioned above, for SAP Netweaver Gateway. It is "**Hub Deployment with Development in the Gateway Hub**". The Gateway component need not be installed in the backend system as *all the development related to SAP Netweaver Gateway takes place in the Gateway Hub*. If we don't want to do any kind of developments in the backend system and leverage the standard/custom what is already existing in the system, this might be the strategy.



Hub Deployment with Development in the Gateway Hub Strategy

Disadvantages of Hub Deployment with Development in the Gateway Hub:

- No access to backend data dictionary objects directly. Only the existing BAPIs and RFCs can be the data source for OData Service development

iii. Additional cost and maintenance effort as compared an embedded deployment option with just one system.

ii. How is the URL for OData Service determined for the Web Browser?

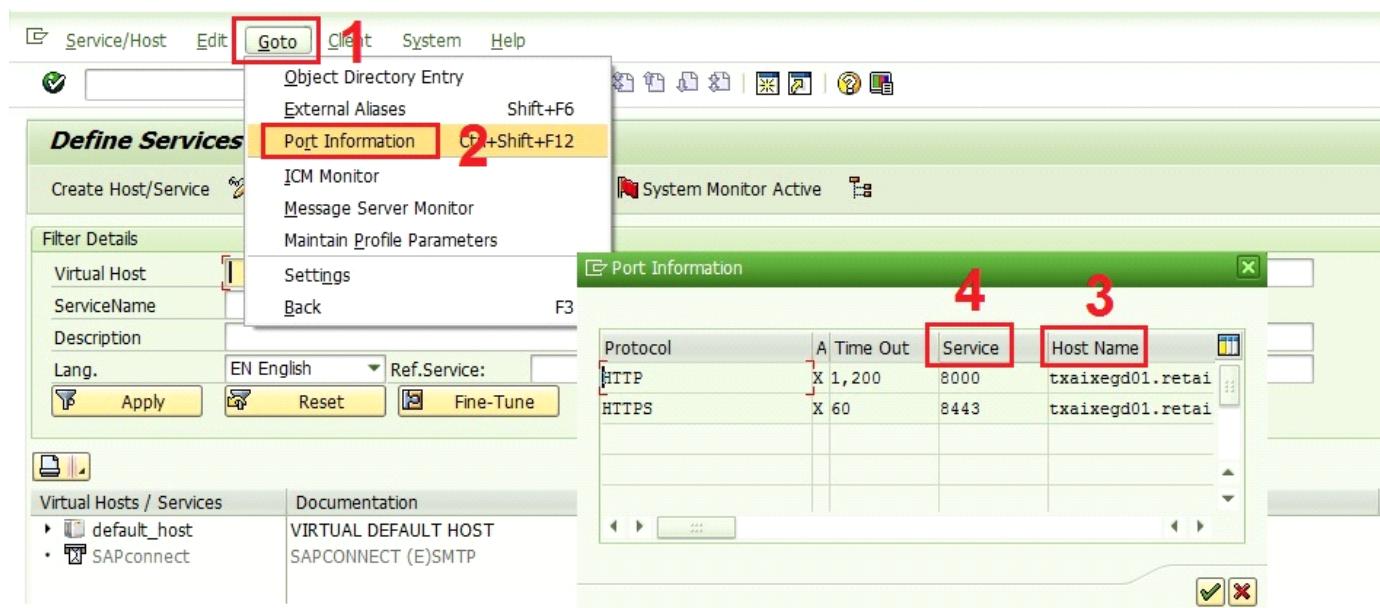
In this post, we have used t-code /IWFND/MAINT_SERVICE and execute SAP Gateway Client (or directly from t-code /IWFND/GW_CLIENT) to test our OData Service in SAP screen. We also saw that we can call the URL in web browser and test it as were doing in SAP Gateway Client. But, how actually is the URL created for Web Browser?

Ans: The actual URL for OData in Web Browser consists of:

http(s)://<Hostname>:<Service i.e Port Number>/ (rest is from the relevant ICF path).

Now another question arises, how do we know the Hostname, Service (Port Number) and ICF path of our SAP system?

Ans: Go to t-code **SICF**. Execute it with default values. Menu **Goto->Port Information**. The pop-up window would have the **Hostname** and the **Service name (Port Number)** for the system.



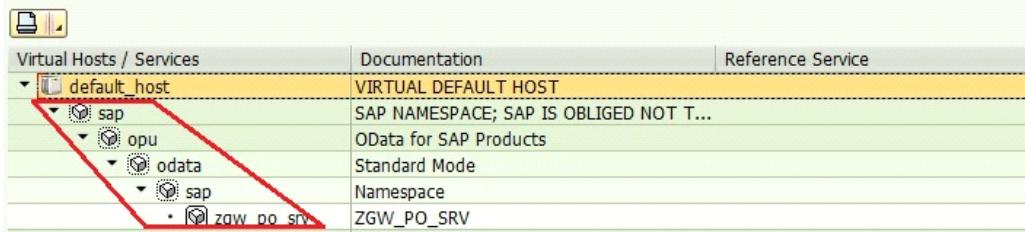
In order to find the default ICF path, provide the service name in the t-code SICF and execute it. Expand the roots till you reach the service name. For our example the ICF path is **sap->opu->odata->sap->zgw_po_srv**.

Define Services

Create Host/Service External Aliases System Monitor Active

Filter Details

Virtual Host	<input type="text"/>	Service Path	<input type="text"/>
ServiceName	ZGW_PO_SRV		
Description			
Lang.	EN English	Ref.Service:	<input type="text"/>
Apply		Reset	Fine-Tune


 A tree view showing the hierarchy of virtual hosts and services. The root node is 'default_host'. It has three children: 'sap', 'opu', and 'odata'. 'odata' has one child 'sap', which in turn has one child 'zgw_po_srv'. A red triangle points from the text 'sap/zgw_po_srv' in the previous section to the 'sap' node under 'odata'.

/sap/opu/odata/sap/ZGW_PO_SRV/

Now concatenate the **http**, **Hostname**, **Service(Port Number)** and **ICF path**. For our example, it would be http://txaixegd01.retail.sapyard.com:8000/sap/opu/odata/sap/ZGW_PO_SRV/. So next time if someone asks you what is the Hostname and Port of your system, you know how to find it. ☺

Next Post: [OData and SAP Netweaver Gateway. Part III. Query Options in OData Service URI](#)

If you want to get such useful articles directly to your inbox, please **SUBSCRIBE**. We respect your privacy and take protecting it seriously.

*If you liked this post, please hit the **share** buttons and like us on [facebook](#).*

Do you have anything to add to this article? Have you faced any issue understanding OData or SAP Netweaver Gateway? Do you want to share any real project requirement or solutions? Please do not hold back. **Please leave your thoughts in the comment section.**

Thank you very much for your time!!

Call for [Guest Authors](#) and [Contributors](#) to write SAP Articles on our page and get noticed and also [receive cool Gifts](#).

*Do you have any tips or tricks to share? **Do you want to write some articles at SAPYard?** Please [REGISTER](#) and start posting and sharing your knowledge to the SAP world and get connected to your readers. Please check our [simple guidelines](#) for contributing your articles and receiving the gifts.*

Step by Step Tutorials on SAP Netweaver Gateway and OData

- [OData and SAP Netweaver Gateway. Part I. Introduction](#)
- [OData and SAP Netweaver Gateway. Part II. Create your first OData Service](#)
- [OData and SAP Netweaver Gateway. Part III. Query Options in OData Service URI](#)
- [OData and SAP Netweaver Gateway. Part IV. Association and Navigation in OData Service](#)
- [OData and SAP Netweaver Gateway. Part V. CRUD Operations in OData Services](#)
- [OData and SAP Netweaver Gateway. Part VI. Frequently Asked Questions](#)
- [OData and SAP Netweaver Gateway. Part VII. Debugging, Trace, Cache Cleanup and F4 Help](#)
- [OData and SAP Netweaver Gateway. Part VIII. SAP's Love for OData – a Tale of the Friendly ABAPer](#)
- [CDS Part 3. Expose CDS Views as OData Service through Annotation](#)

- [OData and SAP Netweaver Gateway. Part IX. How to Add Multiple Entities in One Operation in OData Service](#)
- [OData and SAP Netweaver Gateway. Part X. How to Delete OData Service from Service Catalog?](#)
- [OData and SAP Netweaver Gateway. Part XI. Query Options & HTTP Status Code Summary](#)
- [SAP Netweaver Gateway and OData. Part XII. Media Handling using OData Gateways](#)
- [SAP Netweaver Gateway and OData. Part XIII. Entity Tags in SAP OData Gateways](#)
- [SAP Netweaver Gateway and OData. Part XIV. OData Service using RFC](#)
- [Free Video Course – Introduction to SAP Netweaver Gateway & OData](#)
- [SAPUI5 for ABAPers – Consuming OData Service from SAPUI5 Application – CRUD Operations](#)
- [Advance SAPUI5 – 2- Push Notification in SAP – ABAP Push Channel, ABAP Messaging Channel in SAPUI5 – a Real Time Interaction](#)
- [CDS Part 11. How to Consume CDS View in Smart Business Service KPI Fiori Apps?](#)
- [XSOData and Hana Database Views for Beginners](#)
- [OData Service from CDS Annotation Not Working in Browser Mode](#)
- [SAPUI5 – Custom Control in UI5](#)
- [Using Postman to “POST” without modifying ~CHECK_CSRF_TOKEN in ICF settings.](#)
- [ABAP Programming Model for SAP Fiori – 4 – OData Service Creation](#)
- [ABAP Programming Model for SAP Fiori – 5 – OData Service Generation](#)

Also, Check Step by Step Tutorials on SAPUI5

- [Journey to SAPUI5](#)
- [SAPUI5 Tutorial with WebIDE. Part I. How to Consume Custom OData in SAPUI5 Application](#)
- [SAPUI5 Tutorial with WebIDE. Part II. Routing and Navigation in SAPUI5 Application](#)
- [SAPUI5 Tutorial with WebIDE. Part III. Drop Down in SAPUI5 Applications \(2 Methods\)](#)
- [SAPUI5 Tutorial. Part IV with WebIDE. Routers and Routing in SAPUI5](#)
- [SAPUI5 Tutorial with WebIDE. Part V. Navigation in SAPUI5 without Routers](#)
- [SAPUI5 Tutorial with WebIDE. Part VI. Using Fragments in SAPUI5 Fiori Applications](#)
- [SAPUI5 Tutorial with WebIDE. Part VII. An ABAPer’s First SAPUI5 App in SAP WebIDE](#)
- [SAPUI5 Tutorial with WebIDE. Part VIII. Deploy my First SAPUI5 App in WebIDE](#)
- [SAPUI5 Tutorial with WebIDE. Part IX. Alternative to oModel.setSizeLimit\(\)](#)
- [SAPUI5 Tutorial with WebIDE. Part X. Using Media Queries in UI5 Application](#)
- [SAPUI5 Tutorial with WebIDE. Part XI. An ABAPer’s Second SAPUI5 App](#)
- [SAPUI5 Tutorial with WebIDE. Part XII. SAPUI5 Basic Debugging for Beginners](#)
- [Routing and Navigation in SAP UI5 – Theoretical Explanation Part 1](#)
- [Routing and Navigation in SAP UI5 – Theoretical Explanation Part 2](#)
- [Add Delete & Save Multiple Records in SAPUI5. Part 1 – ADD](#)
- [Add Delete & Save Multiple Records in SAPUI5. Part 2 – DELETE](#)
- [Add Delete & Save Multiple Records in SAPUI5. Part 3 – SAVE](#)
- [Modularization and Large Scale Architecture in SAPUI5](#)
- [Use of Third Party \(or\) External Resources in SAPUI5. Part I – Overview](#)
- [Use of Third Party \(or\) External Resources in SAPUI5. Part II – Practical with Gauges](#)
- [How to Deploy UI5 App without LPD_CUST?](#)
- [SAPUI5 For ABAPers – Component Reuse with Real Time Example](#)
- [SAPUI5 for ABAPers – Consuming OData Service from SAPUI5 Application – CRUD Operations](#)
- [Advance SAPUI5 – 1- Trick to Send QR code or Barcode Data Remotely from Android to PC for SAPUI5 App](#)
- [Advance SAPUI5 – 2- Push Notification in SAP – ABAP Push Channel, ABAP Messaging](#)

[Channel in SAPUI5 – a Real Time Interaction](#)

- [Advance SAPUI5 – 3 – How to send e-mail in SAPUI5 Hybrid App?](#)
- [Advance SAPUI5 – 4 – How to Get Weight from the Weigh Scale/Weigh Bridge Bluetooth Device using SAPUI5 Hybrid App?](#)
- [SAPUI5 – How to Change the Master List Item Selection Based on Changes in Hash Tag URL?](#)
- [SAPUI5 – How to Load a Full Screen and Navigate to Master-Detail Application?](#)

[Approach 1](#)

We have a very active Telegram SAP Technical Group. Please join it using below link.

[Telegram SAP Technical Discuss Group.](#)

WhatsApp had the limitation of 256 members per group and we were finding a hard time syncing and maintaining information in multiple WhatsApp group. Therefore we have moved to Telegram as it can accommodate more than 10,000 users. [Please join.](#)

- TAGS
- [A simple overview on SAP Netweaver Gateway](#)
- [Embedded Deployment Strategy](#)
- [Let's Talk OData](#)
- [NetWeaver Gateway & SAPUI5](#)
- [OData Tutorial](#)
- [URI](#)
- [What is OData?](#)

[Facebook](#)

[Twitter](#)

[Pinterest](#)

[WhatsApp](#)

Previous article [Data Lake and Data Warehouse](#)

Next article [SAP Adobe Interactive Form Tutorial. Part III. Date Time and Floating Fields](#)



[SAP Yard](#)

<https://sapyard.com/>

SAPYARD is one stop page for all Technical Folks in SAP. You would find un-conventional explanations, tutorials, tricks and end to end Free SAP Video Courses and Training. Please like our [Facebook Page](#), follow us at [Twitter](#), [Instagram](#) and also join our [LinkedIn Group](#). Please Subscribe to our [Youtube Channel](#) for Free SAP Video Trainings.

[RELATED ARTICLES](#) [MORE FROM AUTHOR](#)



[ABAP Programming Model for SAP Fiori – 12 – Scenario of Transactional App with Draft Enabled Functionality](#)



[ABAP Programming Model for SAP Fiori – 11 – Enabling Draft Functionality for Transactional Apps](#)



[ABAP Programming Model for SAP Fiori – 10 – Use of Actions in BOPF](#)

47 COMMENTS



[chandeep](#) January 9, 2020 At 5:40 am

awsome tutorial

[Reply](#)



[SAPYard](#) January 13, 2020 At 3:26 pm

Thank you Chandeep. Please keep visiting.

Team SAPYard.

[Reply](#)



[Naveen Kumar](#) December 23, 2019 At 5:43 am

Hi Thanks for your wonderful post and can you explain me how to query the entity set without giving the key parameter Example(I want to find the po for a particular company so should I make BUKRS as a key field or not please explain)

[Reply](#)



Pulkit1307 [December 18, 2019 At 7:35 am](#)

Hi, The above example is for creating an OData service on top of a table. Is it possible to also create it on a standard datasource in SAP ERP?

[Reply](#)



Ravster [August 23, 2019 At 2:15 pm](#)

I have gone through many blogs from different sources but this one is the BEST!! Awesome!!! The diagrams, flow, detail on each item etc., hats off really!!

[Reply](#)



SAP Yard [August 23, 2019 At 4:05 pm](#)

Thank you Mithun. We strive hard to make our content unique and as informative as possible. So that you can return to our page more often. 😊

Please keep visiting and sharing our links with your contacts.

Team SAPYard.

[Reply](#)



Shashikant [July 24, 2019 At 10:57 am](#)

What is the difference between external service name and technical service name ?

[Reply](#)



SAP Yard [July 24, 2019 At 11:35 am](#)

Hi Shashikant – Usually both are kept same. But external service name is the one which can be mapped to external non sap system. Technical service name is the SAP Service name for the corresponding Service.

For now, we keep both names same.

Team SAPYard.

[Reply](#)



SBhat [May 9, 2019 At 1:10 pm](#)

Long live spreading knowledge!

Kudos SAPYard!

[Reply](#)



SAP Yard [May 9, 2019 At 1:32 pm](#)

Thank you SBhat. Please keep visiting.

Do check our [Free End to End Video Course](#).

Regards,

Team SAPYard

[Reply](#)



Nirmi [May 4, 2019 At 6:48 pm](#)

thank you very much to the Author for explaining it from an ABAPer's perspective.
I like everything about this article – the example, flow, screenshots, details, segue into architecture and coming back.
I greatly appreciate for putting all these articles together for the best benefit of the community.

[Reply](#)



SAP Yard [May 5, 2019 At 9:28 am](#)

Thank you Nirmit. Glad you liked it. Please share our links with your friend, colleagues and team members. Please [subscribe to our Youtube channel](#) too.

Regards,
Team SAPYard

[Reply](#)



Nilesh [September 28, 2018 At 12:49 am](#)

HI
Nicely explained. I have one query. Do we need to repeat the process for service register and gateway client every time for Odata, or its one time activity.

[Reply](#)



SAP Yard [September 28, 2018 At 8:33 am](#)

Thanks Nilesh. No. You do not need to repeat the registration. It is one time for a service.

Also, check our New Video Course on OData and Netweaver Gateway.

<https://sapyard.com/courses/sap-netweaver-gateway-and-odata-for-beginners/?tab=tab-curriculum>

Team SAPYard.

[Reply](#)



Prashanth S [May 15, 2018 At 10:36 am](#)

excellent post...is it possible to create a video instead of lengthy post, it will be more effective 😊

[Reply](#)



SAP Yard [May 15, 2018 At 9:16 pm](#)

Thank you Prashanth for the feedback. Sure, we will surely try to start our Video Blog and series. Videos need more effort and time. We will commence it soon.

Team SAPYard.

[Reply](#)



[Pablo May 11, 2018 At 1:39 pm](#)

Excellent tutorial, thanks very much for the post!

[Reply](#)



[Girdhari Mondal April 18, 2018 At 8:29 am](#)

Excellent article! I have read many articles on OData and found SAP YARD as the best place to learn. It helps me to clear the confusions which i had after reading other articles on OData. Thank you SAP YARD! Highly appreciated!

[Reply](#)



[SAP Yard April 18, 2018 At 11:04 am](#)

Thank you Girdhari.. Our articles are from real project issues.. And we know the pain of the beginners. So we write thinking them as the target audience. Glad you liked it.

Please spread the word and keep visiting. ☺

Team SAPYard.

[Reply](#)



[Ahmed Ali Khan February 14, 2018 At 5:08 am](#)

Seriously, I am stunned, how can someone teach you like he is in front of you and knows where are you struggling what questions you have in mind....What a classy teaching and putting words that increase your interest in reading it more not like bookish geeks....wonderful explanation****HATS OFF SAP YARD***

[Reply](#)



[SAP Yard February 14, 2018 At 12:11 pm](#)

Dear Ahmed – We are happy, you liked our presentation style. Most of our articles are from real life learning. We note down the pain we faced while we were learning ourself. And try to put them for others benefits. Nothing better than learning from someone else's mistake.

Please bookmark our page and keep visiting. Our idea is not to be just another SAP ABAP blog. We want to be different and add value.

Please share our links with your friends, colleagues and team members.

Regards,

Team SAPYard.

[Reply](#)



[Kapil January 25, 2018 At 2:41 am](#)

I do not generally subscribe to any blog or website , and if i like it then generally make it as bookmark , but it is the first sap website for which i have subscribed ,Also I would be much obliged if you could let your user customize what are they interested in .

Like I want info on HANA and FIORI so I would like I get more suggestions and articles on these two (I have just subscribed now , so excuse me if i have not discovered your entire functionalities).

Keep up the good work

Thanks

Kapil

[Reply](#)



[SAP Yard January 25, 2018 At 9:29 am](#)

Dear Kapil – Thank you so much for your wonderful feedback. This is one of the best reviews we have ever received. We are glad we convinced you that SAPYard is different and beneficial to SAP Consultants.

Thank you very much for subscribing to our site. We hope we would not let you down. We have very good series on HANA-ABAP. Please check it at your convenience.

<https://sapyard.com/category/tutorial/hana/sap-abap-on-hana/>

We also have Fiori. Please do check it.

<https://sapyard.com/category/tutorial/sap-fiori/>

Please share our link with your friends, colleagues and team members.

Regards,

Team SAPYard.

[Reply](#)



[farheen January 9, 2018 At 6:05 am](#)

very helpful article...desperately waiting for upcoming blogs 😊

[Reply](#)



[SAP Yard January 9, 2018 At 9:03 am](#)

Dear Fahreen – Thank you so much for stopping by and putting your thoughts.

We already have the complete series. Please check the Part 3 below.

<https://sapyard.com/odata-and-sap-netweaver-gateway-part-iii-query-options-in-odata-service-uri/>

Reagards,

Team SAPYard.

[Reply](#)



[Vikas Mishra January 2, 2018 At 6:18 am](#)

Very useful article SAPYard Team. Appreciated 😊

[Reply](#)



[SAP Yard January 2, 2018 At 10:04 am](#)

Thank you Vikas. Please keep visiting and sharing our links.

Regards,

Team SAPYard.

[Reply](#)



[Rakhi December 20, 2017 At 8:04 am](#)

Beautifully explained. 😊

[Reply](#)



[SAP Yard December 20, 2017 At 9:28 am](#)

Thank you Rakhi. Please keep visiting and spread the word. ☺

Regards,

Team SAPYard.

[Reply](#)



[Ashish October 29, 2017 At 3:51 am](#)

You guys are amazing. Huge respect for you guys.

[Reply](#)



[SAP Yard December 19, 2017 At 10:53 am](#)

Thank you Ashish. Please keep visiting and sharing our links.

Team SAPYard.

[Reply](#)



[Anoop Karanam October 25, 2017 At 12:11 pm](#)

Nice blog.

Is it possible to create a nested structure? How will the oData appear in such cases?

Regards,

Anoop

[Reply](#)



[SAP Yard December 19, 2017 At 10:56 am](#)

Sure Anoop. We will put something on Nested Structures. Thanks for the idea.

Team SAPYard.

[Reply](#)



[Amutha Vasudevan September 27, 2017 At 4:03 am](#)

Heartly thanks to SAPYARD team. It is a very nice series of article. After i found the SAPYARD articles, i did not approach any other sites to refrence .

[Reply](#)



[SAP Yard September 27, 2017 At 6:24 am](#)

Dear Amutha – This is one of the best compliments we have received. Please share our links to your friends, colleagues and team members.

Please suggest us, what else you want to see on SAPYard.

Regards,

Team SAPYard.

[Reply](#)



Jun Ai [September 13, 2017 At 10:24 am](#)

Thanks for the posting. Very useful information. Is there a part III for this? I am interested in seeing how the get_entity implemented for the POItem where you have multiple keys. Also would like some more info around how the association/Navigation works. Are those handled by the SAP automatically after you define the relationship or is there any code we have to do as developer? If that is done by SAP, is there any BAdI/enhancement point available in case we want to change the standard behavior?

Thanks. Enjoy the reading a lot. Please keep going. ☺

Jun

[Reply](#)



Jun Ai [September 13, 2017 At 10:28 am](#)

Actually just found the Part III link when reading other people's comments. Guess I should read before commenting. ☺

[Reply](#)



Prit_M [January 30, 2017 At 12:38 pm](#)

Hi Gurus @Sapyard, first of all I would like to thank you for this nice series of tutorials. However I have a question for you. I would like to know if we need to have any specific license for these kind of developments. What kind of licensing is required for the use of SAP NetWeaver component or SAP Gateway component?

Thanks Again ☺

[Reply](#)



sapyard [January 30, 2017 At 4:42 pm](#)

Dear Prit – Thank you for your visit and your question. There is no licensing fee for the components and Add-ons.

Regards,

Team SAPYard.

[Reply](#)



Sarah [January 9, 2017 At 1:59 pm](#)

Nice article!! very similar to <http://www.sapdev.co.uk/webapps/seqw-odata-gateway-service.htm> even uses the same database table(ekko) and select code!!

[Reply](#)



sapyard [January 9, 2017 At 4:03 pm](#)

Dear Sarah – Thank you very much for your feedback and the link.

The similarity of the database table used is pure coincidental. ☺ . At the end of our series, our service would have many more other database tables and operations. In [Part III](#) we have already added EKPO. Part IV would have more.

Hope you would find our content better and covering the subject in more detail.

The simple point where we showed the way to find Hostname and Port, our readers

found them interesting. 😊

Please keep visiting and leaving your feedback.

Regards,

Team SAPYard.

[Reply](#)



[Feleciano Buenavista January 6, 2017 At 2:14 am](#)

Very useful article SAPYard Team. Copied and bookmarked for further readings.

[Reply](#)



[sapyard January 6, 2017 At 3:23 am](#)

Thank you Feleciano. Happy that you found our post worthy enough for your bookmark.

The third part on OData is in the final stage of review. It would be published soon.

Please keep visiting and leaving your valuable feedback.

Regards,

Team SAPYard.

[Reply](#)



[purushotham January 5, 2017 At 6:54 am](#)

Hi, Thanks for providing such a useful information

Waiting for part 3.....

[Reply](#)



[sapyard January 5, 2017 At 2:22 pm](#)

Dear Purshotham – Thank you so much for your feedback. Part 3 is already drafted. It would be published soon.

Please stay tuned.

Regards,

Team SAPYard.

[Reply](#)



[Joy Claxino January 3, 2017 At 10:39 am](#)

Thank you for the blog 😊

[Reply](#)



[sapyard January 3, 2017 At 4:02 pm](#)

You are welcome Dear Joy. We are happy, you find it useful.

The draft for our Part III on OData and SAP NWGW is ready. We would publish them soon.

Please keep visiting.

Regards,

Team SAPYard.

[Reply](#)

Please help us improve. Please Shout Often & Loud!!

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

SAPYard's YouTube Channel

Video Player

https://youtu.be/9-QSXJz_Qpw

00:00

00:00

[Use Up/Down Arrow keys to increase or decrease volume.](#)

Donate & Support SAPYard

[Donate](#)



Recent Comments

- Vignesh on [CDS Part 18 – Bar Chart & Donut Chart using CDS Views](#)
- tohid786 Shaikh on [Introduction to JavaScript & SAPUI5 for ABAP Developers Training – Details](#)
- Pavan on [Introduction to JavaScript & SAPUI5 for ABAP Developers Training – Details](#)
- Vignesh on [ABAP Programming Model for SAP Fiori – 9 – Use of Determinations in BOPF](#)
- Vignesh on [ABAP Programming Model for SAP Fiori – 8 – Developing Transactional Application – CRUD Operations using BOPF](#)

EDITOR PICKS



[ABAP Programming Model for SAP Fiori – 12 – Scenario of...](#)

March 28, 2020



[ABAP Programming Model for SAP Fiori – 11 – Enabling Draft...](#)

March 28, 2020



[ABAP Programming Model for SAP Fiori – 9 – Use of...](#)

March 28, 2020

POPULAR POSTS



[OData and SAP Netweaver Gateway. Part II. Create your first OData...](#)

January 2, 2017



[ABAP on SAP HANA. Part I. First Program in ABAP HANA](#)

May 19, 2016



[OData and SAP Netweaver Gateway. Part I. Introduction](#)

December 29, 2016

POPULAR CATEGORY

- [Tutorial313](#)
- [ABAP122](#)
- [Tweaks84](#)
- [SAPUI565](#)
- [HANA47](#)
- [HANA-ABAP45](#)
- [Code Snippets41](#)
- [Tips41](#)
- [SAP ABAP on HANA35](#)



ABOUT US

Contact us: mail@sapyard.com

FOLLOW US

- [Forums](#)
- [All SAP Tutorials](#)
- [Tutorials on SAP ABAP on HANA](#)
- [Write & Earn](#)
- [SAP Interview Q&A](#)
- [SAPYard YouTube Channel](#)
- [Shop & Support](#)
- [Video Tutorials](#)
- [Donate](#)

© 2020 TAC Global Business Services LLC

[Sumo](#)

Shares





Inserted from <<https://sapyard.com/odata-and-sap-netweaver-gateway-part-ii-create-your-first-odata-service/>>

Odata on Gateway

Wednesday, January 25, 2023 7:35 AM

The screenshot shows the SAP Fiori Launchpad interface. The top navigation bar includes icons for 'Take control', 'Pop out', 'People', 'Chat', and 'Re'. Below the header, there's a search bar with a dropdown arrow, followed by 'Get Services', 'Cancel', and 'More' buttons. A 'Filter' section contains fields for 'System Alias' (D47CLNT200), 'Technical Service Name' (*KANBAN*), 'External Service Name', 'Co-Deployed' (checkbox), 'Version', and 'External Mapping ID'. The main area is titled 'Select Backend Services' and lists two entries: 'BEP ZDEMO_KANBAN_SRV_SRV' and 'BEP ZODATA_KANBAN_SRV'. The second entry is selected, highlighted with a blue background.

The screenshot shows the SAP Fiori Launchpad interface. The top navigation bar includes icons for 'Create Host/Service', 'Delete', 'Edit', 'New', 'External Aliases', 'System Monitor Active', 'Cancel', and 'More'. A 'Filter Details' section contains fields for 'Virtual Host' (KANBAN), 'Service Path', 'ServiceName' (*KANBAN*), 'Description', 'Lang.' (EN English), 'Ref.Service', and buttons for 'Apply', 'Reset', and 'Fine-Tune'. The main area is titled 'Virtual Hosts / Services' and displays a hierarchical tree of service components under the 'sap' virtual host. The 'sap' node has children 'bc', 'bsp', and 'ZRTC_WM_KANBAN'. The 'bsp' node has a child 'sap' which contains 'ZRTC_WM_KANBAN'. The 'sap' node under 'sap' also has children 'gui', 'its', and 'ui5'. The 'ui5' node has a child 'sap' which contains 'ZRTC_WM_KANBAN'. Other nodes include 'opu', 'odata', and 'ZRTC_WM_KANBAN'. To the right of the tree, there are columns for 'Documentation' and 'Reference Service'.

SAP Gateway Service Builder

Runtime Artifacts

Name	Generated Artifact Type	Program ID	Object Type	Object Name
ZCL_RTC_ODATA_KANBAN_DPC	Data Provider Base Class	R3TR	CLAS	ZCL_RT
ZCL_RTC_ODATA_KANBAN_DPC_EXT	Data Provider Extension Class	R3TR	CLAS	ZCL_RT
ZCL_RTC_ODATA_KANBAN_MPC	Model Provider Base Class	R3TR	CLAS	ZCL_RT
ZCL_RTC_ODATA_KANBAN_MPC_EXT	Model Provider Extension Class	R3TR	CLAS	ZCL_RT
ZRTC_ODATA_KANBAN_MDL	Registered Model	R3TR	IWMO	ZRTC_O
ZRTC_ODATA_KANBAN_SRV	Registered Service	R3TR	IWSV	ZRTC_O

AssignmentSet

- Create
- Delete
- GetEntity (Read)
- GetEntitySet (Query)
- Update

StatusChangeResultSet

- Create
- Delete
- GetEntity (Read)
- GetEntitySet (Query)
- Update

TransferOrderItemSet

TransferOrderSet

Runtime Artifacts

- ZCL_RTC_ODATA_KANBAN_DPC
- ZCL_RTC_ODATA_KANBAN_DPC_EXT
- ZCL_RTC_ODATA_KANBAN_MPC
- ZCL_RTC_ODATA_KANBAN_MPC_EXT
- ZRTC_ODATA_KANBAN_MDL
- ZRTC_ODATA_KANBAN_SRV**

Service Maintenance

- UIS_FIORI
- UIS_FIORI

SAP Gateway Service Builder

ODP ODATA

OData Service for KANBAN App

ZBW_ODP_ODATA

ZRTC_ODATA_KANBAN

Data Model

- Entity Types
- Assignment
- StatusChangeResult
- TransferOrder
- TransferOrderItem
- Associations
- TransferOrderToTransferOrderItem
- Entity Sets
- Association Sets
- Function Imports
- UpdateKanban
- Function Import Parameters
- KanbanId
- Status

Service Implementation

Runtime Artifacts

Service Maintenance

SAP Gateway Service Builder

ODP ODATA

OData Service for KANBAN App

ZBW_ODP_ODATA

ZRTC_ODATA_KANBAN

Discuss Fiori App development

21:29 Take control Pop out People Chat Reactions Rooms Apps More Camera Mic

SAP Define Services

/nsgw/ Create Host/Service External Aliases System Monitor Active Cancel More

Filter Details

Virtual Host Service Path
ServiceName *FLP*
Description
Lang. EN English Ref.Service:
Apply Reset Fine-Tune

Virtual Hosts / Services Documentation Reference Service

- default_host VIRTUAL DEFAULT HOST
- sap SAP NAMESPACE; SAP IS OBLIGED NOT TO DELIVER ANY SERVICES OF THE HTTP FRAME...
 - bc BASIS TREE (BASIS FUNCTIONS)
 - bsp BUSINESS SERVER PAGES (BSP) RUNTIME
 - ra CDM/runtime agent
- ui2 UI Extension
 - ftp SAP Fiori launchpad with dynamic page processing Supports cache-busting
 - ui5_apis SAPUI5 Application Handler SAPUI5 Application called via HTTP out of SAPUI5 repository
 - ui5 sap sap Namespace for SAPUI5 Applications
 - sui_ftp_app_sup App Support Plugin
 - ui2 namespaces
 - c2gflppugin C2G Plugin for Fiori Launchpad
 - vui namespace
 - vui Fiori Launchpad Chart Tile
 - webdynpro Web Dynpro (WD) Runtime
- webdynpro

How to create OData service in SAP NetWeaver Gateway - SAPUI5 Tutorials

Thursday, January 26, 2023 12:30 PM

Clipped from: <http://www.sapui5tutors.com/2021/01/how-to-create-odata-service-in-sap.html>

In our previous blog, we looked into the introduction to OData and SAP NetWeaver Gateway. In this blog we will go through step by step process to create a SAP OData service. Later on, we will register our service and also test if it is working fine.

Step by Step Process to create SAP OData service

In this section we will follow multiple predefined steps to create an OData, these are:

1. We will use SAP Service Gateway Builder (SEGW) to create a new project
2. We will Import DDIC/CDS or other Structure to create an OData Model
3. We will Generate our OData
4. We will Register and test our Service

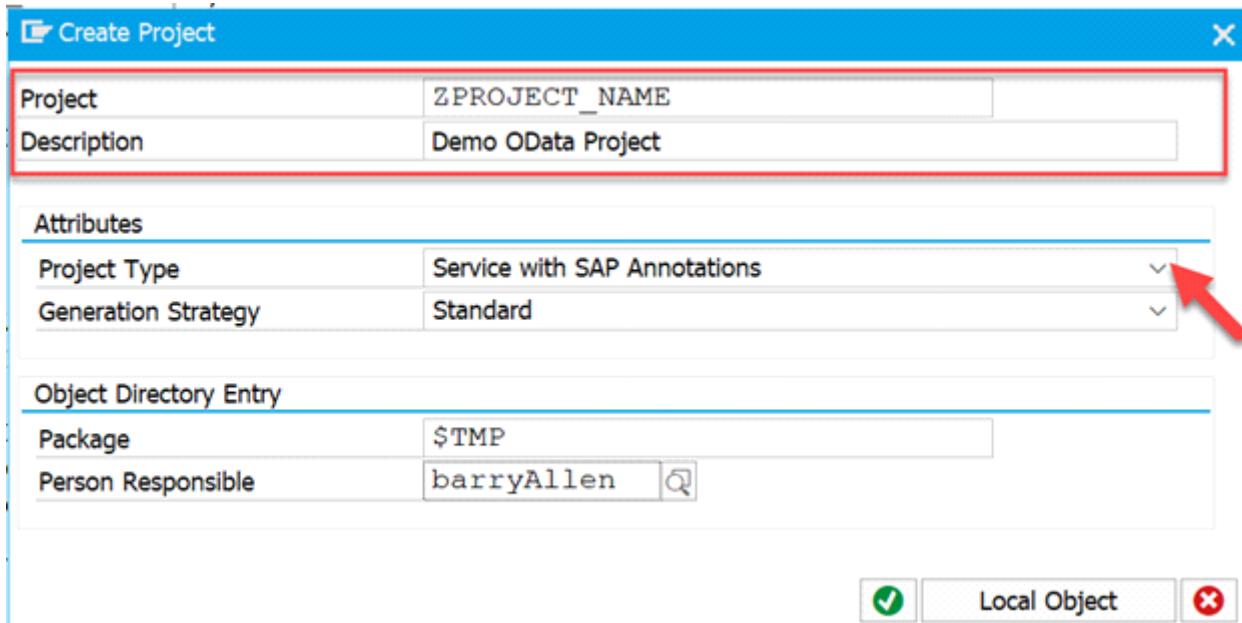
Create a new Service Builder project

To create a new project, follow the given steps:

1. Go to transaction code SEGW, it will start the Gateway Service Builder.
2. Click on the Create Project button, as shown below:
3. Enter the given information

Field	Description	Example
Project	It is the name of your SAP OData Service. It will start with Z or Y	ZPROJECT_N AME
Description	Enter a description for your OData	Demo OData Project
Project Type	SAP Provides given four Project Options: 1. Service with SAP Annotations 2. Service with Vocabulary-Based Annotations 3. Annotation Model for Referenced Service 4. OData 4.0 Service We will learn about them in details, later.	Service with SAP Annotations
Generation Strategy	It will be preloaded as Standard which means, it Standard has all standard configurations.	
Package	Enter your project package or temporary package i.e. \$TMP	\$TMP

Person Responsible It will be preloaded with your User ID/ User Name barryAllen

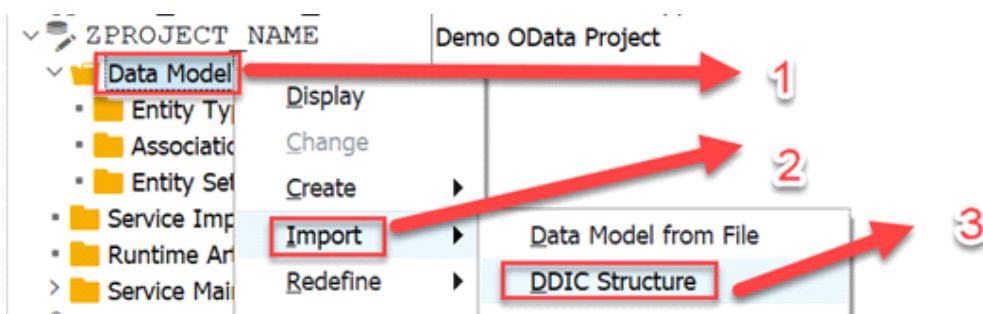


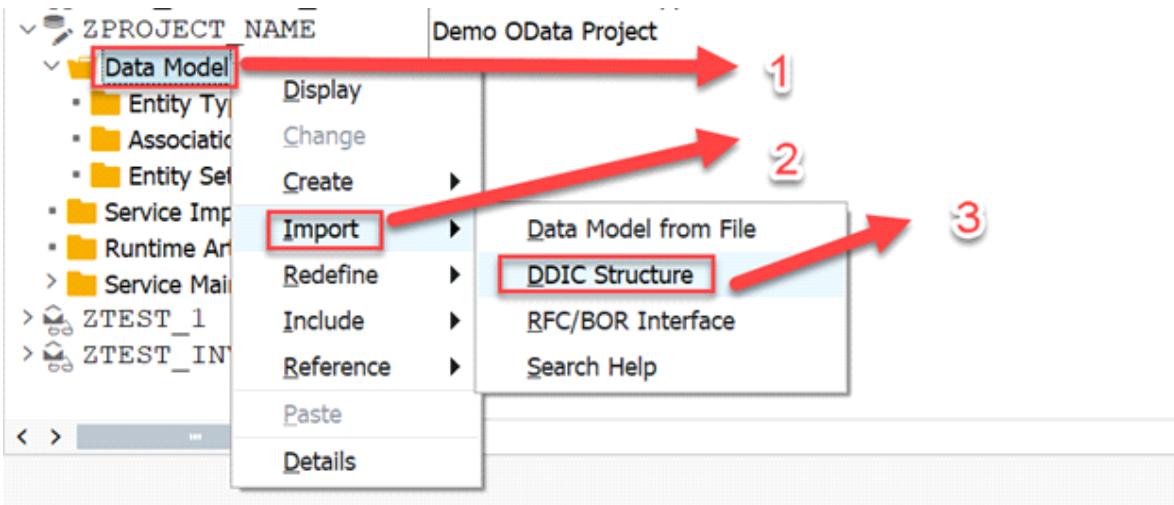
1. Press Continue and Save.

Import a DDIC structure

To use an OData Service, it should be designed in such way that it will read/update one or more table. Therefore, we need to import any table, structure or view. These are part of DDIC structure. We can also import Data model from file, RFC/ BOR Interface or a search help. But in this section, we will learn how to import a DDIC structure:

1. Click on the Data Model under the project we have recently created to expand it.
2. Right Click on **Data Model** and click **Import** and then click **DDIC Structure**

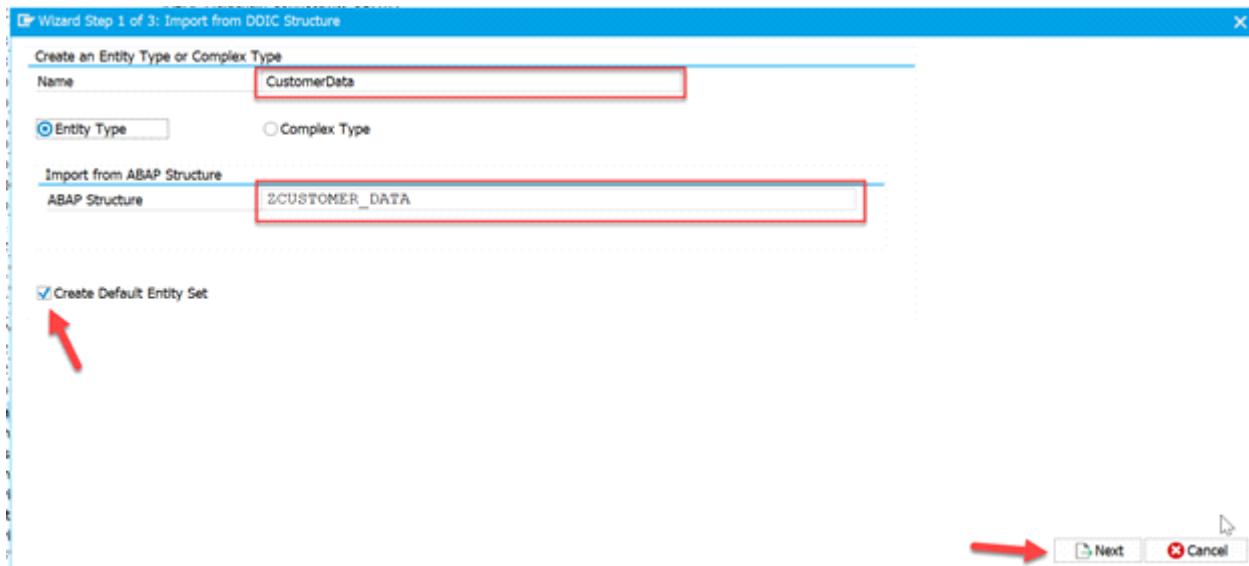




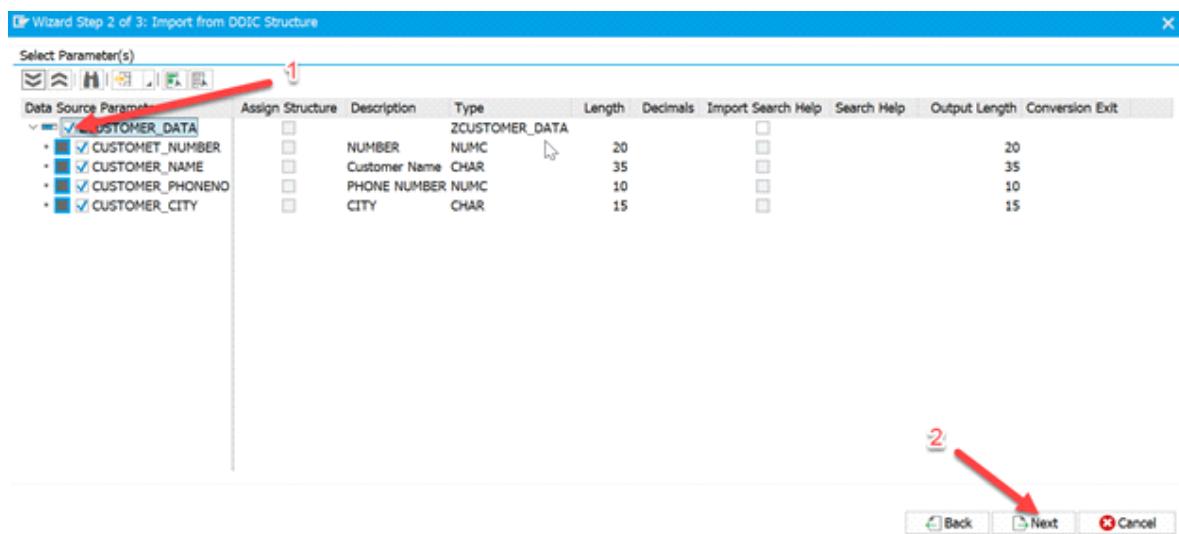
3. Now, a Wizard will open. Enter the following details, there:

Field	Description	Example
Name	Enter a name of Model. It should be explanatory.	CustomerData
Type	We have two options here: 1. Entity Type: This means a simple DDIC structure/table/view will be imported 2. Complex Type: This means a complex DDIC structure/table/view will be imported	Choose Entity Type
ABAP Structure	Choose your required table or structure name. Just copy and paste the name of your DDIC table/structure/view	zcustomer_data
Create Default Entity Set	It will create an Entity Type with the same fields as of Entity. Also, the name will remain same. We recommend you to leave it check marked.	Leave it check marked

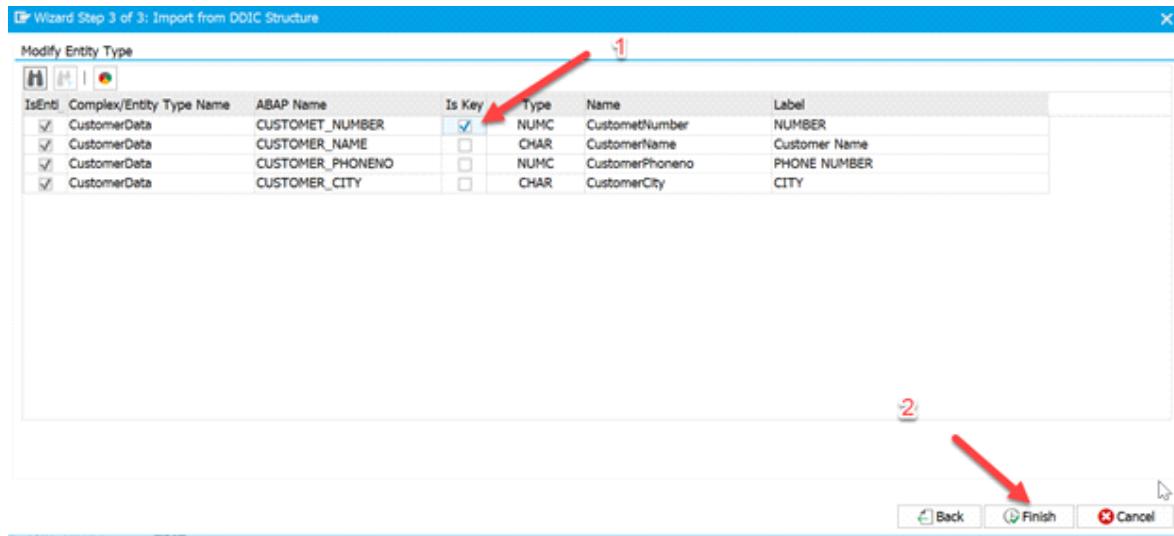
The above details are shown below:



1. In the next step, the wizard will show all the fields of the imported table. Just select them all and click next as shown below:



2. In the next step, the wizard will show all the checked fields. Here you need to check the primary keys and finish, as shown below:

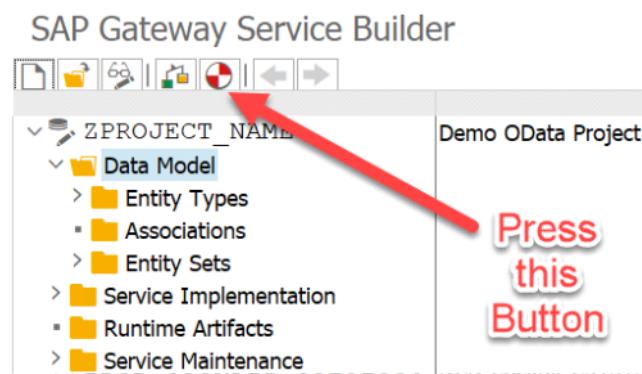


Generate OData Service

Now, we have created an OData object with a model. Now to use this Object we need to generate all the required classes. To generate, press the **Generate Runtime Objects**



button, as shown below:



Note: The Generate Runtime Objects button automatically saves the project before generating classes.

Pressing the above button, will open a dialog box as shown below:

Model and Service Definition

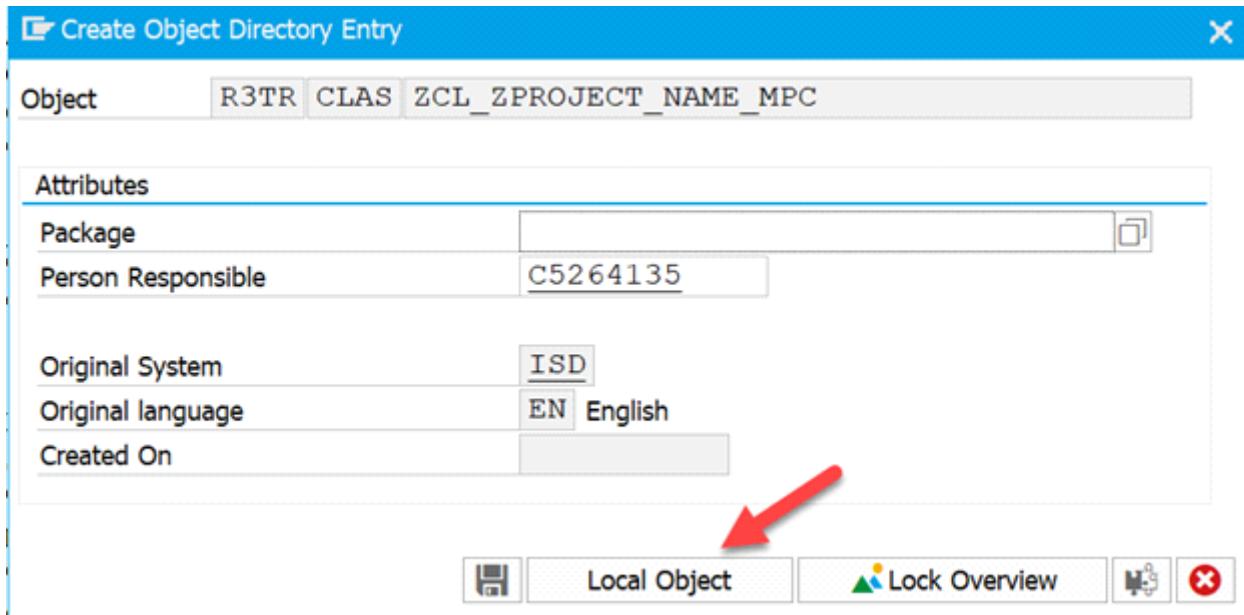
Model Provider Class	
Class Name	ZCL_ZPROJECT_NAME_MPC_EXT
Base Class Name	ZCL_ZPROJECT_NAME_MPC
Data Provider Class	
<input checked="" type="checkbox"/> Generate Classes	
Class Name	ZCL_ZPROJECT_NAME_DPC_EXT
Base Class Name	ZCL_ZPROJECT_NAME_DPC
Service Registration	
Technical Model Name	ZPROJECT_NAME_MDL
Model Version	1
Technical Service Name	ZPROJECT_NAME_SRV
Service Version	1



Just press Continue, as shown above. The above process will generate following classes:

Class Name	Description
MPC Class	MPC stands for Model Provider Class. This is generated and refreshed every time, we generate a class. It is not recommended to extend this class for Model Annotations, because all the methods are deleted whenever the OData is generated.
MPC EXT Class	This is the Extended MPC class, and all the methods are conserved even the OData is generated.
DPC Class	DPC stands for Data Provider Class. This is generated and refreshed every time, we generate a class. It is not recommended to extend this class for CRUD operation, because all the methods are deleted whenever the OData is generated.
DPC EXT Class	This is the Extended DPC class, and all the methods are conserved even the OData is generated.
MDL Class	The is the Model Class.
SRV Class	This is the Service Class.

Just click on Lock Object to generate the classes, as shown below:

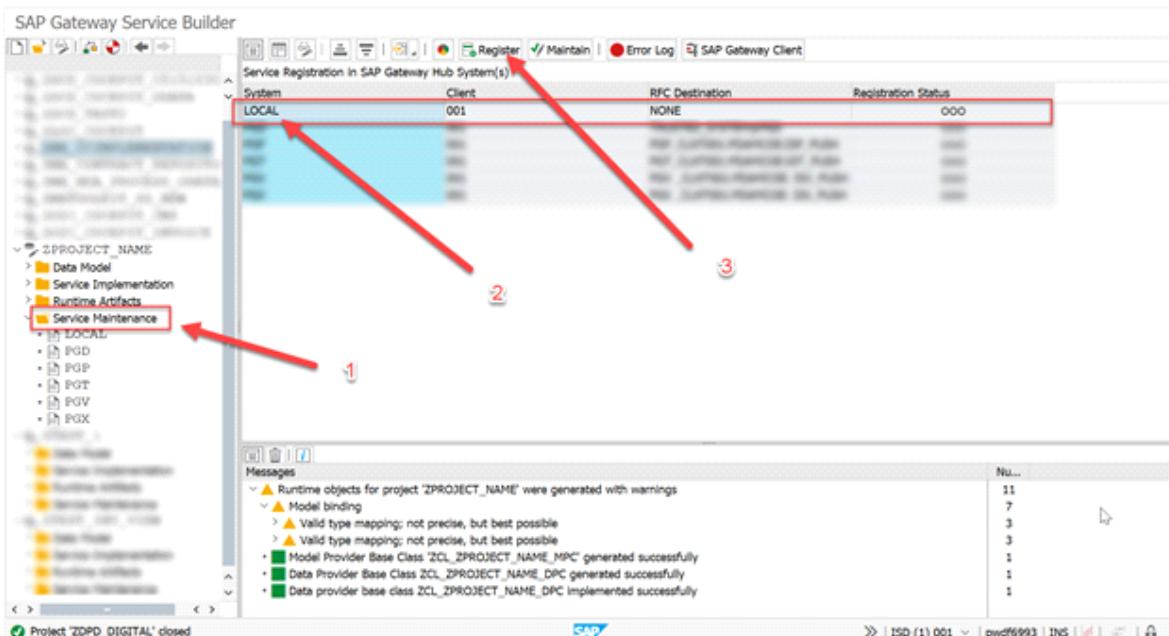


Register SAP OData using the Gateway client

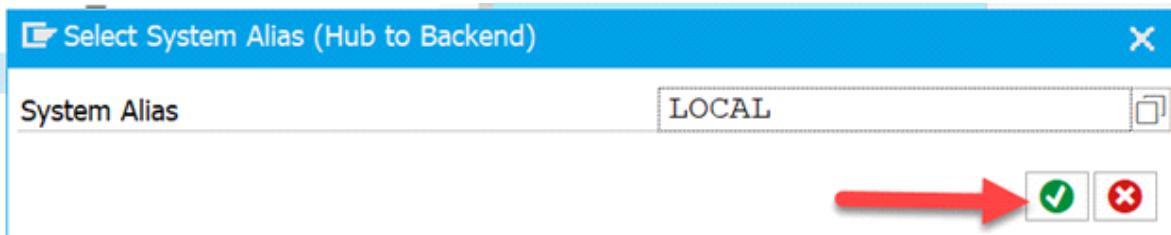
Once the OData is generated, now it is the time to register the service. Once the service is registered, then only it can be used anywhere.

To generate an OData, follow the given steps:

1. Double click the node **Service Maintenance**
2. Select any System and click **Register** as shown below:



3. Enter an Alias, if asked as shown below:



4. Now, in the next screen you get following options:

Field	Description
Technical Service Name	This is auto generated service name
Service Version	It is auto generated service version
Description	It is the Description of the OData you have mention earlier
External Service Name	It is same to the auto generated service name
Namespace	It is by default bank
External Mapping ID	It is by default Empty
External Data Source Type	It is by Default set to source type C
Technical Model Name	It is auto generated Model Name
Model Version	It is auto generated Model Version
Package Assignment	Assign your package name here, or just choose Local Object from the button below. It will be then \$TMP
ICF Node	It is by default "Standard Mode"
oAuth Enablement	It provides option to add extra authentication, in case you need oAuth2 enablement choose it, else leave it blank

In our case, we just need to choose package and click confirm, as shown below:

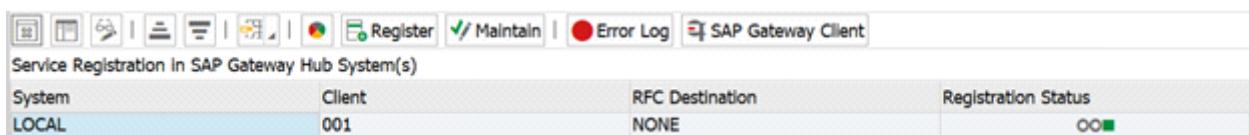
Add Service

Service	
Technical Service Name	ZPROJECT_NAME_SRV
Service Version	1
Description	Demo OData Project
External Service Name	ZPROJECT_NAME_SRV
Namespace	
External Mapping ID	
External Data Source Type	C
Model	
Technical Model Name	ZPROJECT_NAME_MDL
Model Version	1
Creation Information	
Package Assignment	<input type="button" value="Local Object"/> ←
ICF Node	<input checked="" type="radio"/> Standard Mode <input type="radio"/> None
<input checked="" type="checkbox"/> Set Current Client as Default Client in ICF Node	
OAuth enablement	
<input type="checkbox"/> Enable OAuth for Service	

Select Local
 Object or Enter
 your Package
 Name

→ ✓ ✗

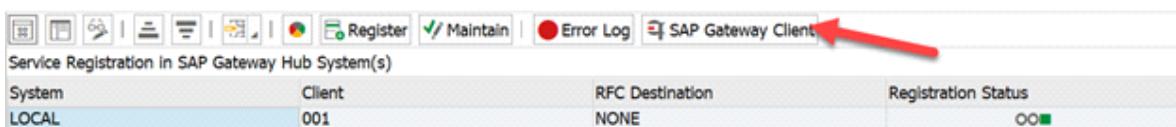
Once the service is registered, the Registration will turn green, as shown below:



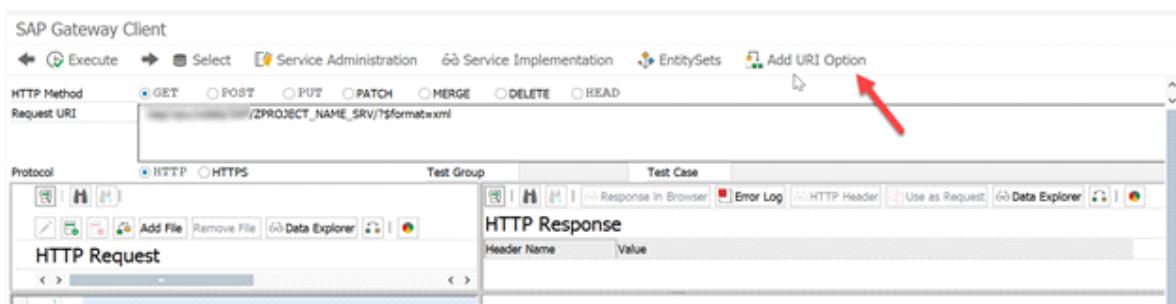
Service Registration in SAP Gateway Hub System(s)			
System	Client	RFC Destination	Registration Status
LOCAL	001	NONE	OO ✓

Test the OData service

- To test the OData service, click the **SAP Gateway Client** button, as shown below:



- It will open a new screen, here click **Add URI Option** as shown below:



3. Choose **\$metadata**, here:

Add URI Option (1) 13 Entries found

URI Option

- \$count
- \$format=json
- \$format=xlsx
- \$inlinecount=allpages
- \$metadata**
- \$metadata?sap-documentation=all
- \$metadata?sap-value-list=all
- \$skip=2
- \$top=2

13 Entries found

4. Click **Execute** as shown below:

SAP Gateway Client

HTTP Method: GET

Request URI: ZPROJECT_NAME_SRV/\$metadata

Protocol: HTTP

HTTP Request

Check this

HTTP Response - Processing Time = 137 ms

Header Name	Value
~status_code	200
~status_reason	OK

```

<?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:DataService m:DataServiceVersion="2.0">
    - <Schema xml:lang="en" xmlns="http://schemas.microsoft.com/ado/2008/09/edmx"
      version="1" Namespace="ZPROJECT_NAME_SRV">
        - <EntityType sap:content-version="1" Name="CustomerData">
          - <Key>
            <PropertyRef Name="CustomerNumber"/>
          </Key>
          <Property Name="CustomerNumber" sap:filterable="false" sap:sortable="false"
            sap:updatable="False" sap:createable="False" sap:label="NUMBER" sap:unicod
            MaxLength="20" Nullable="false" Type="Edm.String"/>
          <Property Name="CustomerName" sap:filterable="false" sap:sortable="false"
            sap:updatable="False" sap:createable="False" sap:label="Customer Name" sa
            MaxLength="35" Nullable="false" Type="Edm.String"/>
          <Property Name="CustomerPhoneno" sap:filterable="false" sap:sortable="false" />
        </EntityType>
      </Schema>
    </edmx:DataService>
  </edmx:Edmx>

```

If the status is 200, as shown above, then it means your OData is working fine.

In the next blog, we will look into how to do CRUD operations in OData

Create a simple OData SAP Gateway Service

Tuesday, March 31, 2020 4:32 PM

- [Home](#)
- [SAP Help](#)
- [Core ABAP](#)
- [ALV](#)
- [Web based](#)
- [Repository](#)
- [Enhancement](#)
- [Comments & Questions](#)
- [Resources](#)
- [≡](#)
- [Comments](#)

Creating your first very simple OData SAP Gateway Service

There is so much information and seemingly new technologies available about developing mobile and responsive applications within SAP at the moment it makes it very difficult to know where to start. So Fiori is the future along with SAPUI5, what about the SAP Gateway and OData, or even HANA, Ariba(or whatever it is called) and the endless other names and technologies around..... How do these all fit into the big picture for an SAP developer?

Well let's start at the ground level i.e. the Netweaver Gateway where traditional SAP ABAP skills are required as part of building a Fiori app. The SAP Gateway has actually been around for many years but with the move to Fiori/Mobile/Responsive apps it is being used more and more and is now a key part of your SAP landscape. The Gateway allows data within your SAP system(s) to be accessed by the outside world via OData services.

...So in this new Fiori world the front end is created using SAPUI5 using the Web IDE,

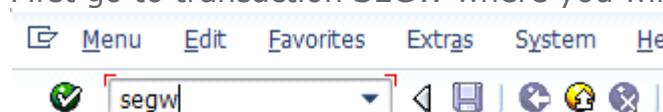
...this calls an OData service that has been created on your SAP system via the SAP Gateway

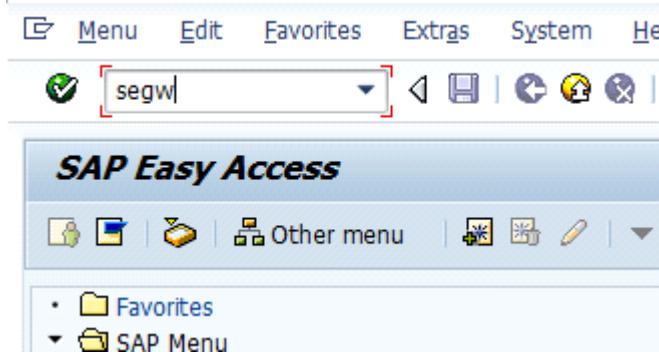
...the OData service uses ABAP to retrieve SAP data and returns it via the OData service

The example below will show you how to quickly create your first OData Gateway service using basic ABAP code to select data from a standard table.

Step 1 – SAP Netweaver Gateway Service Builder

First go to transaction SEGW where you will be able to build your service





Step 2 – Create project

Next using the create button you have to create a project to store all your data models, implementtions, entity types, entity sets etc. Don't worry too much about the terminology at this stage, all will become clear.



Step 3 – Enter project details

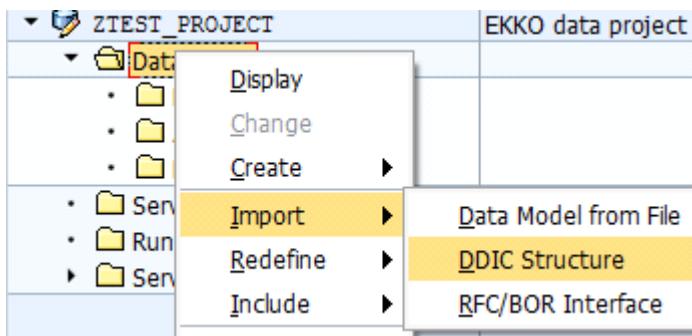
Enter a name, description and package. Leave everything else as default unless you know you need something specific.

The screenshot shows the 'Create Project' dialog box. It has several sections:

- Project:** ZTEST_PROJECT
- Description:** EKKO data project
- Attributes:**
 - Project Type:** Service with SAP Annotations
 - Generation Strategy:** Standard
- Object Directory Entry:**
 - Package:** \$TMP (highlighted with a red box)
 - Person Responsible:**
- Buttons at the bottom:** A green checkmark icon, a 'Local Object' button, and a red X icon.

Step 4 – Import DDIC structure

We are going to base this example on a subset of the standard SAP table EKKO. Within the newly created project, right click on the "Data Model" node and select Import->DDIC Structure



Step 5 – Enter structure details

Enter EKKO within the ABAP structure field and enter an object a name i.e. purchaseorder

Field Path	Ref...	Type	Usage	Name
MANDT	<input type="checkbox"/>	Edm.String	Key	Mandt
EBELN	<input type="checkbox"/>	Edm.String	Key	Ebeln
BUKRS	<input type="checkbox"/>	Edm.String	Property	Bukrs
BSTYP	<input type="checkbox"/>	Edm.String	Property	Bstyp
BSART	<input type="checkbox"/>	Edm.String	Property	Bsart
AEDAT	<input type="checkbox"/>	Edm.DateTime		

Step 6 – View Properties

Expand the newly created node "Entity Type->purchaseorder" and double click the Properties node

Name	Key	Edm Type	Prec.	Scale	Max.	Unit Prop.	Cre...	Up...	Sor...	Nul...
Mandt	<input checked="" type="checkbox"/>	Edm.String	0	0	3					
Ebeln	<input checked="" type="checkbox"/>	Edm.String	0	0	10					
Bukrs	<input type="checkbox"/>	Edm.String	0	0	4					
Bstyp	<input type="checkbox"/>	Edm.String	0	0	1					
Bsart	<input type="checkbox"/>	Edm.String	0	0	4					
Bsakz	<input type="checkbox"/>	Edm.String	0	0	1					
Loekz	<input type="checkbox"/>	Edm.String	0	0	1					
Statu	<input type="checkbox"/>	Edm.String	0	0	1					
Aedat	<input type="checkbox"/>	Edm.DateTime	0	0	0					

We are just going to use the top few fields so select all the fields below Statu and delete them

Name	Key	Edm Type	Prec.	Scale	Max.	Unit Prop.	Cre...	Up...	Sor...	Nul...
Mandt	<input checked="" type="checkbox"/>	Edm.String	0	0	3					
Ebeln	<input checked="" type="checkbox"/>	Edm.String	0	0	10					
Bukrs	<input type="checkbox"/>	Edm.String	0	0	4					
Bstyp	<input type="checkbox"/>	Edm.String	0	0	1					
Bsart	<input type="checkbox"/>	Edm.String	0	0	4					
Bsakz	<input type="checkbox"/>	Edm.String	0	0	1					
Loekz	<input type="checkbox"/>	Edm.String	0	0	1					
Statu	<input type="checkbox"/>	Edm.String	0	0	1					
Aedat	<input type="checkbox"/>	Edm.DateTime	0	0	0					
Emam	<input type="checkbox"/>	Edm.String	0	0	12					
Pinr	<input type="checkbox"/>	Edm.String	0	0	5					
Lponr	<input type="checkbox"/>	Edm.String	0	0	5					
Ifnr	<input type="checkbox"/>	Edm.String	0	0	10					

Until it looks like this

Name	Key	Edm Core Type	Prec.	Scale	Max...	Unit Prop.	Cre...	Up...	Sor...	Null...	Filt...	Lab...
Mandt	<input checked="" type="checkbox"/>	Edm.String	0	0	3		<input type="checkbox"/>					
Ebeln	<input checked="" type="checkbox"/>	Edm.String	0	0	10		<input type="checkbox"/>					
Bukrs	<input type="checkbox"/>	Edm.String	0	0	4		<input type="checkbox"/>					
Bstyp	<input type="checkbox"/>	Edm.String	0	0	1		<input type="checkbox"/>					
Bsart	<input type="checkbox"/>	Edm.String	0	0	4		<input type="checkbox"/>					
Bsakz	<input type="checkbox"/>	Edm.String	0	0	1		<input type="checkbox"/>					
Loekz	<input type="checkbox"/>	Edm.String	0	0	1		<input type="checkbox"/>					
Statu	<input type="checkbox"/>	Edm.String	0	0	1		<input type="checkbox"/>					

Note: The reason I'm not going to use all the fields is because some are incompatible with a gateway service without changing the data type. I will show you where you would get the error further down if you had used all the fields. This info might just help you understand the error quicker when you are creating one for real in the future.

Step 7 – Create Entity Set

Save your process so far and then right click on the Entity Set node and select Create

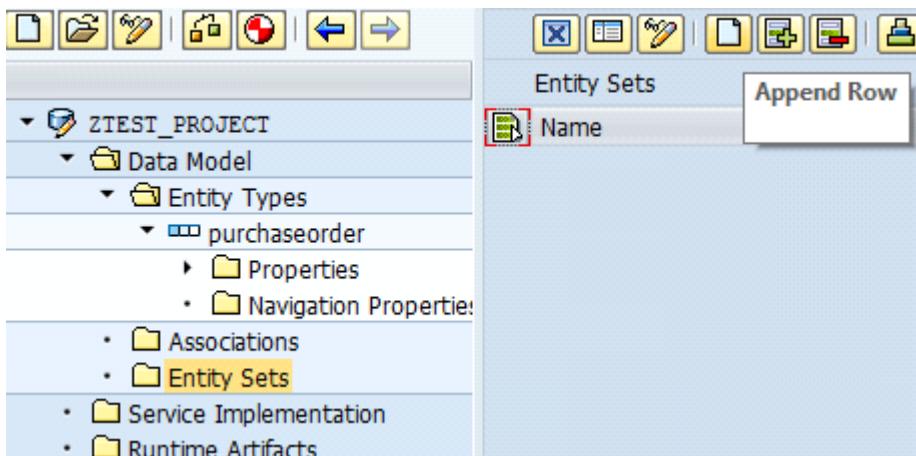
The screenshot shows the SAP Studio interface with the following details:

- Project Tree:** Shows 'ZTEST_PROJECT' expanded, with 'Data Model' and 'Entity Types' nodes.
- Entity Types Tree:** Shows 'purchaseorder' expanded, with 'Properties' node.
- Properties View:** A table showing entity type properties:

Name	Key	Edm Core Type
Mandt	<input checked="" type="checkbox"/>	Edm.String
Ebeln	<input type="checkbox"/>	Edm.String
Bukrs	<input type="checkbox"/>	Edm.String
Bstyp	<input type="checkbox"/>	Edm.String
Bsart	<input type="checkbox"/>	Edm.String
Bsakz	<input type="checkbox"/>	Edm.String
Loekz	<input type="checkbox"/>	Edm.String
Statu	<input type="checkbox"/>	Edm.String
- Context Menu:** Opened at the 'Entity Sets' node, showing options: Display, Change, Create, Copy, Paste.

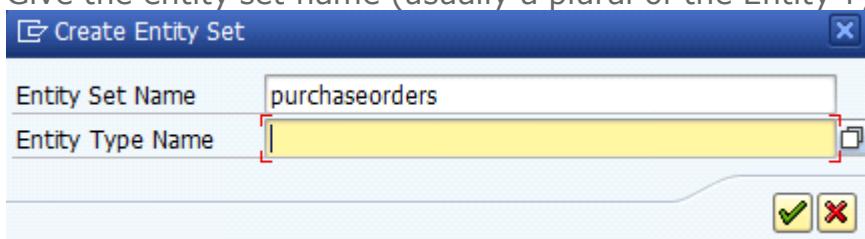
Step 8 – Alternate way to create Entity Set

Alternatively double click the Entity Set node and then click the Append Row button

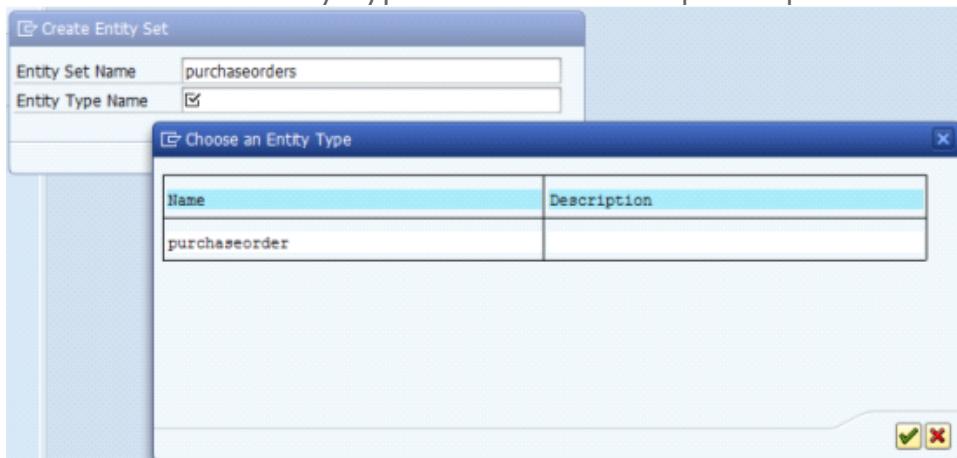


Step 9 – Entity set details

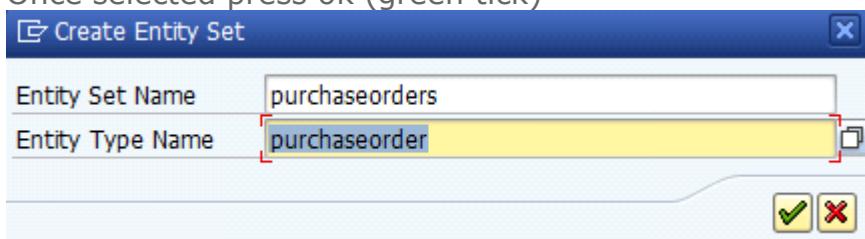
Give the entity set name (usually a plural of the Entity Type)



Then select the Entity Type from selection input help



Once selected press ok (green tick)



Entity set has now been created

The screenshot shows the SAP NetWeaver Gateway Service Builder interface. On the left, the project tree shows a node 'ZTEST_PROJECT' under 'Data Model / Entity Types / purchaseorder / Entity Sets'. On the right, a table titled 'Entity Sets' lists one item: 'purchaseorders' with 'purchaseorder' as the 'Entity Type Name'.

Name	Entity Type Name
purchaseorders	purchaseorder

Step 10 – Generate gateway service

First double click the project node and generate the whole project using the "Generate Runtime Objects" button

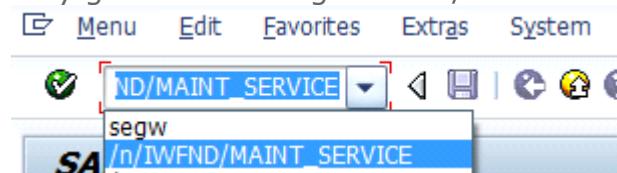
The screenshot shows the 'SAP NetWeaver Gateway Service Builder' window. A dialog box titled 'Generate Runtime Objects' is open, covering part of the project tree. The tree shows 'ZMLPROJ' and 'ZTEST_PROJECT' (selected), both under 'Data Model / Entity Types / purchaseorder / Entity Sets'. The 'Generate Runtime Objects' dialog has a 'Run' button at the bottom.

You will now notice that Service Implementation Objects have been created. Create, Delete, GetEntity(Read), GetEntitySet(Query) and Update

The screenshot shows the expanded project tree. Under 'ZTEST_PROJECT / Data Model / Entity Types / purchaseorder / Entity Sets', the 'purchaseorders' node is selected. Under 'Service Implementation', there is a node 'purchaseorders' which contains five methods: 'Create', 'Delete', 'GetEntity (Read)', 'GetEntitySet (Query)', and 'Update'. Other nodes like 'Runtime Artifacts' and 'Service Maintenance' are also visible.

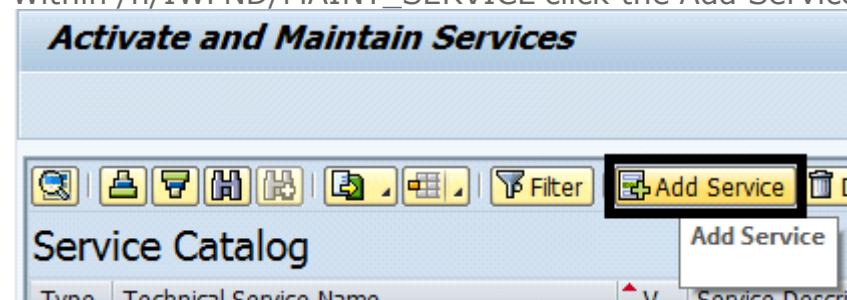
Step 11 – Active and maintain service

Now you need to go to transaction "/IWFND/MAINT_SERVICE" I always find I can only get this working if i add /n at the start i.e. /n/IWFND/MAINT_SERVICE

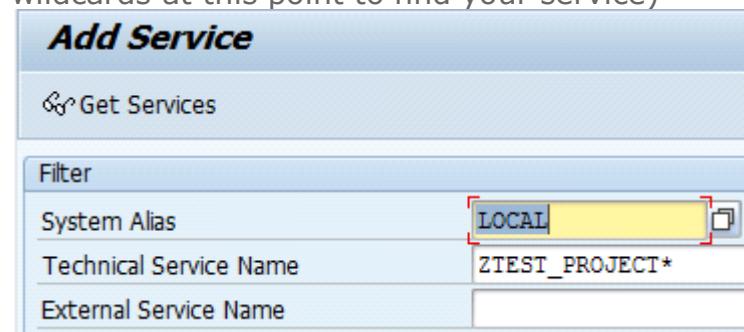


Step 12 – Add service

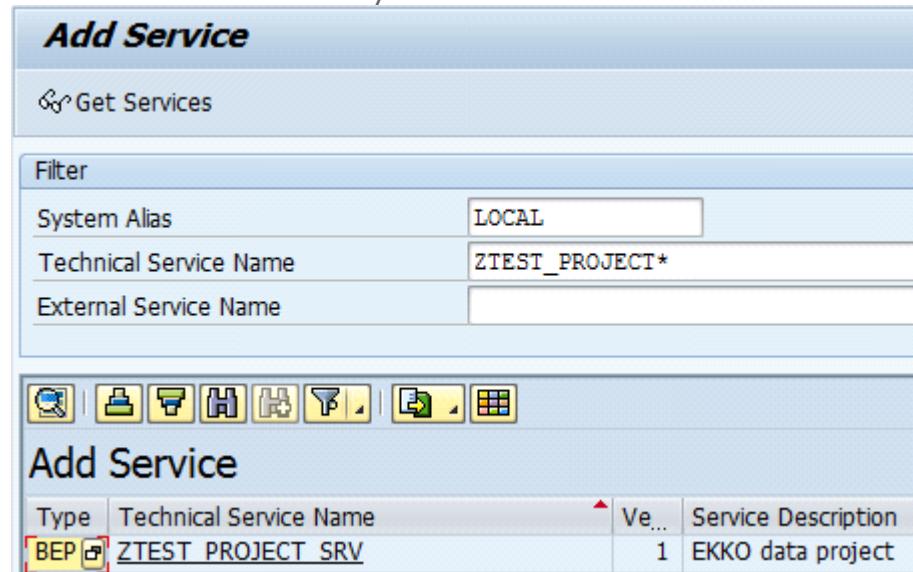
Within /n/IWFND/MAINT_SERVICE click the Add Service button



Then enter the information of the service you want to add (Notice you can use wildcards at this point to find your service)

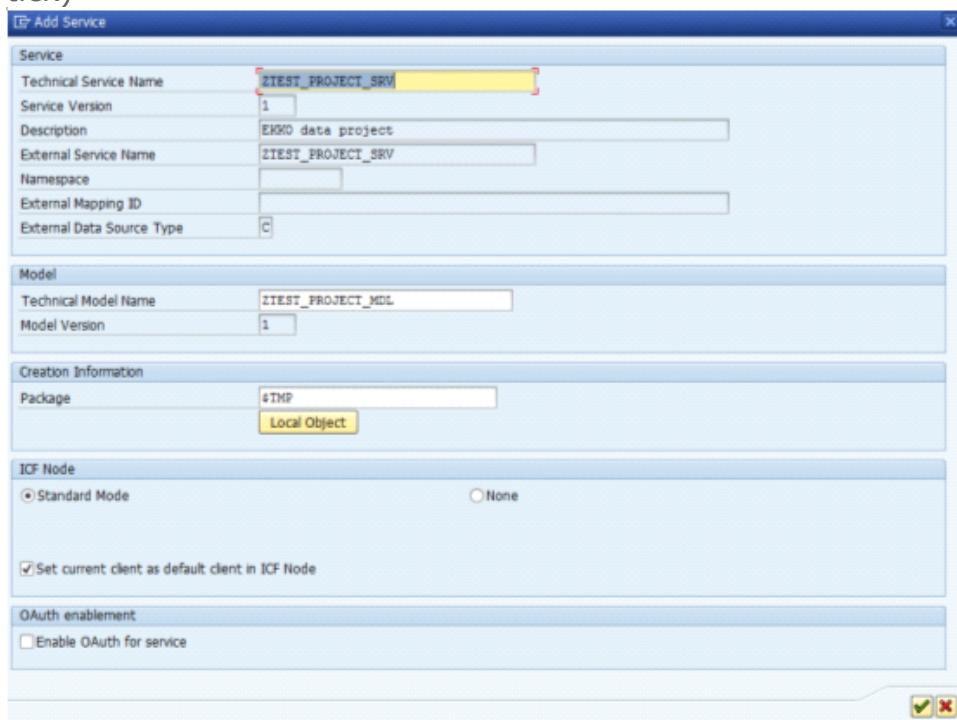


Press enter to find your service or services that match your search criteria. Once it does click on the one you want to add

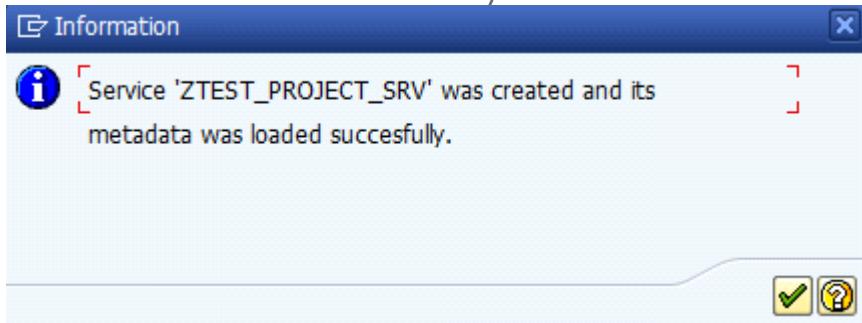


The next screen shows you the selected service details, enter package details(i.e.

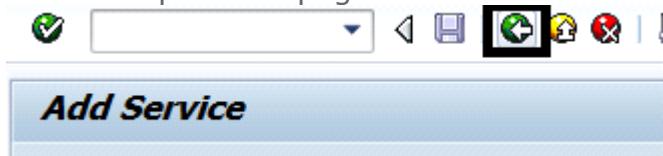
local Object) and leave everything as default. Then press the ok button (green tick)



You should now receive a message popup that your service "was created and its metadata was loaded successfully"

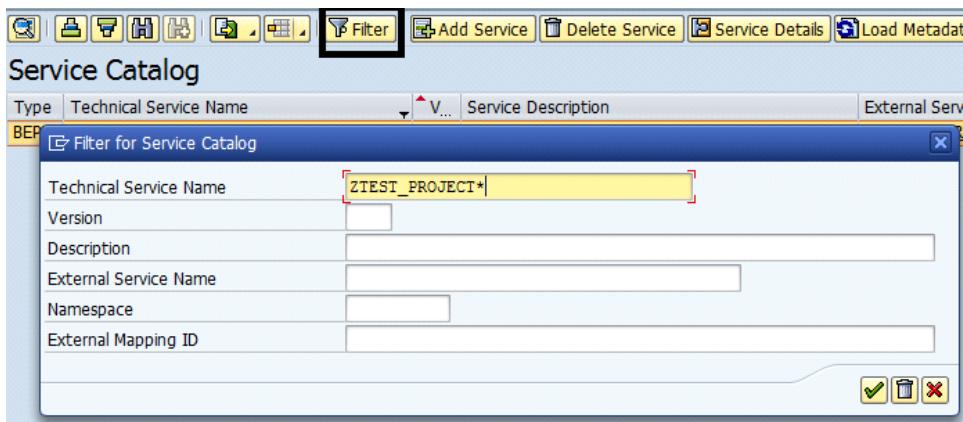


Return to previous page



Step 13 – Find your added service

You will now be returned to the service catalog, depending on how many services you have setup in your system you may need to use the filter functionality to find your newly added service.



You should now see the service setup details and a green traffic light next to the ODATA ICF node in the bottom left hand corner.

Step 14 – SAP Netweaver Gateway Client

We now need to test it using the SAP Netweaver Gateway Client, which is accessed via the "Gateway Client" button just above the ODATA node in the bottom left hand corner

Then, leaving the Request URI as default

"/sap/opu/odata/sap/ZTEST_PROJECT_SRV/?\$format=xml" simply click the "Execute" button

You should then get a HTTP response similar to this with a green status_code

SAP Netweaver Gateway Client

HTTP Method: GET

Request URI: /sap/opu/odata/sap/ZTEST_PROJECT_SRV/\$metadata

Protocol: HTTP

Test Group: Test Case

Add URL Option

HTTP Request

HTTP Response

Header Name	Value
~status_code	200
~status_reason	OK

```
<?xml version="1.0" encoding="utf-8" ?>
- <app:service xml:lang="en" xml:base="http://.../sap/opu/odata/sap/ZTEST_PROJECT_SRV/">
  <atom:link href="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom">
    <app:workspace>
      <atom:title type="text">Data</atom:title>
      <app:collection sap:content-version="1" href="purchaseorders">
        <atom:link href="http://www.sap.com/Protocols/SAPData">
          <atom:title type="text">purchaseorders</atom:title>
          <atom:member-title>purchaseorder</atom:member-title>
        </app:collection>
      </app:workspace>
      <atom:link rel="self" href="http://.../sap/opu/odata/sap/ZTEST_PROJECT_SRV/" />
      <atom:link rel="latest-version" href="http://.../sap/opu/odata/sap/ZTEST_PROJECT_SRV/" />
    </app:service>
```

SAP Netweaver Gateway Client

HTTP Response

Header Name	Value
~status_code	200
~status_reason	OK
~server_protocol	HTTP/1.0
set-cookie	sap-usercontext=sap-language=DE&sap-client=800; path=/

```
<?xml version="1.0" encoding="utf-8" ?>
- <app:service xml:lang="en" xml:base="http://.../sap/opu/odata/sap/ZTEST_PROJECT_SRV/">
  <atom:link href="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom">
    <app:workspace>
      <atom:title type="text">Data</atom:title>
      <app:collection sap:content-version="1" href="purchaseorders">
        <atom:link href="http://www.sap.com/Protocols/SAPData">
          <atom:title type="text">purchaseorders</atom:title>
          <atom:member-title>purchaseorder</atom:member-title>
        </app:collection>
      </app:workspace>
      <atom:link rel="self" href="http://.../sap/opu/odata/sap/ZTEST_PROJECT_SRV/" />
      <atom:link rel="latest-version" href="http://.../sap/opu/odata/sap/ZTEST_PROJECT_SRV/" />
    </app:service>
```

Step 15 – Further tests via SAP Netweaver Gateway Client

Now you can modify the URL so that it ends with "\$metadata?sap-ds-debug=true" and then press execute again, so that we can return metadata properties of the purchaseorder entity

HTTP Method: GET

Request URI: /sap/opu/odata/sap/ZTEST_PROJECT_SRV/\$metadata?sap-ds-debug=true

Protocol: HTTP

Test Group: 1

..... "/sap/opu/odata/sap/ZTEST_PROJECT_SRV/\$metadata?sap-ds-debug=true"

HTTP Response

Header Name	Value
~server_protocol	HTTP/1.0
set-cookie	sap-usercontext=sap-language=E&sap-client=800; path=/
set-cookie	SAP_SESSIONID_TST_800=203khWptZh1K0X4fgzz5erZ9r9-cBHznPQ5UrDxA;
content-type	text/html; charset=utf-8
content-length	119933
last-modified	Mon, 19 Sep 2016 13:47:09 GMT

Body

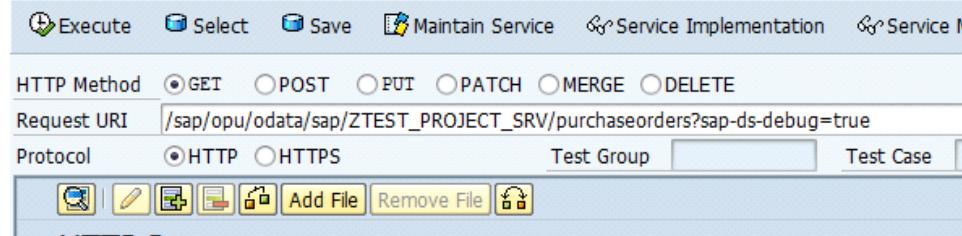
```

<edmx:Edmx xmlns="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  <edmx:DataServices xmlns="http://schemas.microsoft.com/ado/2007/06/dataservices/metadata" m:DataServiceVersion="2.0">
    <Schema xmlns="en" xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="ZTEST_PROJECT_SRV">
      <EntityType xmlns="http://www.sap.com/Protocols/SAPData" sap:content-version="1" Name="purchaseorder">
        <Key>
          <PropertyRef Name="Mandt"/>
          <PropertyRef Name="Ebeln"/>
        </Key>
        <Property Name="Mandt" sap:label="Client" MaxLength="3" Nullable="false" Type="Edm.String"/>
        <Property Name="Ebeln" sap:label="Purchasing Doc." MaxLength="10" Nullable="false" Type="Edm.String"/>
        <Property Name="Bukrs" sap:label="Company Code" MaxLength="4" Nullable="false" Type="Edm.String"/>
        <Property Name="Bstyp" sap:label="Doc. Category" MaxLength="4" Nullable="false" Type="Edm.String"/>
        <Property Name="Bstart" sap:label="Document Type" MaxLength="4" Nullable="false" Type="Edm.String"/>
        <Property Name="Bzkt" sap:label="Control" MaxLength="1" Nullable="false" Type="Edm.String"/>
        <Property Name="Loek1" sap:label="Deletion Ind." MaxLength="1" Nullable="false" Type="Edm.String"/>
        <Property Name="Status" sap:label="Status" MaxLength="1" Nullable="false" Type="Edm.String"/>
      </EntityType>
      <EntityContainer Name="ZTEST_PROJECT_SRV" m:isDefaultEntityContainer="true">
        <EntitySet xmlns:sap="http://www.sap.com/Protocols/SAPData" sap:content-version="1" Name="purchaseorders">
          <EntityType>ZTEST_PROJECT_SRV.purchaseorder</EntityType>
        </EntityContainer>
        <atomlink xmlns:atom="http://www.w3.org/2005/Atom" href="http://.../odata/sap/ZTEST_PROJECT_SRV/purchaseorders" rel="latest-version"/>
      </Schema>
    </edmx:DataServices>
</edmx:Edmx>

```

Now change the URI to have "/purchaseorders?sap-ds-debug=true" at the end so that we can target the data of the entity set purchaseorders

...."/sap/opu/odata/sap/ZTEST_PROJECT3_SRV/purchaseorders?sap-ds-debug=true"



When you press execute you should get the following error as we have not implemented the method that populates this data yet

HTTP Response

Header Name	Value
~server_protocol	HTTP/1.0
set-cookie	sap-usercontext=sap-language=E&sap-client=800; path=/

Body

```

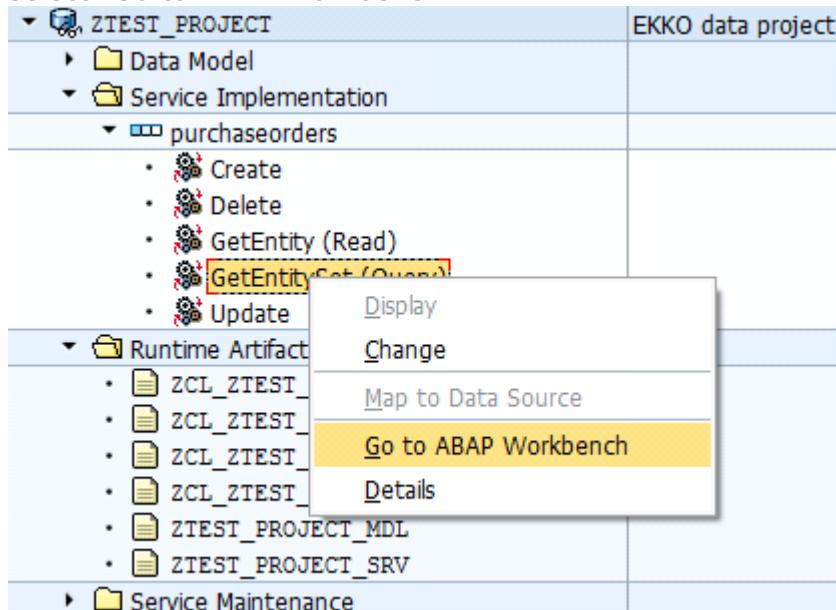
<error xmlns="http://schemas.microsoft.com/ado/2007/06/dataservices/metadata">
  <code>SY/530</code>
  <message>
    <xml:lang=en>Method 'PURCHASEORDERS_GET_ENTITYSET' not implemented in data provider class.</message>
  </innererror>
  <transactionid>124D7EE63F72F107ACFF50E549C2C40B</transactionid>
  <errordetails>
    <errordetail>
      <code>/IMBEP/CX_MGW_NOT_IMPL_EXC</code>
      <message>Method 'PURCHASEORDERS_GET_ENTITYSET' not implemented in data provider class.</message>
      <propertyref/>
      <severity>error</severity>
    </errordetail>
  </errordetails>
</innererror>
</error>

```

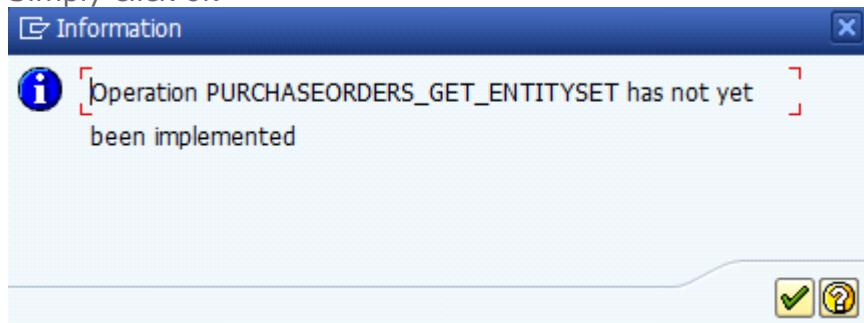
Step 16 – Implement GetEntitySet method

Now return to the SEGW transaction and find the service implementation methods created before. Find the one called GetEntitySet(Query) and right click on it. Then

```
select "Go to ABAP Workbench"
```

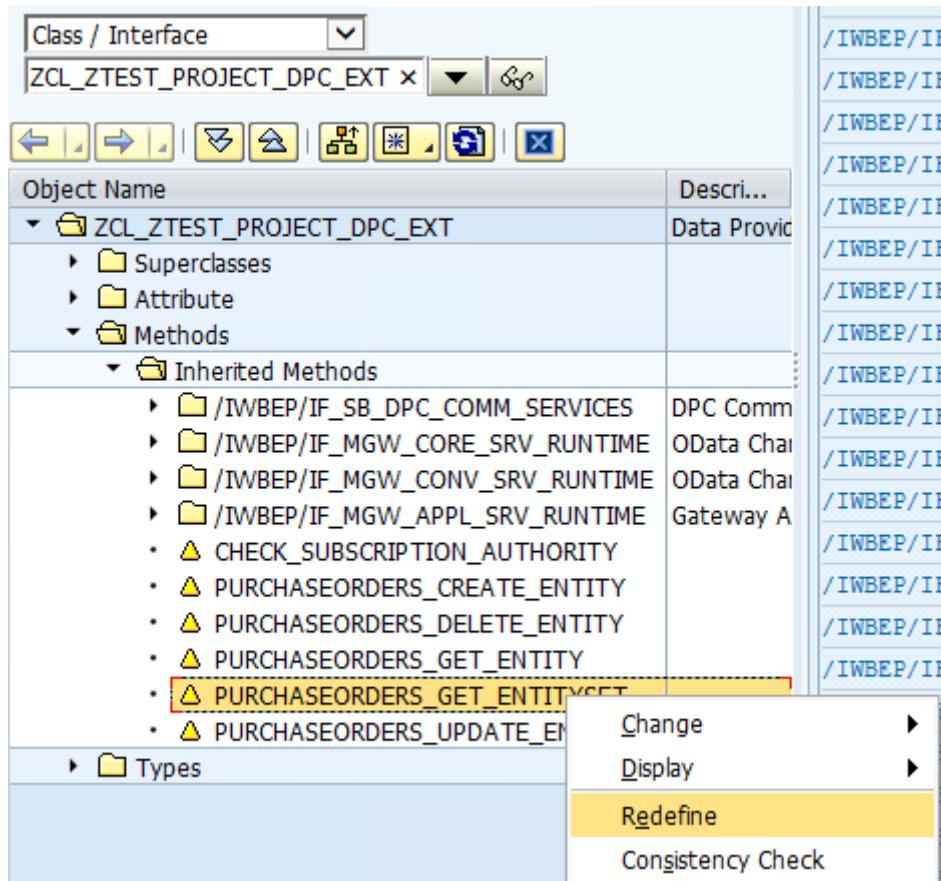


The following message will appear saying operation has not yet been implemented.
Simply click ok



You will now be taken to tcode SE80. Within the object list on the left find the PURCHASEORDERS_GET_ENTITYSET method within Methods->Inherited Methods.

Once you have found this, right click on it and select Redefine



Within the method code that appears simply add the following ABAP select statement

```
SELECT *
UP TO 10 ROWS
FROM ekko
INTO CORRESPONDING FIELDS OF TABLE et_entityset.
```

Method PURCHASEORDERS_GET_ENTITYSET Inactive

```

1  method PURCHASEORDERS_GET_ENTITYSET.
2  **TRY.
3  *CALL METHOD SUPER->PURCHASEORDERS_GET_ENTITYSET
4  * EXPORTING
5  *   IV_ENTITY_NAME      =
6  *   IV_ENTITY_SET_NAME   =
7  *   IV_SOURCE_NAME       =
8  *   IT_FILTER_SELECT_OPTIONS =
9  *   IS_PAGING            =
10 *    IT_KEY_TAB          =
11 *    IT_NAVIGATION_PATH   =
12 *    IT_ORDER              =
13 *    IV_FILTER_STRING     =
14 *    IV_SEARCH_STRING     =
15 **    io_tech_request_context  =
16 ** IMPORTING
17 **    et_entityset         =
18 **    es_response_context   =
19 *
20 ** CATCH /iwbep/cx_mgw_busi_exception .
21 ** CATCH /iwbep/cx_mgw_tech_exception .
22 **ENDTRY.
23  SELECT *
24  up to 10 rows
25  from ekko
26  into corresponding fields of table et_entityset.
27
28 endmethod.
```

Save and Activate

Class Builder: Class ZCL_ZTEST_PROJECT_DPC_EXT Change

D.	Object	Obj. name
CPRO	ZCL_ZTEST_PROJECT_DPC_EXT	ZCL_ZTEST_PROJECT_DPC_EXT
CPUB	ZCL_ZTEST_PROJECT_DPC_EXT	ZCL_ZTEST_PROJECT_DPC_EXT
METH	ZCL_ZTEST_PROJECT_DPC_EXT	PURCHASEORDERS_GET_ENTITYSET

Step 17 – Re-test the service

Again change the URI to "/sap/opu/odata/sap/ZTEST_PROJECT_SRV/purchaseorders?sap-ds-debug=true"

Execute Select Save Maintain Service Service Implementation Service M

HTTP Method: GET POST PUT PATCH MERGE DELETE

Request URI: /sap/opu/odata/sap/ZTEST_PROJECT_SRV/purchaseorders?sap-ds-debug=true

Protocol: HTTP HTTPS Test Group: Test Case:

Add File Remove File

This time you should get some data returned

Body Request Response Server URI Runtime



```

<feed xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
      xmlns="http://www.w3.org/2005/Atom" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
      xml:base="http://.../sap/opu/odata/sap/ZTEST_PROJECT3_SRV/"
      <id>http://.../sap/opu/odata/sap/ZTEST_PROJECT3_SRV/purchaseorders</id>
      <title type="text">purchaseorders</title>
      <updated>2016-09-19T14:24:59Z</updated>
      - <author>
          <name/>
      </author>
      <link title="purchaseorders" href="purchaseorders" rel="self"/>
      - <entry>
          <id>http://.../sap/opu/odata/sap/ZTEST_PROJECT3_SRV/purchaseorders(Mandt='800',Ebeln='3000000004')</id>
          <title type="text">purchaseorders(Mandt='800',Ebeln='3000000004')</title>
          <updated>2016-09-19T14:24:59Z</updated>
          <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"
                     term="ZTEST_PROJECT3_SRV.Ekko"/>
          <link title="Ekko" href="purchaseorders(Mandt='800',Ebeln='3000000004')" rel="edit"/>
          - <content type="application/xml">
              - <m:properties>
                  <d:Mandt>800</d:Mandt>
                  <d:Ebeln>3000000004</d:Ebeln>
                  <d:Bukrs>3000</d:Bukrs>
                  <d:Bstyp>F</d:Bstyp>
                  <d:Bsart>EC</d:Bsart>
                  <d:Bsakz/>
                  <d:Loekz/>
                  <d:Statu>I</d:Statu>
              </m:properties>
          </content>
      </entry>
  </feed>

```

Step 18 – Further info

Remember I mentioned about not using all the fields from the EKKO table as some are not compatible. If you had included them all when executing the URI "/sap/opu/odata/sap/ZTEST_PROJECT_SRV/purchaseorders?sap-ds-debug=true" you would have received the following http error message:

Response in Browser Error Log Use as Request

HTTP Response

Header Name	Value
~status_code	500
~status_reason	INTERNAL SERVER ERROR

```

<?xml version="1.0" encoding="utf-8" ?>
- <error xmlns:xsi="http://www.w3.org/2001/XMLSchema-Instance">
    <code>HTTP/500/E/Internal Server Error</code>
    <message>Call of service /sap/opu/odata/sap/ZTEST_PROJECT_SRV/purchaseorders terminated because of an error. The following error text was processed in system TST : The current statement is only supported for character-type data objects.The error occurred on the application server erpukpltm. The termination type was: RABAX_STATE.If the termination type is RABAX_STATE, you will find more information on the cause of termination in system TST in transaction ST22. If the termination type is ABORT_MESSAGE_STATE, you will find more information on the cause of termination on the application server erpukpltm in transaction SM21. If the termination type is ERROR_MESSAGE_STATE, you can search for further information in the trace file for the work process in transaction ST11 on the application server erpukpltm. You may also need to analyze the trace files of other work processes. If you do not yet have a user ID, contact your system administrator.</message>
</error>

```

```

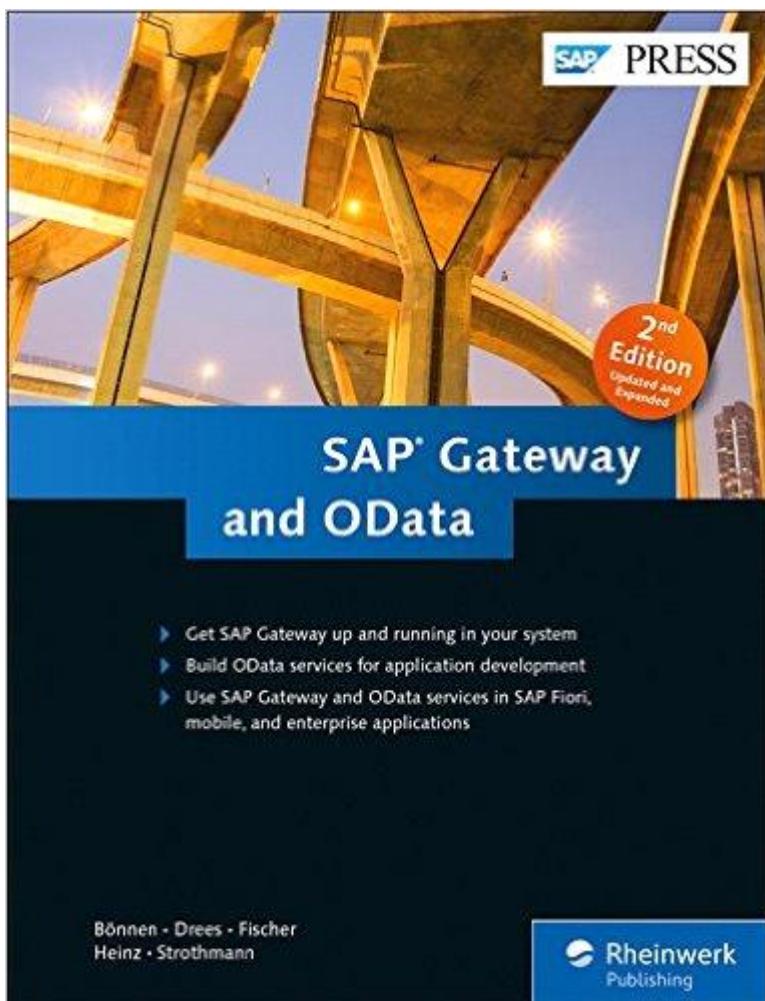
<?xml version="1.0" encoding="utf-8" ?>
- <error xmlns:xsi="http://www.w3.org/2001/XMLSchema-Instance">
    <code>HTTP/500/E/Internal Server Error
    <message>Call of service /sap/opu/odata/sap/ZTEST_PROJECT_SRV/purchaseorders terminated because of an error. The following error text was processed in system TST : The current statement is only supported for character-type data objects.The error occurred on the application server erpukpltm. The termination type was: RABAX_STATE.If the termination type is RABAX_STATE, you will find more information on the cause of termination in system TST in transaction ST22. If the termination type is ABORT_MESSAGE_STATE, you will find more information on the cause of termination on the application server _____ in transaction SM21. If the termination type is ERROR_MESSAGE_STATE, you can search for further information in the trace file for the work process in transaction ST11 on the application server _____. You may also need to analyze the trace files of other work processes. If you do not yet have a user ID, contact your system administrator.
    </message>
</error>

```

Step 19 – Further development

The next step is to access this gateway service from your SAP Fiori App

[Learn More↓](#)



[Beginners Guide](#) [ABAP tips](#) [SAP Reporting](#) [Send Emails](#) [ALV Reports](#) [Repository Objects](#) [Web Applications](#) [ABAP statements](#)



AddThis Sharing Buttons

- [Share to Facebook](#)
- [Share to Twitter](#)
- [Share to Print](#)
- [Share to Email](#)
- [Share to Pinterest](#)
- [Share to Gmail](#)

[Share to LinkedIn](#)
[Share to Email App](#)
[Share to Tumblr](#)
[Share to More](#)

Can't find something on ERPWorkbench? A quick search should fix that:

Follow

[Follow on Facebook](#)[Follow on Twitter](#)

[**Contact Us**](#) [**Partners**](#) [**Terms of Service**](#) [**Privacy Policy**](#) [**Advertise**](#)

AddThis Sharing Sidebar

Share to Facebook

, Number of shares

[Share to Twitter](#)[Share to Print](#)[Share to Email](#)[More AddThis Share options](#)

, Number of shares

Hide

Show

Close

[AddThis](#)

AddThis Sharing

Follow

Hide

Follow

[Follow on Facebook](#)

[Follow on Twitter](#)

Show

Inserted from <<https://www.erpworkbench.com/sap-webapps/segw-odata-gateway-service.htm>>

SAP Fiori LL11 – Consultants should know about OData troubleshooting | SAP Blogs

Friday, February 19, 2021 11:52 AM

Clipped from: <https://blogs.sap.com/2013/11/13/sap-fiori-ll11-consultants-should-know-about-odata-troubleshooting/>

SAP Fiori Lessons Learned 11

Consultants should know about OData troubleshooting

Background:

SAP Fiori apps are out of the solution but you face several issues because main functions are in backend ERP and each customer has different backend setup/configuration. OData is the communication channel to the backend servers via get/post/put method.

Lessons Learned:

Understanding OData and backend function are must requirements for consultants.

- Application consultants should learn about OData technical and runtime behavior
- Technical consultants should learn about ERP application functions
- Both application consultants and technical consultants should work together

Consultants should know following steps.

Step 1. Identity the OData service call

1.1 Run the app

1.2 Inspect element

Right click -> Inspect element (Chrome), F12 (IE), then go to Network tab

The screenshot shows a SAP Fiori application titled "Approve Purchase Orders". On the left, a list of "Purchase Orders (1)" is displayed, showing one item: "C.E.B. BERLIN" with order number "1". On the right, the details for this purchase order are shown: "C.E.B. BERLIN", "140.00 EUR", and "77 days ago". Below the main details, there is a note: "Forwarded by WF-MM-4". A context menu is open over the order number "1" in the list, with the "Inspect element" option highlighted by a red circle. Other options in the menu include Back, Forward, Reload, Save as..., Print..., Translate to English, View page source, View page info, and Inspect element. At the bottom of the screen, there is a toolbar with icons for gear, checkmark, X, and a square.

Ignore the warning or error message in the Console tab at this point in time.
Focus on OData troubleshooting.

1.3 Refresh the app or browser

The screenshot shows the same SAP Fiori application as the previous one. The "REFRESH" button on the left has been circled in red. The developer tools toolbar at the bottom has tabs for Elements, Resources, Network, Sources, Timeline, Profiles, Audits, and Console. The Network tab is currently selected and has been circled in red. A message in the console area says: "⚠ No requests captured. Reload the page to see detailed information on the network activity." The developer tools toolbar also includes icons for document, style sheet, image, script, XHR, font, and a gear icon.

1.4 Find the last called OData URI

Scroll down to the bottom of inspect element and find the OData /sap/opu/odata/sap/.. from bottom.

The screenshot shows the SAP Fiori Launchpad interface with the Network tab selected. A list of requests is displayed, and a context menu is open over the first item. The 'Copy link address' option in the menu is circled in red.

[http://<host>.<domain>:<port>/sap/opu/odata/sap/GBAPP_POAPPROVAL;mo/WorkflowTaskCollection\(SAP__Origin='ONE_800',WorkitemID='000001104681'\)/HeaderDetails?\\$expand=ItemDetails,Notes,Attachments,ItemDetails,Limits](http://<host>.<domain>:<port>/sap/opu/odata/sap/GBAPP_POAPPROVAL;mo/WorkflowTaskCollection(SAP__Origin='ONE_800',WorkitemID='000001104681')/HeaderDetails?$expand=ItemDetails,Notes,Attachments,ItemDetails,Limits)

Step 2. Test OData in the Gateway client

2.1 Copy & Paste the OData URI

Transaction: /IWFND/GW_CLIENT. You should logon with the application user same as Fiori app logon.

The screenshot shows the SAP NetWeaver Gateway Client interface. The 'Execute' button is circled in red. The 'Request URI' field contains the copied OData URI: /sap/opu/odata/sap/GBAPP_POAPPROVAL;mo/WorkflowTaskCollection(SAP__Origin='ONE_800',WorkitemID='000001104681')/HeaderDetails?\$expand=ItemDetails,Notes,Attachments,ItemDetails,Limits

2.1 Check the OData response

The screenshot shows the SAP NetWeaver Gateway Client interface with the 'HTTP Response' panel open. The status code is listed as 200. The XML response body is displayed, showing delivery address details: <d:DeliveryAddress m:type="GBAPP_POAPPROVAL.DeliveryAddress"> <d:Plant>1000</d:Plant> <d:PlantName>Werk Hamburg</d:PlantName> <d:AddressString>Alsterdorfer Strasse 13 / D-22299 HAMBURG-ALSTERDO</d:AddressString>

If http response is not 200, you may have some connection setting issue.

If ttp response is 200 but OData service does not return application data,

you should go Step.5 debug section.

Step 3. Check error log

3.1 Transaction /IWFND/ERROR_LOG

The screenshot shows the SAP NetWeaver Gateway: Error Log interface. At the top, there is a toolbar with various icons. Below it is a sub-toolbar with icons for search, print, etc., followed by a red circle around the "Error Context" button. The main area is titled "Overview" and contains a table with several error entries. The table has columns for Line, Entr..., Date, Time, User, T100 Error ID, T10..., Error ..., ICF Node, B, and Error Text. The first entry in the table is highlighted with a yellow background.

Line	Entr...	Date	Time	User	T100 Error ID	T10...	Error ...	ICF Node	B	Error Text
5	1	13.11.2013	10:25:11	BLACKM	/IWFND/CM_MGW033		1	odata		Fehler beim Parsen
4	1		10:18:21	MANAGER	/IWFND/CM_MGW033		2	odata		Fehler beim Parsen
3	1		09:27:14	BLACKM	/IWFND/CM_MGW033		2	odata		Fehler beim Parsen
2	1		07:23:21	BLACKM	/IWFND/CM_MGW033		1	odata		Error while parsing a
1	1		06:27:22	BLACKM	/IWFND/CM_MGW053		1	odata		Missing customizing;

Select the "Error Context"

The screenshot shows the "Error Context" screen. At the top, there is a toolbar with various icons. Below it is a sub-toolbar with icons for XML Format, Call Stack, Application Log, and Request Data, with the "Call Stack" button circled in red. The main area is titled "Error Context" and contains a table with system properties. The table has columns for Ex... and Name, and a Value column. The properties listed include .ERROR_CONTEXT, ..ERROR_INFO, ..SERVICE_INFO, ...NAMESPACE, ...VERSION, ...SERVICE_NAME, ...REQUEST_DIRECTION, and ...AGENT.

Ex...	Name	Value
	.ERROR_CONTEXT	
	..ERROR_INFO	Missing customizing; contact your system administrator
	..SERVICE_INFO	
	...NAMESPACE	/IWFND/
	...VERSION	0001
	...SERVICE_NAME	RMTSAMPLEFLIGHT
	...REQUEST_DIRECTION	Consumer to Gateway
	...AGENT	/IWFND/MGW_RMT

Call Stack helps you to set external break points.

The screenshot shows the "Call Stack" screen. At the top, there is a toolbar with a "Call Stack" icon. The main area is a table with columns for S..., Event Type, Event, Program, and Line... It lists a series of method calls from the application code down to the database layer, including FIND_SVC_DESTINATION_OLD, GET_MODEL, and various provider methods like CL_MED_MDL_PROVIDER.

S...	Event Type	Event	Program	Line...
2...	METHOD	FIND_SVC_DESTINATION_OLD	/IWFND/CL_MGW_DEST_FINDER=====CP	80
1...	METHOD	GET_MODEL	/IWFND/CL_MGW_MED_MDL_LOAD=====CP	70
1...	METHOD	/IWFND/IF_MED_MDL_LOAD~LOAD STRUCTURAL_MODEL	/IWFND/CL_MGW_MED_MDL_LOAD=====CP	16
1...	METHOD	GET_EXTERNAL_MODEL	/IWFND/CL_MED_MDL_DB_ACCESS====CP	16
1...	METHOD	GET_META_DATA_FOR_EXT_MODEL	/IWFND/CL_MED_MDL_DB_ACCESS====CP	13
1...	METHOD	/IWFND/IF_MED_PERSIST RE~GET_META_DATA_MODEL	/IWFND/CL_MED_MDL_DB_ACCESS====CP	46
1...	METHOD	CREATE_CLUSTERED_MODEL	/IWFND/CL_MED_MDL_PROVIDER=====CP	85
1...	METHOD	GET_META_MODELS	/IWFND/CL_MED_MDL_PROVIDER=====CP	80
1...	METHOD	GET_SHARED_META_MODELS	/IWFND/CL_MED_MDL_PROVIDER=====CP	10
1...	METHOD	GET_SERVICE_GROUP	/IWFND/CL_MED_MDL_PROVIDER=====CP	94
1...	METHOD	/IWFND/IF_MED_PROVIDER~GET_SERVICE_GROUP	/IWFND/CL_MED_MDL_PROVIDER=====CP	3
9	METHOD	GET_SERVICE_GROUP_METADATA	/IWFND/CL_TRANSACTION_HANDLER=CP	15

3.2 Transaction /IWFND/APPS_LOG

SAP NetWeaver Gateway Application Logs				
Technical Information				
External ID of the Log	Num...	Altern ID	GSDO Type	Operation
005056A2046C1EE3938488A02795B2F2	8			
Technical message	8			
/IWFND/MGW_RMT	1			
/IWFND/CL_MGW_DEST_FINDER=====CP	1			
FIND_SVC_DESTINATION_OLD	1			
000002	1			
Metadata access	2			
/IWFND/CL_MGW_MED_MDL_LOAD=====CP	2			
GET_MODEL	2			
000003	1			
000004	1			
OData Lib 1.0 Handler	4			
SAP Netweaver Gateway Logger	1			

Buttons:

Message Bar: T... Message Text
2 Missing customizing; contact your system administrator

Reference: [Troubleshooting a SAP Netweaver Gateway Service](#)

Step 4. Find the Runtime Data Provider class

4.1 Transaction /IWFND/MAINT_SERVICE

Activate and Maintain Services				
Service Catalog				
Type	Technical Service Name	Ver...	Service Description	Ext...
	/IW_CNT/EMPLOYEE TIMESHEET	1	Employee Timesheet Recording	EMF...
	/IW_CNT/SG_ERPSALES	1	Service Document for ERP SALES	ERP...
BEP	ZGBAPP_POAPPROVAL	1	SAP Purchase Order Approval (MM-PUR)	GBA...
RFP	ZGBAPP_PRSPPRVWMAI	1	SAP Purchase Requisition Approval (MM-PRV)	GBA...

Buttons:

System Aliases

Display Service

Service Information	
Technical Service Name	GBAPP_POAPPROVAL
Service Version	1
Description	SAP Purchase Order Approval (MM-PUR)
External Service Name	GBAPP_POAPPROVAL
Namespace	
Data Provider Class	CL_GBAPP_APV_PO_RDP

4.2 Transaction SE24

Open the Data Provider Class. Go to Property tab.

Class Builder: Display Class CL_GBAPP_APV_PO_RDP

← → | Local Definitions

Class Interface **CL_GBAPP_APV_PO_RDP** Implemented / Active

Properties Interfaces Friends Attributes Methods Events Types AI

Superclass Undo inheritance Change Inheritance

Superclass **/IWBEP/CL_MGW_PUSH_ABS_DATA** Modeled only

Description Runtime class for PO approval

Inst.Generation Public Final

Forward declarations

Type group/Object type **CL_GBAPP_APV_PO_MDP**

General Data

Fixed point arithmetic Unicode checks active

Shared Memory-Enabled

Message Class

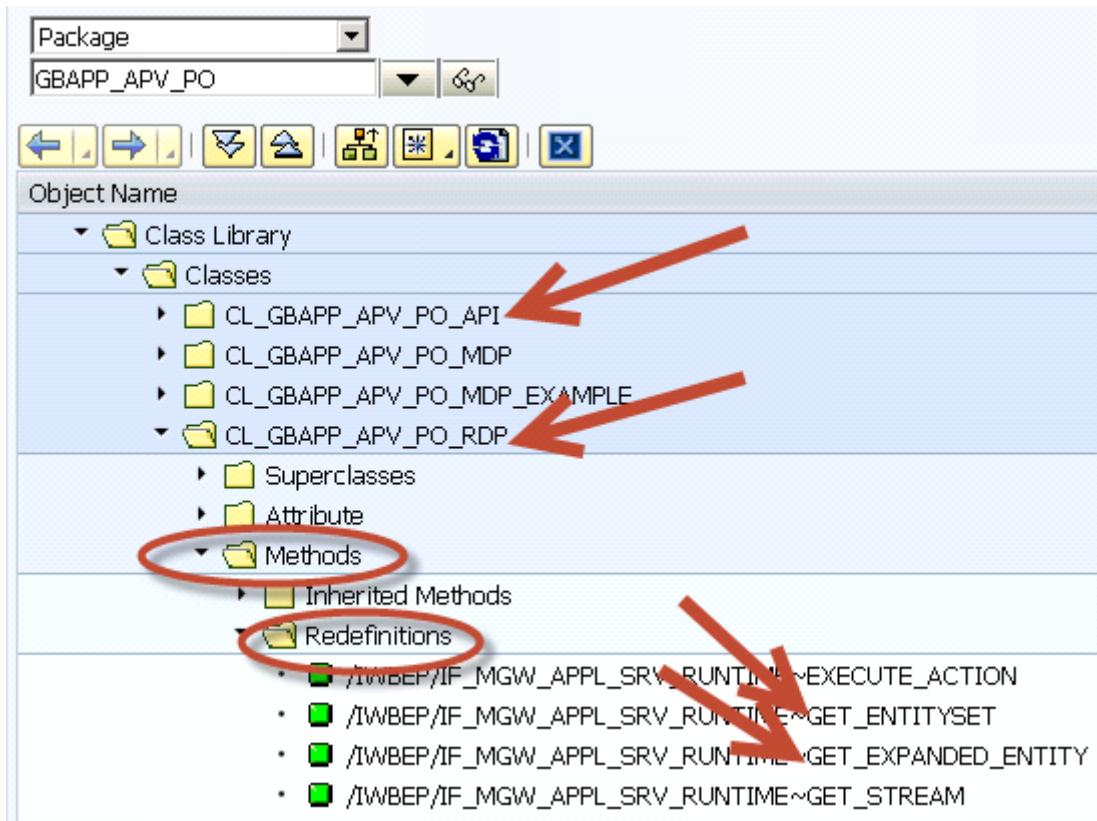
Program status

Category General Object Type

Package **GBAPP_APV_PO**

4.3 Transaction SE80

Open the package.



Class -> Runtime Data Provider Class -> Method -> Redefinitions

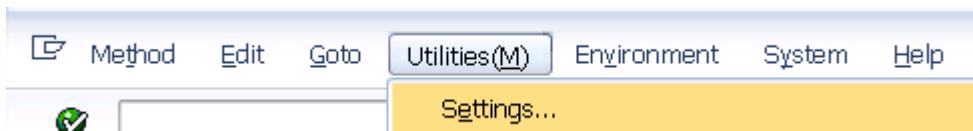
The example OData URI has &expand option, so you should look at the method GET_EXPANDED_ENTITY.

HTTP Method	Base URL plus	Query String	XML Body	OData Operation	ABAP Method
GET	Nothing			Handled by the Gateway Framework	
GET	Entity Set Name	None or \$filter		QUERY	GET_ENTITYSET
GET	Entity Set Name	Key Predicate		READ	GET_ENTITY
PUT	Entity Set Name	Key Predicate	Yes	UPDATE	UPDATE_ENTITY
POST	Entity Set Name		Yes	CREATE	CREATE_ENTITY
DELETE	Entity Set Name	Key Predicate		DELETE	DELETE_ENTITY
GET/POST	Action Name	Variable parameter format		Custom Action	EXECUTE_ACTION

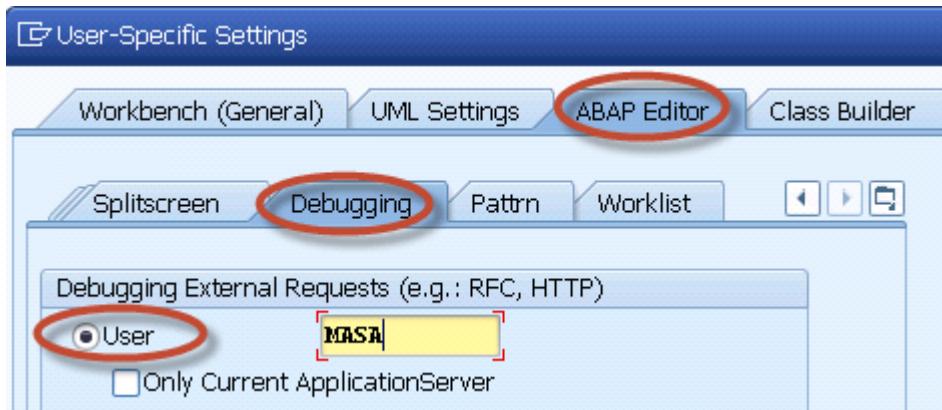
Step 5. Set External break points and debug

5.1 Check the user

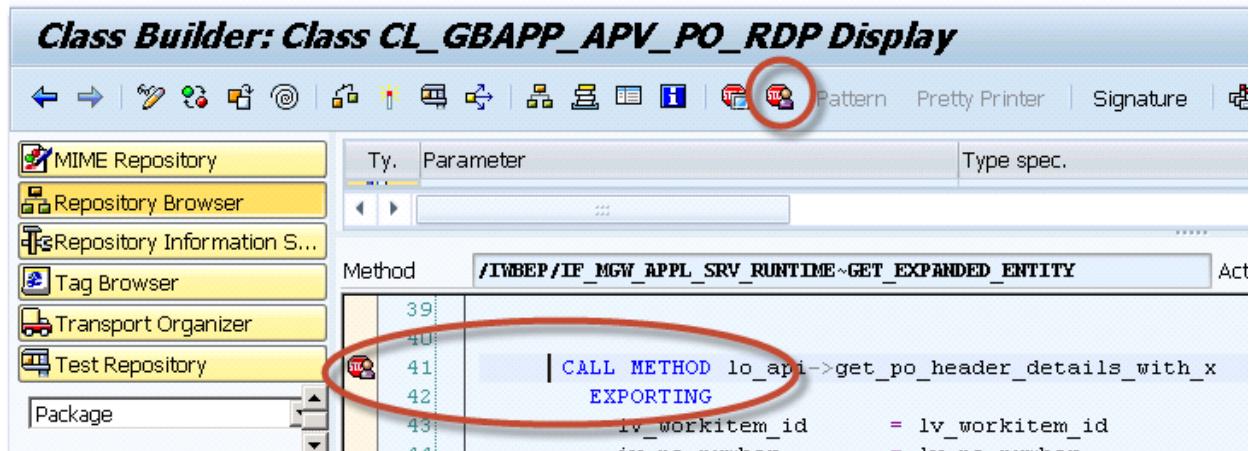
Go to Utilities -> Settings



Make sure the user is application user.



5.2 Set external break points



5.3 Debug

Test the OData URI from Gateway Client. You can now debug the code. Most of case, issue is related with ERP customizing or master data.

Each customer has different customizing setting and master data.

Regards,

Masa

SAP Rapid Innovation Group – RIG

A Step by Step process to create Odata services in SAP / SAP HANA system | SAP Blogs

Thursday, April 13, 2023 4:49 PM

Clipped from: <https://blogs.sap.com/2021/05/06/a-step-by-step-process-to-create-odata-services-in-sap-sap-hana-system/>

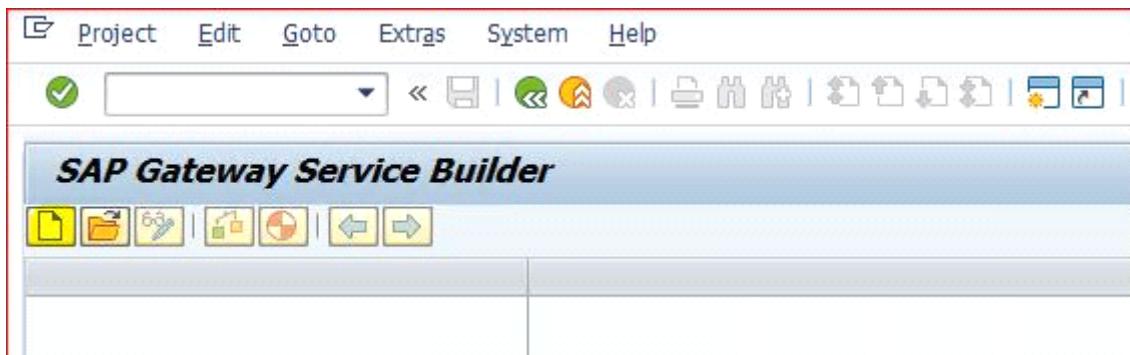
Hello,

While creating Odata services, I had gone through multiple blogs post and found that there are few steps which are not self explanatory and hence at the end of creating an odata service we get bad HTTP response . This is mainly due to missing explanation for how to load meta data and how to test the entity set data .

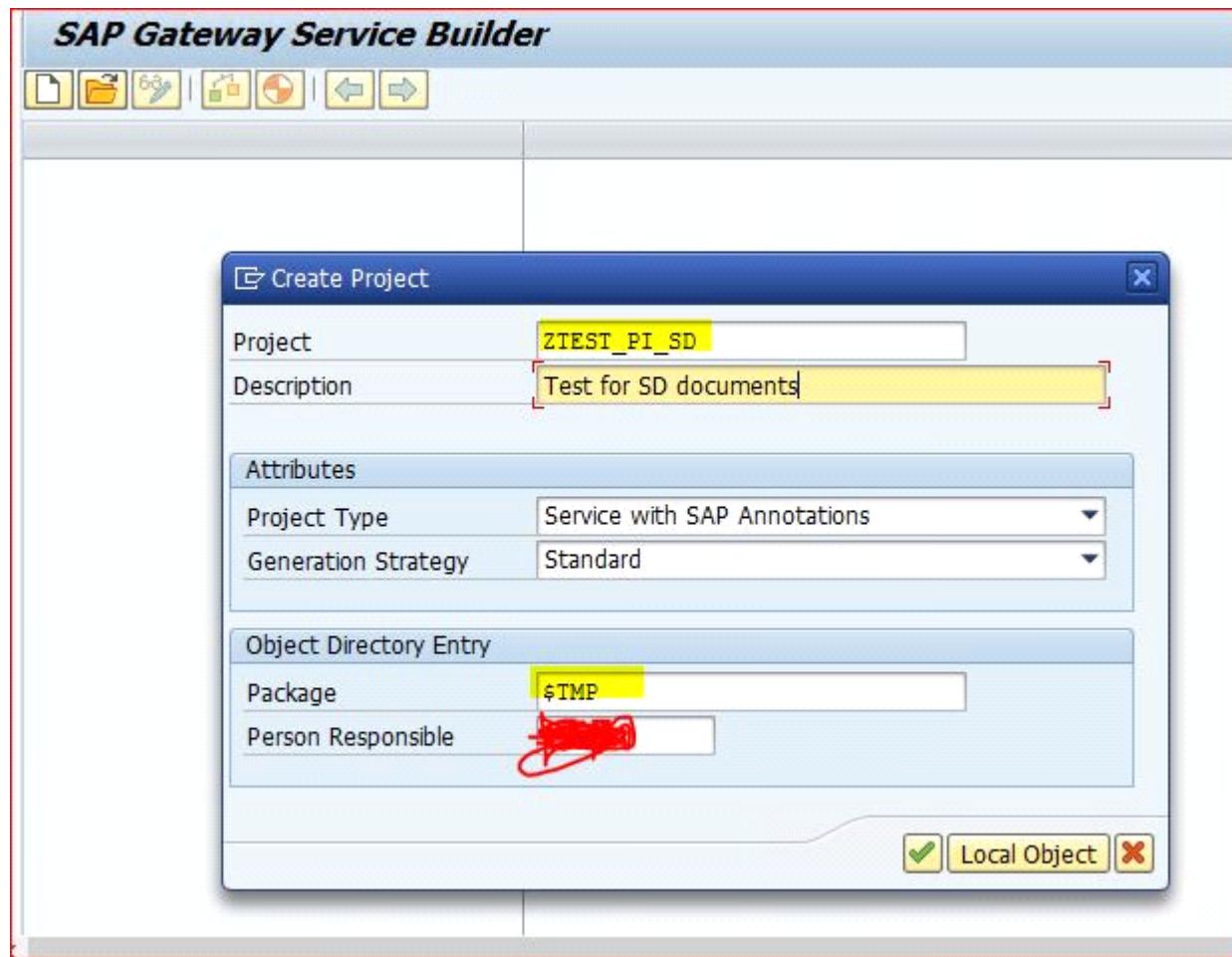
In this blog post, I am going to tell you step by step process to create Odata services for SAP /SAP HANA system and how to test it.

I tried to explain the whole process in 4 steps.

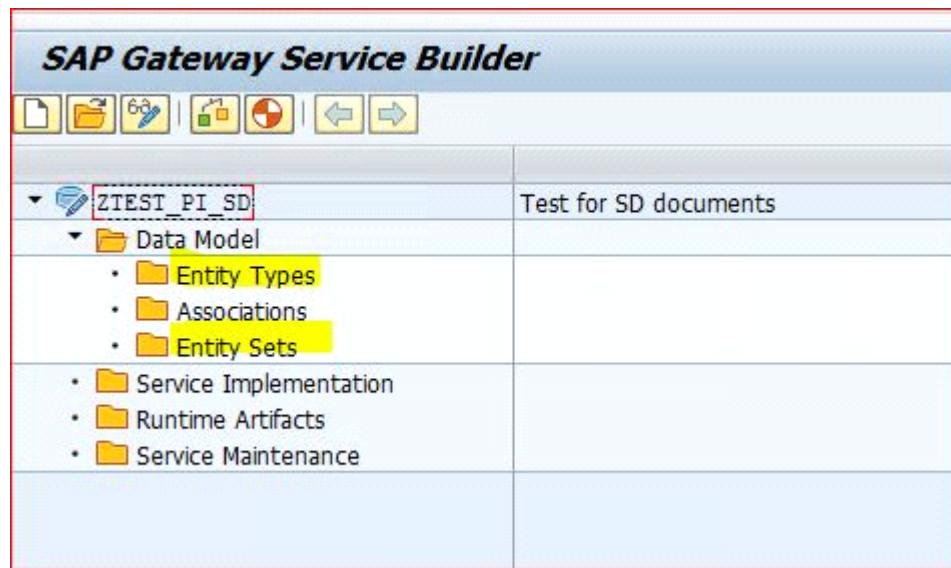
Step 1. Go to transaction code – SEGW .



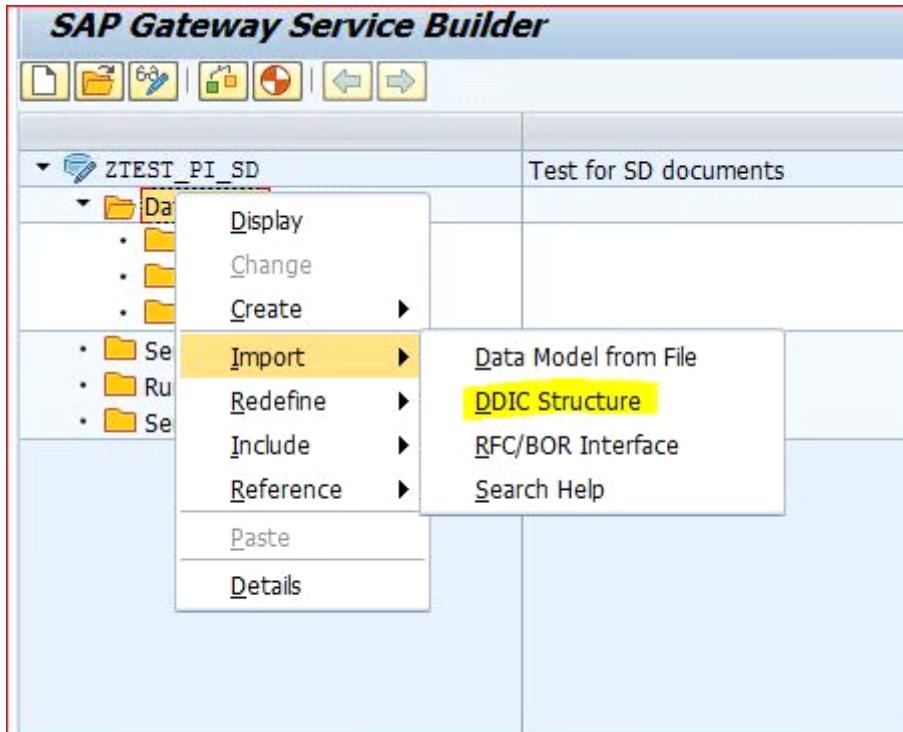
Click on Icon Create. A pop window will appear , Fill the details as per below mention in screen shot and click on check icon or enter.



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.

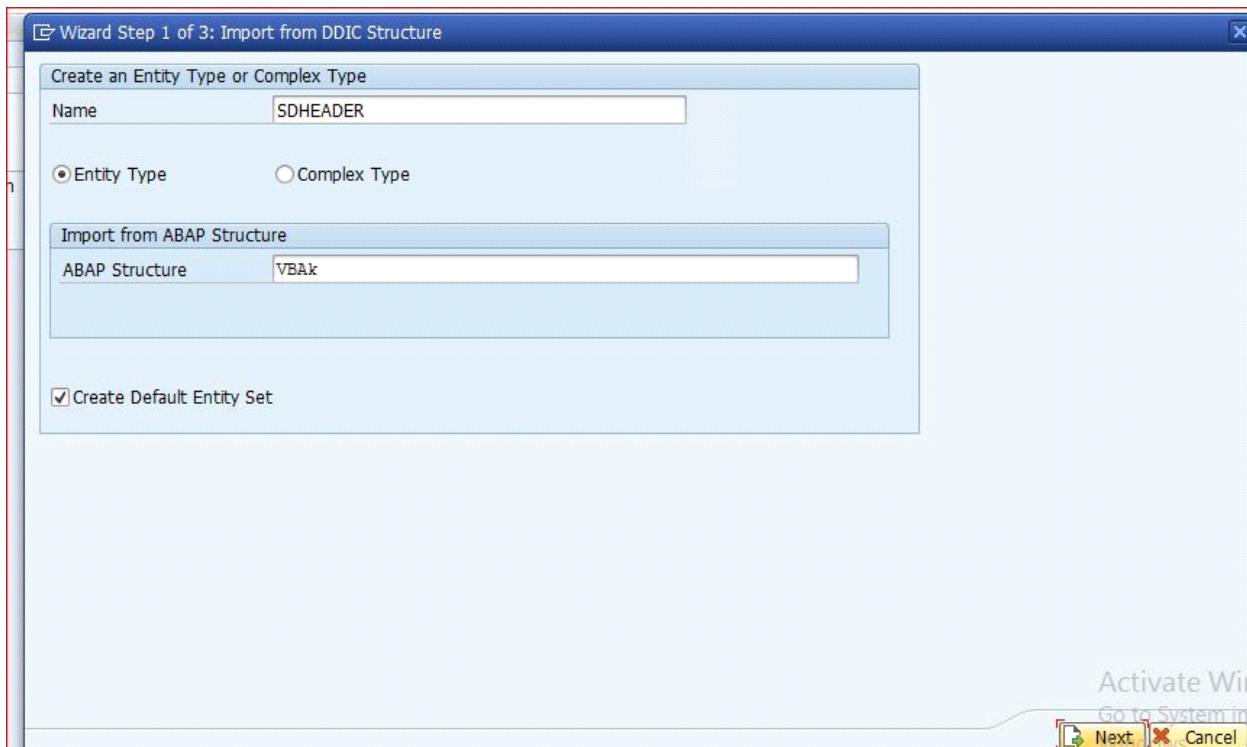


Now we are going to define structure of work area and internal table , Right click on Data model select import and select DDIC structure .



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



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
• <input checked="" type="checkbox"/> VBAK	<input type="checkbox"/>	Client	VBAK	
• <input type="checkbox"/> MANDT	<input type="checkbox"/>	Sales Document	CLNT	3
• <input checked="" type="checkbox"/> VBELN	<input type="checkbox"/>	Created on	CHAR	10
• <input checked="" type="checkbox"/> ERDAT	<input type="checkbox"/>	Time	DATS	8
• <input checked="" type="checkbox"/> ERZET	<input type="checkbox"/>	Created by	TIMS	6
• <input checked="" type="checkbox"/> ERNAM	<input type="checkbox"/>	Valid From	CHAR	12
• <input checked="" type="checkbox"/> ANGDT	<input type="checkbox"/>	Valid To	DATS	8
• <input checked="" type="checkbox"/> BNDDT	<input type="checkbox"/>	Document Date	DATS	8
• <input checked="" type="checkbox"/> AUDAT	<input type="checkbox"/>	Document Cat.	CHAR	4
• <input type="checkbox"/> VBTYP	<input type="checkbox"/>	Transact. Group	CHAR	1
• <input type="checkbox"/> TRVOG	<input type="checkbox"/>	Sales Doc. Type	CHAR	4
• <input type="checkbox"/> AUART	<input type="checkbox"/>	Order Reason	CHAR	3
• <input type="checkbox"/> AUGRU	<input type="checkbox"/>	Begin guarantee	DATS	8
• <input type="checkbox"/> GWLDT	<input type="checkbox"/>	Collective No.	CHAR	10
• <input type="checkbox"/> SUBMI	<input type="checkbox"/>	Delivery Block	CHAR	2
• <input type="checkbox"/> LIFSK	<input type="checkbox"/>	Billing Block	CHAR	2
• <input type="checkbox"/> FAKSK	<input type="checkbox"/>	Net Value	CURR	15
• <input type="checkbox"/> NETWR	<input type="checkbox"/>	Doc. Currency	CURR	15
• <input type="checkbox"/> WAFERK	<input type="checkbox"/>			

Activate Window Go to System Info Back Next Cancel

project5

Other window will appear where we have to 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
<input checked="" type="checkbox"/>	SDHEADER	VBELN	<input checked="" type="checkbox"/>	CHAR	Vbeln	Sales Document
<input checked="" type="checkbox"/>	SDHEADER	ERDAT	<input type="checkbox"/>	DATS	Erdat	Created on
<input checked="" type="checkbox"/>	SDHEADER	ERZET	<input type="checkbox"/>	TIMS	Erzet	Time
<input checked="" type="checkbox"/>	SDHEADER	ERNAM	<input type="checkbox"/>	CHAR	Ernam	Created by
<input checked="" type="checkbox"/>	SDHEADER	ANGDT	<input type="checkbox"/>	DATS	Angdt	Valid From
<input checked="" type="checkbox"/>	SDHEADER	BNDDT	<input type="checkbox"/>	DATS	Bnddt	Valid To
<input checked="" type="checkbox"/>	SDHEADER	AUDAT	<input type="checkbox"/>	DATS	Audat	Document Date

Activate Window Go to System Info Back Finish Cancel

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.

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

Similarly Create Structure for SD Item following same steps

Wizard Step 1 of 3: Import from DDIC Structure

Create an Entity Type or Complex Type

Name: SDITEM

Entity Type: Entity Type Complex Type

Import from 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	<input type="checkbox"/>	VBAP				
MANDT	<input type="checkbox"/>	Client	CLNT	3		
VBELN	<input checked="" type="checkbox"/>	Sales Document	CHAR	10		
POSNR	<input checked="" type="checkbox"/>	Item	NUMC	6		
MATNR	<input checked="" type="checkbox"/>	Material	CHAR	40		
MATWA	<input checked="" type="checkbox"/>	MaterialEntered	CHAR	40		
PMATN	<input checked="" type="checkbox"/>	Pr. Ref. Matl	CHAR	40		
CHARG	<input checked="" type="checkbox"/>	Batch	CHAR	10		
MATKL	<input checked="" type="checkbox"/>	Material Group	CHAR	9		
ARKTX	<input checked="" type="checkbox"/>	Item Descr.	CHAR	40		
PSTYV	<input type="checkbox"/>	Item category	CHAR	4		
POSAR	<input type="checkbox"/>	Item Type	CHAR	1		
LFREL	<input type="checkbox"/>	Item Delf.Div.	CHAR	1		
FKREL	<input type="checkbox"/>	Relev.for Bill.	CHAR	1		
UEPOS	<input type="checkbox"/>	Higher-Lev.Item	NUMC	6		
GRPOS	<input type="checkbox"/>	Altern. to Item	NUMC	6		
ABGRU	<input type="checkbox"/>	RejectionReason	CHAR	2		
PRODH	<input type="checkbox"/>	Prod.hierarchy	CHAR	18		
PRODUNIT	<input type="checkbox"/>	Node ID	CHAR	40		

Wizard Step 3 of 3: Import from DDIC Structure

Modify Entity Type

IEn...	Complex/Entity Type	ABAP Name	Is K...	Type	Name	Label
<input checked="" type="checkbox"/>	SDITEM	VBELN	<input checked="" type="checkbox"/>	CHAR	Vbeln	Sales Document
<input checked="" type="checkbox"/>	SDITEM	POSNR	<input checked="" type="checkbox"/>	NUMC	Posnr	Item
<input checked="" type="checkbox"/>	SDITEM	MATNR	<input type="checkbox"/>	CHAR	Matnr	Material
<input checked="" type="checkbox"/>	SDITEM	MATWA	<input type="checkbox"/>	CHAR	Matwa	MaterialEntered
<input checked="" type="checkbox"/>	SDITEM	PMATN	<input type="checkbox"/>	CHAR	Pmatn	Pr. Ref. Matl
<input checked="" type="checkbox"/>	SDITEM	CHARG	<input type="checkbox"/>	CHAR	Charg	Batch
<input checked="" type="checkbox"/>	SDITEM	MATKL	<input type="checkbox"/>	CHAR	Matkl	Material Group
<input checked="" type="checkbox"/>	SDITEM	ARKTX	<input type="checkbox"/>	CHAR	Arktx	Item Descr.

Activate Window

Go to System Information

Back Finish Cancel

Properties

Name	Key	Edm Type	Prec.	Scale	Max.	Unit Prop.	Cre...	Upd...	Sort...	Null...	Filt...	Label	L...	Comp. Type	ABAP Field	A...	Semantics
Vbeln	<input checked="" type="checkbox"/>	Edm.String	0	0	10		<input checked="" type="checkbox"/>	Sales Docu...	T	VBELN	<input checked="" type="checkbox"/>						
Posnr	<input checked="" type="checkbox"/>	Edm.String	0	0	6		<input checked="" type="checkbox"/>	Item	T	POSNR	<input checked="" type="checkbox"/>						
Matnr	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	Material	T	MATNR	<input checked="" type="checkbox"/>						
Matwa	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	MaterialEnt...	T	MATWA	<input checked="" type="checkbox"/>						
Pmatn	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	Pr. Ref. Matl	T	PMATN	<input checked="" type="checkbox"/>						
Charg	<input type="checkbox"/>	Edm.String	0	0	10		<input checked="" type="checkbox"/>	Batch	T	CHARG	<input checked="" type="checkbox"/>						
Matkl	<input type="checkbox"/>	Edm.String	0	0	9		<input checked="" type="checkbox"/>	Material Gro...	T	MATKL	<input checked="" type="checkbox"/>						
Arktx	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	Item Descr.	T	ARKTX	<input checked="" type="checkbox"/>						

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 details and proceed further . These are also knowns as runtime artifacts.

Model and Service Definition

Model Provider Class	
Class Name	ZCL_ZTEST_PI_SD_MPC_EXT
Base Class Name	ZCL_ZTEST_PI_SD_MPC
Data Provider Class	
<input checked="" type="checkbox"/> 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

Step 2. Go to transaction code /IWFND/MAINT_SERVICE .

Click on push button Add services .

Activate and Maintain Services

Service Catalog								
Type	Technical Service Name	V...	Service Description	External Service Name	Nsp.	OAuth	Soft State	Processing Mode
BEP	ZABAP REPOSITORY_SRV	1	SAPUI5 ABAP Repository services via oData	ABAP REPOSITORY_SRV	/UI5/	<input type="checkbox"/>	Not Supported	Routing-based
BEP	ADT	1	Gateway service for ADT	ADT_SRV		<input checked="" type="checkbox"/>	Not Supported	Routing-based
BEP	ZAPJ JOB MANAGEMENT_SRV	1	Application job management	APJ JOB MANAGEMENT_SRV		<input checked="" type="checkbox"/>		Co-deployed only
BEP	ZAPL LOG MANAGEMENT_SRV	1	Application Log management	APL LOG MANAGEMENT_SRV		<input checked="" type="checkbox"/>		Co-deployed only
BEP	ZAPS CDS GKE_SRV	1	Generic CDS key user access	APS CDS GKE_SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZAPS CHANGE DOCUMENTS_SRV	1	OData Service for Change Documents	APS CHANGE DOCUMENTS_SRV		<input checked="" type="checkbox"/>		Co-deployed only
BEP	ZAPS EXT CBO D_SRV	1	Custom Business Objects: Draft	APS EXT CBO D_SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZAPS EXT CCL_SRV	1	Custom Code List	APS EXT CCL_SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZAPS EXT TRC_SRV	1	OData Services for Tracing	APS EXT TRC_SRV		<input type="checkbox"/>	Not Supported	Routing-based

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

Add Selected Services

Get Services

Filter

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

Add Selected Services

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	

Activate Windows
Go to System in Control
Windows

Now double click on your project name . 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	\$IMP
	Local Object
ICF Node	
<input checked="" type="radio"/> Standard Mode	<input type="radio"/> None
<input checked="" type="checkbox"/> Set Current Client as Default Client in ICF Node	
OAuth enablement	
<input type="checkbox"/> 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

The screenshot shows the SAP Gateway Client interface. At the top, there is a table listing various technical services with their descriptions, external service names, namespaces, OAuth status, soft state, and processing mode. Some rows are highlighted in yellow. Below this is a toolbar with icons for ICF Node, Call Browser, and SAP Gateway Client. Underneath the toolbar, there are two tabs: 'ICF Nodes' and 'System Aliases'. The 'ICF Nodes' tab shows a single entry for 'ODATA' with a status of 'Standard Mode'. The 'System Aliases' tab shows a single entry for 'LOCAL' with a description of 'Local System Alias'.

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

The screenshot shows the SAP Gateway Client interface after executing a transaction. The 'Request URI' field contains '/sap/opu/odata/sap/ZTEST_PT_SD_SRV/?\$format=xml'. The 'HTTP Response' section shows a processing time of 138 ms. The response header table includes rows for 'status_code' (200) and 'status_reason' (OK). The XML response body is displayed below, showing the structure of the OData service endpoint.

```

<?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_PT_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:createable="false">
          <atom:title type="text">SDHEADERSet</atom:title>
        
```

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.

The screenshot shows the SAP ABAP Workbench interface. In the top bar, 'Class/Interface' is set to 'ZCL_ZTEST_PI_SD_DPC_EXT'. Below the toolbar, the object name is 'Object Name' and the description is 'Description'. The 'Methods' section is expanded, showing 'Inherited Methods' which include several methods from the /IWBEPE/IF_SB_DPC_COMM_SE interface. One method, 'SDHEADERSET_GET_ENTITYYS', is highlighted with a yellow box. A context menu is open over this method, with the 'Redefine' option highlighted in yellow. Other options in the menu include 'Change', 'Display', 'Consistency Check', 'Activate', 'Copy...', 'Where-Used List', 'Version Management', and 'Additional Functions'. To the right of the menu, there are two columns of related objects: 'Related EntitySet Name: SDHEA' and 'Related EntitySet Name: SDITEM'.

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

Builder Class ZCL_ZTEST_PI_SD_DPC_EXT Change

```

Type: Parameter          Typing
IT_ORDER                 TYPE /IWBEPI/T_MGW_SORTING_ORDER
IV_FILTER_STRING          TYPE STRING
IV_SEARCH_STRING          TYPE STRING
IO_TECH_REQUEST_CONTEXT   TYPE REF TO /IWBEPI/IF_MGW_REQ_ENTITYSET OPTIONAL
ET_ENTITYSET               TYPE ZCL_ZTEST_PI_SD_MPC->TT_SDHEADER
ES_RESPONSE_CONTEXT        TYPE /IWBEPI/IF_MGW_APPL_SRV_RUNTIME->TY_S_MGW_RESPONSE_CONTEXT
/IWBEPI/CX_MGW_BUSI_EXCEPTION
/IWBEPI/CX_MGW_TECH_EXCEPTION

Method: SDHEADERSET_GET_ENTITYSET
active
1   method SDHEADERSET_GET_ENTITYSET.
2
3     Select * from vbak into CORRESPONDING FIELDS OF table ET_ENTITYSET
4       UP TO 5 rows.
5
6   **TRY.

```

Similarly , Write code for GetEntitySet for Item and activate .

```

Type: Parameter          Typing
IT_ORDER                 TYPE /IWBEPI/T_MGW_SORTING_ORDER
IV_FILTER_STRING          TYPE STRING
IV_SEARCH_STRING          TYPE STRING
IO_TECH_REQUEST_CONTEXT   TYPE REF TO /IWBEPI/IF_MGW_REQ_ENTITYSET OPTIONAL
ET_ENTITYSET               TYPE ZCL_ZTEST_PI_SD_MPC->TT_SDITEM
ES_RESPONSE_CONTEXT        TYPE /IWBEPI/IF_MGW_APPL_SRV_RUNTIME->TY_S_MGW_RESPONSE_CONTEXT
/IWBEPI/CX_MGW_BUSI_EXCEPTION
/IWBEPI/CX_MGW_TECH_EXCEPTION

Method: SDITEMSET_GET_ENTITYSET
active
1   method SDITEMSET_GET_ENTITYSET.
2
3     Select * from VBAP into CORRESPONDING FIELDS OF TABLE ET_ENTITYSET
4       UP TO 10 rows.
5
6   **TRY.
7     *CALL METHOD SUPER->SDITEMSET_GET_ENTITYSET
8     + EXPORTING
9       IV_ENTITY_NAME      =
10      IV_ENTITY_SET_NAME  =
11      IV_SOURCE_NAME      =
12      IT_FILTER_SELECT_OPTIONS =
13      IS_PAGING            =

```

Step -4

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

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 .

The screenshot shows the SAP Gateway Client interface. At the top, there's a table titled 'EntitySets' with several entries. Below it are three tabs: 'ICF Nodes', 'System Aliases', and others. The 'ICF Nodes' tab shows a single entry for 'ODATA'. The 'System Aliases' tab shows a single entry for 'LOCAL'.

BEP	ZTEST ODATA_SRV	1 My first Odata project	ZTEST PI_SD_SRV	Not Supported	Routing
BEP	ZTEST PIN PO_SRV	1 test PO Odata	ZTEST PIN PO_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTEST PIN SD_SRV	1 Test for SD documents	ZTEST PIN SD_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTEST SERVICE GENERATION_SRV	1 Odata for Service Generatio technique	ZTEST SERVICE GENERATION_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTEST_SRV	1 test	ZTEST_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTEST TABLE1_SRV	1 test	ZTEST TABLE1_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTRANSLATIONS_SRV	1 cds view translations	ZTRANSLATIONS_SRV	<input type="checkbox"/>	Not Supported Co-dep
BEP	ZX SO CDS	1 Sales Order	ZX SO CDS	<input type="checkbox"/>	Not Supported Routing

Now Click on Entity set.

The screenshot shows the SAP Gateway Client interface. The 'EntitySets' tab is selected, displaying a table with two entries: 'SDHEADERSet' and 'SDITEMSet'. A modal dialog box titled 'EntitySets (1) 2 Entries found' is open, showing the same two entries. The 'HTTP Request' and 'HTTP Response' panes are visible at the bottom.

Select entity for header and enter

The screenshot shows the SAP Gateway Client interface. The 'EntitySets' tab is selected, displaying a table with two entries: 'SDHEADERSet' and 'SDITEMSet'. A modal dialog box titled 'EntitySets (1) 2 Entries found' is open, showing the same two entries. The 'SDHEADERSet' entry is highlighted. The 'HTTP Request' and 'HTTP Response' panes are visible at the bottom.

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

HTTP Method: GET

Request URI: /sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDHEADERSet

Protocol: HTTP

Test Group: Test Case

HTTP Response - Processing Time = 876 ms

Header Name Value

~status_code	200
~status_reason	OK

```

<category schema="http://schemas.microsoft.com/ado/2007/08/dataservices" type="Collection(SDHEADERSet)" title="SDHEADERSet">
  <link href="SDHEADERSet('1')/SDHEADERSet('1')>
  <content type="application/xml">
    <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
      <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:Bndt m:null="true"/>
      <d:Audat>2020-01-14T00:00:00</d:Audat>
    </m:properties>
  </content>
</category>

```

Activate Windows
Go to System in Control Panel to activate

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

HTTP Method: GET

Request URI: /sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDITEMSet

Protocol: HTTP

Test Group: Test Case

HTTP Response - Processing Time = 1167 ms

Header Name Value

~status_code	200
~status_reason	OK

```

<category schema="http://schemas.microsoft.com/ado/2007/08/dataservices" type="Collection(ITEMSet)" title="ITEMSet">
  <link href="ITEMSet('1')/ITEMSet('1')>
  <content type="application/xml">
    <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
      <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>
</category>

```

Here , We had completed our Odata service creation with a Service 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.

Thanks

Test working service in ECC

Thursday, April 13, 2023 4:49 PM

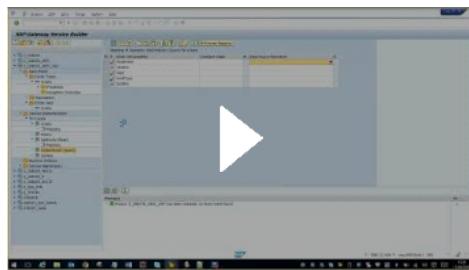
http://sapuxr3d01.ci.root:8001/sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDHEADERSet

http://sapuxr3d01.ci.root:8001/sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDHEADERSet

http://sapuxr3d01.ci.root:8001/sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDITEMSet

BEP	ZSLD_MU_APPROVAL_SRV	1 MU approval test	ZSLD_MU_APPROVAL_SRV	not supported
BEP	ZSLD_PO_APPROVAL_SRV_01	1 PO APPROVL	ZSLD_PO_APPROVAL_SRV_01	Not Supported
BEP	ZSLD_PROJ_SEV_SRV	1 test Project	ZSLD_PROJ_SEV_SRV	Not Supported
BEP	ZSLD_TEST_SRV	1 test	ZSLD_TEST_SRV	Not Supported
BEP	ZTEST_PI_SD_SRV	1 test for sd documents	ZTEST_PI_SD_SRV	Not Supported
BEP	ZTRAINING_READDATA_SRV	1 Read ODATA service for training	ZTRAINING_READDATA_SRV	Not Supported

[Step by Step SAP Gateway exposing a BAPI program via Web Services](#)



ZBW_GET_CDPOS_SIMPLE FM in ECC

Steps to build an RFC based OData Service with multiple selection values | SAP Blogs

Thursday, August 24, 2023 12:28 PM

Clipped from: <https://blogs.sap.com/2014/12/03/steps-to-build-an-rfc-based-odata-service-with-multiple-selection-values/>

Here is a step by step guide on how to build an RFC based OData service with multiple selection (input) values. This document would help developers who are new to the Odata services. You can also find few sample queries at the end of the document.

The business requirement given here is to build an OData service which can retrieve shopping cart information from the system based on a combination of input fields (single and multiple values).

Below is the list of input fields:

Selection fields	Multiple values?
-------------------------	-------------------------

User ID (Requestor)	
------------------------	--

User ID (Creator)	
-------------------	--

Creation Date (From)	
----------------------	--

Creation Date (To)	
--------------------	--

Product Category	Yes
Code	

Vendor Number	Yes
---------------	-----

Shopping Cart Number	
----------------------	--

External Req Number	
---------------------	--

Shopping Cart Status	
----------------------	--

Supplier Part Number	
----------------------	--

Cost Center	
-------------	--

Line Item Description	
-----------------------	--

Company Code	
--------------	--

Purchasing Group	Yes
------------------	-----

Plant

Below is the expected output:

Display fields

Shopping Cart Number

Creation Date

Shopping Cart Status

Shopping Cart

Requestor

Shopping Cart Creator

Shopping Cart

Description

Total Price of
Shopping Cart

Process Level

Current Work Agent

Step1.

Create a RFC function module in the backend system with below parameters as input

Function Builder: Display ZGSTS_ITM_GET_SC_HEADER								
Function module ZGSTS_ITM_GET_SC_HEADER Active								
Attributes Import Export Changing Tables Exceptions Source code								
Parameter Name	Type...	Associated Type		Default value	Op...	Pa...	Short text	Lo...
IM_ISID_REQUESTOR	TYPE	CRMT_CREATED_BY			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	User that Requested the Transaction	
IM_ISID_CREATOR	TYPE	CRMT_CREATED_BY			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	User that Created the Transaction	
IM_CREATE_DATE	TYPE	BBP_CREATE_DATE			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Created From	
IM_CREATE_DATE_TO	TYPE	BBP_CREATE_DATE			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Created To	
IM_SHOP_CART_ID	TYPE	CRMT_OBJECT_ID_DB			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Shopping Cart Number	
IM_CATEGORY_ID	TYPE	ZGSTS_RANGE_CATEGORY_T			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Product Category ID	
IM_VENDOR_ID	TYPE	ZGSTS_RANGE_PARTNER_T			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vendor Number	
IM_EXT_REQ_NO	TYPE	BBP_EXT_DEMID			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	External Requirement Number	
IM_SC_STATUS	TYPE	CRM_J_STATUS			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Shopping Cart Status	
IM_MAN_PART_NO	TYPE	CRMT_ITEM_DESCR_PARTNER			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Supplier Part Number	
IM_COST_CENTER	TYPE	KOSTL			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cost Center	
IM_ITEM_DESC	TYPE	CRMT_PRSHTEXTG_DB			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Item Description	
IM_PUR_GROUP	TYPE	ZGSTS_RANGE_PUR_GROUP_T			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Purchasing group	
IM_COMP_CODE	TYPE	BUKRS			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Company Code	
IM_PLANT	TYPE	BE_EWERK			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Plant	
IM_LANGU	TYPE	LANGU			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Language Key	
					<input type="checkbox"/>	<input type="checkbox"/>		
					<input type="checkbox"/>	<input type="checkbox"/>		
					<input type="checkbox"/>	<input type="checkbox"/>		
					<input type="checkbox"/>	<input type="checkbox"/>		
					<input type="checkbox"/>	<input type="checkbox"/>		
					<input type="checkbox"/>	<input type="checkbox"/>		

For the input parameters that need multiple values to be supplied, create

them as range table types.

The screenshot shows two related configuration screens for a range table type.

Range Table Type Configuration:

- Table Type:** ZGSTS_RANGE_CATEGORY_T (Active)
- Short text:** Table type for Category ID Ranges
- Line Type:** ZGSTS RANGE CATEGORY (selected)
- Data Type:** (empty field)
- No. of Characters:** 0
- Decimal Places:** 0

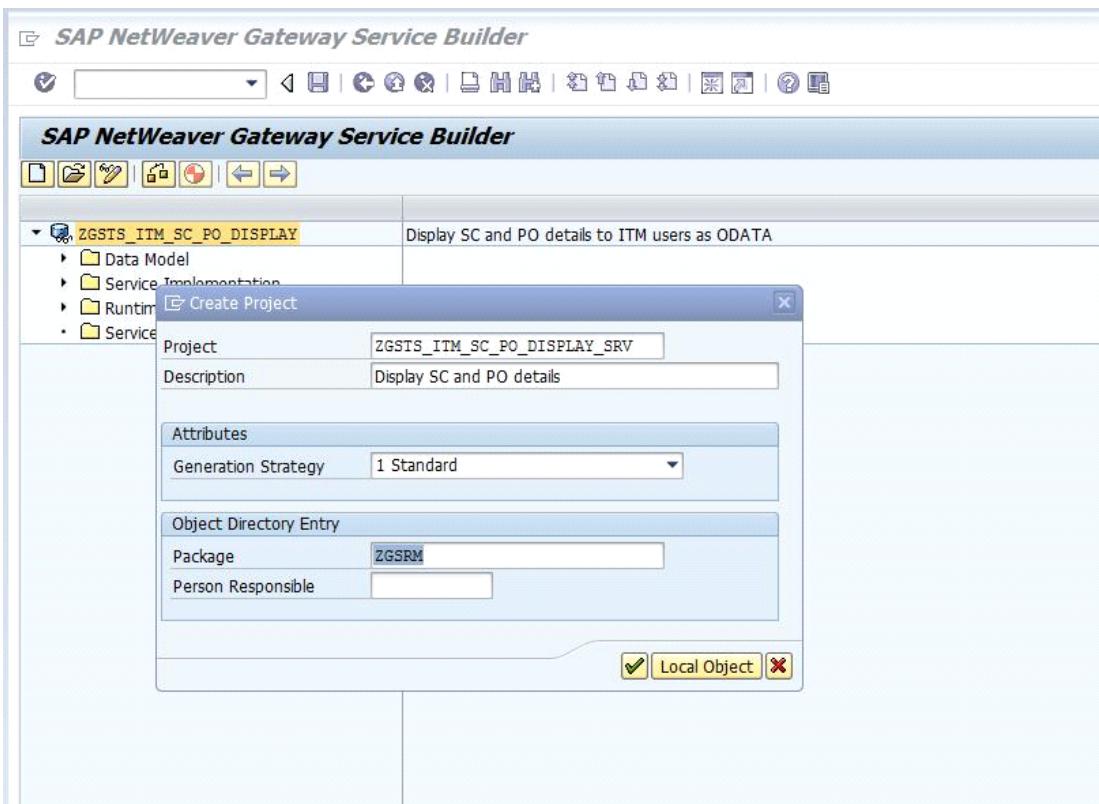
Structure Definition:

- Structure:** ZGSTS_RANGE_CATEGORY (Active)
- Short Description:** Structure for Category ID Ranges
- Components:** Predefined Type
- Table Data:**

Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description
SIGN	1 Types	BAPISIGN	CHAR	1	0	Inclusion/exclusion criterion SIGN for range tables
OPTION	1 Types	BAPIOPTION	CHAR	2	0	Selection operator OPTION for range tables
LOW	1 Types	BBP_CATEGORY_ID	CHAR	20	0	Product Category ID
HIGH	1 Types	BBP_CATEGORY_ID	CHAR	20	0	Product Category ID

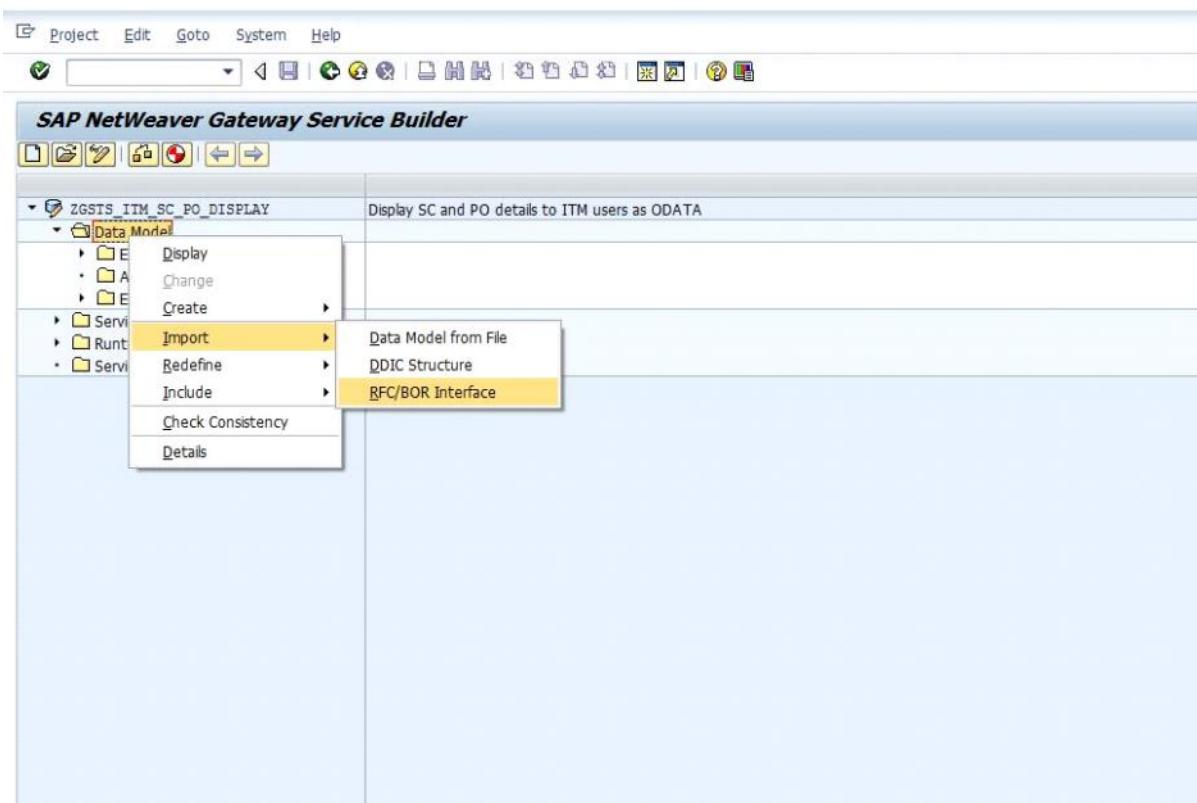
Step2.

Go to SAP Netweaver Gateway Service builder (SEGW) and create a project.



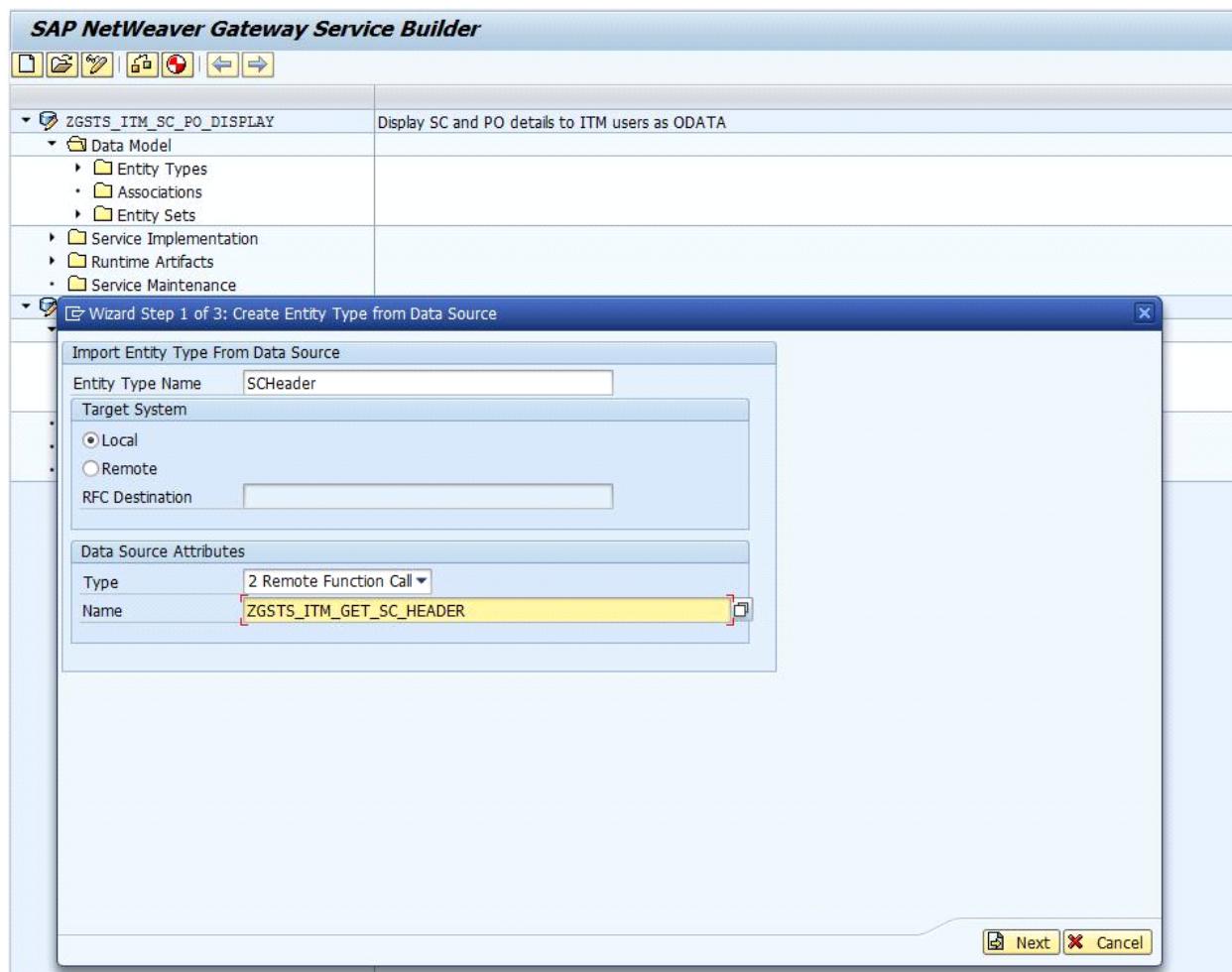
Step3.

Click on Data Model -> Import -> RFC/BOR Interface



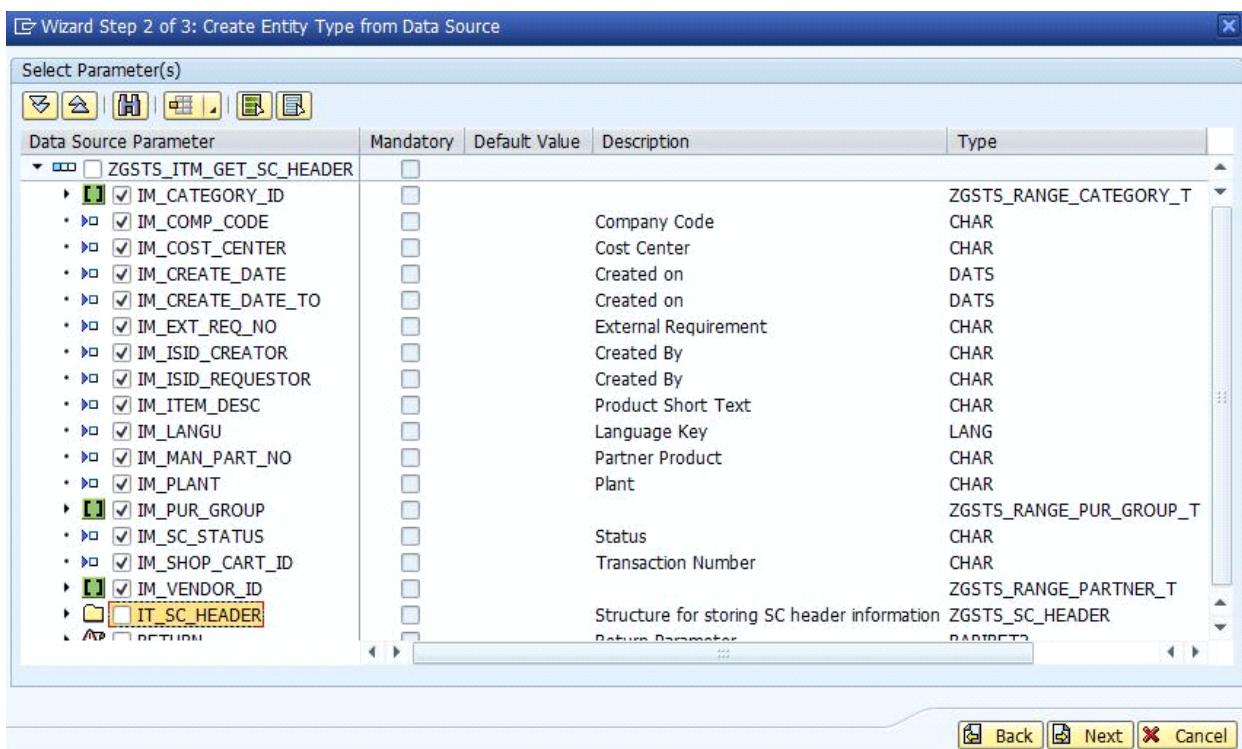
Step4.

Enter the Entity Type name and the RFC details.



Step5.

Select all the fields to be used in the entity type



Step6.

Make the Shopping Cart ID as the key field.

Is E...	Complex/Entity Type Name	ABAP Name	Is K...	Type	Name
<input checked="" type="checkbox"/>	SCHeader	IM_ITEM_DESC	<input type="checkbox"/>	CHAR	ImItemDesc
<input checked="" type="checkbox"/>	SCHeader	IM_LANGU	<input type="checkbox"/>	LANG	ImLangu
<input checked="" type="checkbox"/>	SCHeader	IM_MAN_PART_NO	<input type="checkbox"/>	CHAR	ImManPartNo
<input checked="" type="checkbox"/>	SCHeader	IM_PLANT	<input type="checkbox"/>	CHAR	ImPlant
<input checked="" type="checkbox"/>	SCHeader	IM_SC_STATUS	<input type="checkbox"/>	CHAR	ImScStatus
<input checked="" type="checkbox"/>	SCHeader	IM_SHOP_CART_ID	<input type="checkbox"/>	CHAR	ImShopCartId
<input checked="" type="checkbox"/>	SCHeader	SHOP_CART_ID	<input checked="" type="checkbox"/>	CHAR	ShopCartId
<input checked="" type="checkbox"/>	SCHeader	SC_CREATE_DATE	<input type="checkbox"/>	DATS	ScCreateDate
<input checked="" type="checkbox"/>	SCHeader	SC_HEADER_STAT	<input type="checkbox"/>	CHAR	ScHeaderStat
<input checked="" type="checkbox"/>	SCHeader	SC_REQUESTER	<input type="checkbox"/>	CHAR	ScRequester
<input checked="" type="checkbox"/>	SCHeader	SC_CREATOR	<input type="checkbox"/>	CHAR	ScCreator
<input checked="" type="checkbox"/>	SCHeader	SC_DESCRIPTION	<input type="checkbox"/>	CHAR	ScDescription
<input checked="" type="checkbox"/>	SCHeader	SC_TOTAL_PRICE	<input type="checkbox"/>	CURR	ScTotalPrice
<input checked="" type="checkbox"/>	SCHeader	PROC_LEVEL	<input type="checkbox"/>	CHAR	ProcLevel
<input checked="" type="checkbox"/>	SCHeader	CURR_AGENT	<input type="checkbox"/>	CHAR	CurrAgent
<input checked="" type="checkbox"/>	SCHeader	DUMMY_FIELD	<input type="checkbox"/>	CHAR	DummyField
<input checked="" type="checkbox"/>	SCHeader	ERROR_MSSGS	<input type="checkbox"/>	CHAR	ErrorMssgs

Step7.

Change the field names and set rest of the attributes as required.

Name	K.	Edm Core Type	Pre.	Sc.	Ma.	Unit Prop.	C.	U	D	S.	N	F..	Label	L..	Comp. Type	ABAP Field Name	A	Semantics
IN_CategoryId	<input type="checkbox"/>	Edm.String	0	0	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_CATEGORY_ID	<input type="checkbox"/>		
IN_CompCode	<input type="checkbox"/>	Edm.String	0	0	4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_COMP_CODE	<input type="checkbox"/>		
IN_CostCenter	<input type="checkbox"/>	Edm.String	0	0	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_COST_CENTER	<input type="checkbox"/>		
IN_CreateDate	<input type="checkbox"/>	Edm.DateTime	0	0	0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_CREATE_DATE	<input type="checkbox"/>		
IN_CreateDateTo	<input type="checkbox"/>	Edm.DateTime	0	0	0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_CREATE_DATE_TO	<input type="checkbox"/>		
IN_ExReqNo	<input type="checkbox"/>	Edm.String	0	0	32		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_EX_REQ_NO	<input type="checkbox"/>		
IN_IsdCreator	<input type="checkbox"/>	Edm.String	0	0	12		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_ISD_CREATOR	<input type="checkbox"/>		
IN_IsdRequestor	<input type="checkbox"/>	Edm.String	0	0	12		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_ISD_REQUESTOR	<input type="checkbox"/>		
IN_ItemDesc	<input type="checkbox"/>	Edm.String	0	0	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_ITEM_DESC	<input type="checkbox"/>		
IN_Langu	<input type="checkbox"/>	Edm.String	0	0	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_LANGU	<input type="checkbox"/>		
IN_ManPartNo	<input type="checkbox"/>	Edm.String	0	0	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_MAN_PART_NO	<input type="checkbox"/>		
IN_Plant	<input type="checkbox"/>	Edm.String	0	0	4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_PLANT	<input type="checkbox"/>		
IN_PurGroup	<input type="checkbox"/>	Edm.String	0	0	3		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_PUR_GROUP	<input type="checkbox"/>		
IN_ScStatus	<input type="checkbox"/>	Edm.String	0	0	5		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_SC_STATUS	<input type="checkbox"/>		
IN_ShopCartId	<input type="checkbox"/>	Edm.String	0	0	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_SHOP_CART_ID	<input type="checkbox"/>		
IN_VendorId	<input type="checkbox"/>	Edm.String	0	0	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		IM_VENDOR_ID	<input type="checkbox"/>		
OP_CreateDate	<input checked="" type="checkbox"/>	Edm.String	0	0	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		SHOP_CART_ID	<input type="checkbox"/>		
OP_CreateDate	<input type="checkbox"/>	Edm.DateTime	0	0	0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		SC_CREATE_DATE	<input type="checkbox"/>		
OP_HeaderStat	<input type="checkbox"/>	Edm.String	0	0	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		SC_HEADER_STAT	<input type="checkbox"/>		
OP_Requester	<input type="checkbox"/>	Edm.String	0	0	80		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		SC_REQUESTER	<input type="checkbox"/>		
OP_Creator	<input type="checkbox"/>	Edm.String	0	0	80		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		SC_CREATOR	<input type="checkbox"/>		
OP_Description	<input type="checkbox"/>	Edm.String	0	0	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		SC_DESCRIPTION	<input type="checkbox"/>		
OP_TotalPrice	<input type="checkbox"/>	Edm.Decimal	15	2	0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		SC_TOTAL_PRICE	<input type="checkbox"/>		
OP_ProLevel	<input type="checkbox"/>	Edm.String	0	0	60		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		PROC_LEVEL	<input type="checkbox"/>		
OP_CurrAgent	<input type="checkbox"/>	Edm.String	0	0	80		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		CURR_AGENT	<input type="checkbox"/>		
OP_Dummy	<input type="checkbox"/>	Edm.String	0	0	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		DUMMY_FIELD	<input type="checkbox"/>		
ErrorMsgs	<input type="checkbox"/>	Edm.String	0	0	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	T		ERROR_MSSGS	<input type="checkbox"/>		

Step8.

Now create an entity set and link the entity type to the entity set.

Name	Entity Type Name	Label	L	Semantics	C	U	D	P	A	S	S	R
SCHeaderCollection	SCHeader	SCHeaderCollection		C, U, D, P, A, S, S, R	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Step9.

Now go to Service Implementation and create a GetEntitySet (Query) operation

Operation Name	Data Source Group	Data Source Name	Data Source Type	RFC Destination	Implementation Class	Method Name
Create	ZGSTS_ITM	ZGSTS_ITM_GET_SC_HEADER	Remote Function Call	ZOL_ZGSTS_ITM_SC_	ZOL_ZGSTS_ITM_SC_	SCHEADERCOLLECTI_
Delete				ZOL_ZGSTS_ITM_SC_	ZOL_ZGSTS_ITM_SC_	SCHEADERCOLLECTI_
GetEntity (Read)				ZOL_ZGSTS_ITM_SC_	ZOL_ZGSTS_ITM_SC_	SCHEADERCOLLECTI_
GetEntitySet (Query)	ZGSTS_ITM	ZGSTS_ITM_GET_SC_HEADER	Remote Function Call	ZOL_ZGSTS_ITM_SC_	ZOL_ZGSTS_ITM_SC_	SCHEADERCOLLECTI_
Update				ZOL_ZGSTS_ITM_SC_	ZOL_ZGSTS_ITM_SC_	SCHEADERCOLLECTI_

Step10.

Do the mapping between entityset and RFC for both input and output fields.

The fields which are of range table type would appear with a [] sign.

The screenshot shows the SAP NetWeaver Gateway Service Builder interface. On the left, the project tree for 'ZGSTS_ITM_SC_PO_DISPLAY' is visible, with 'Service Implementation' expanded and 'GetEntitySet (Query)' selected. The middle pane shows the mapping of the 'GetEntitySet (Query)' operation for the 'SCHeaderCollection' entity set. The right pane shows the 'Data Source Parameter' table, which includes columns for 'Data Source Parameter' and 'Constant Value'. The 'Constant Value' column contains various SAP internal identifiers like IM_CATEGORY_ID, IM_COMP_CODE, etc.

Step11.

After all the mapping is completed, click on the “Generate” button (highlighted below in yellow).

The screenshot shows the SAP NetWeaver Gateway Service Builder interface again. The left pane shows the service structure with 'ZGSTS_ITM_SC_PO_DISPLAY' selected. The right pane displays a table titled 'Operations of entity set SCHeaderCollection' with four rows: 'Create', 'Delete', 'GetEntity (Read)', and 'GetEntitySet (Query)'. The 'GetEntitySet (Query)' row is highlighted in yellow, indicating it is the current operation being generated.

This will generate all the runtime artifacts for this service as shown below.

Name	Generated Artifact Type	Program ID	Object Type	Object Name
ZCL_ZGSTS_ITM_SC_PO_DI_DPC	Data Provider Base Class	R3TR	CLAS	ZCL_ZGSTS_ITM_SC_PO_DI_DPC
ZCL_ZGSTS_ITM_SC_PO_DI_DPC_E...	Data Provider Extension Class	R3TR	CLAS	ZCL_ZGSTS_ITM_SC_PO_DI_DP...
ZCL_ZGSTS_ITM_SC_PO_DI_MPC	Model Provider Base Class	R3TR	CLAS	ZCL_ZGSTS_ITM_SC_PO_DI_MPC
ZCL_ZGSTS_ITM_SC_PO_DI_MPC_E...	Model Provider Extension Class	R3TR	CLAS	ZCL_ZGSTS_ITM_SC_PO_DI_MP...
ZGSTS_ITM_SC_PO_DISPLAY_MDL	Registered Model	R3TR	IWMO	ZGSTS_ITM_SC_PO_DISPLAY_M...
ZGSTS_ITM_SC_PO_DISPLAY_SRV	Registered Service	R3TR	IWSV	ZGSTS_ITM_SC_PO_DISPLAY_S...
ZIF_ZGSTS_ITM_GET_SC_HEADER	Backend Operation Proxy	R3TR	INTF	ZIF_ZGSTS_ITM_GET_SC_HEAD...

Step12.

Now go to tcode **/IWFND/MAINT_SERVICE** in the Netweaver gateway system and search for the service.

Add Service

Get Services

Filter

Type	Technical Service Name	Service Description	External Service Name	Namespace	External Mapping ID
BEP	/IVBEP/EPN RETAIL SCENARIO SRV	1 /IVBEP/CL_EPM_RETAIL_S_DPC_EXT	EPN RETAIL SCENARIO_SRV	/IVBEP/	
BEP	/IVBEP/GW/DEMO	1 Service for Gateway NW Demo	GW/DEMO	/IVBEP/	
BEP	/IVBEP/MDX_SFLIGHT_AU	1 Service used for abap units	MDX_SFLIGHT_AU	/IVBEP/	
BEP	/IVBEP/RMTSAMPLEFLIGHT_2	1 OData Channel for SFlight Data Provider	RMTSAMPLEFLIGHT_2	/IVBEP/	
BEP	/IVBEP/SRV_BEP_BW_MDL_CMP	1 Carbon Dioxide Data for Flights	RMTSAMPLEFLIGHT_C02	/IVBEP/	
BEP	/IVBEP/SRV_BEP_BW_MDL_CMP	1 Combining BEP and BW models	SRV_BEP_BW_MDL_CMP	/IVBEP/	
BEP	/IVBEP/SRV_EXTERNAL_MDL_CMP	1 Combining two external BEP models - Composition	SRV_EXTERNAL_MDL_CMP	/IVBEP/	
BEP	/IVBEP/SUBSCRIPTIONMANAGEMENT	1 Technical Service name for Admin on-behalf Subs	SUBSCRIPTIONMANAGEMENT	/IVBEP/	
BEP	/IVBEP/SVC_SAMPLE_SFLIGHT_EXT	1 extension of sample flight	SVC_SAMPLE_SFLIGHT_EXT	/IVBEP/	
BEP	/IVBEP/TEA_TEST_APPLICATION	1 Test Application Service TEA	TEA_TEST_APPLICATION	/IVBEP/	
BEP	/IVC/ITSG_GW/IL SERVICE	1 Workflow GWIL (Generic Worklist) Runtime	GENERICWORKLISTPROCESSING	/IVC/ITSG/	
BEP	/IVEND/SG_ERROR_HANDLING	1 Error handling test application	RMTERRORHANDLINGTEST	/IVEND/	
BEP	/IVEND/SG_SAMPLE_RMT_MDL_COM	1 Model Group for Sample Composition Model ext.	RMTSAMPLEFLIGHTCOMPOSITION	/IVEND/	
BEP	/IVWRK/DUET_WORKFLOW_CORE	1 Workflow service for Duet Enterprise	DUET_WORKFLOW_CORE	/IVWRK/	
BEP	/IVWRK/WHUBSERVICE	1 Workflow pure ODC runtime	WHUBSERVICE	/IVWRK/	
BEP	/IVWRK/WF/WSERVICE	1 Workflow service for SP3 or higher	WF/WSERVICE	/IVWRK/	
BEP	ZGSTS_ITM_SC_PO_DISPLAY_SRV	1 ZCL_ZGSTS_ITM_SC_PO_DL_DPC_EXT	ZGSTS_ITM_SC_PO_DISPLAY_SRV		

Step13.

Double click on the service ZGSTS_ITM_SC_PO_DISPLAY_SRV to add and activate it.

Activate and Maintain Services

Service Catalog

Type	Technical Service Name	Service Description	External Service Name	Namespace
BEP	/IVC/ITSG_MATERIAL	1 Consumption Model for Material	MATERIAL	/IVC/ITSG/
BEP	/IVEND/SG_META_EXPLORATION	1 Gateway Service Repository	MetaRepository	/IVEND/
BEP	/IVEND/SG_MGW_TEA	1 Tea	MGWTEST	/IVEND/
BEP	/IVEND/SG_MGW_NOTIFICATION_STORE	1 ODATA Channel: Notification Store	NOTIFICATIONSTORE	/IVEND/
BEP	/IVEND/SG_PRODUCTS	1 Consumption Model for NW Demo Model Product	PRODUCT	/IVEND/
BEP	/IVEND/SG_REPORTING	1 Service Document for Reporting Management	REPORTING	/IVEND/
BEP	/IVEND/SG_SAMPLE_SFIGHT	1 OData Channel - Reference SFlight Data Provider	RMTSAMPLEFLIGHT	/IVEND/
BEP	/IVEND/SG_EPNSALESORDER	1 Service for NW Demo Model Sales Order	SALESORDER	/IVEND/
BEP	/IVEND/SG_SAMPLE_SFIGHT	1 Sample Flight Scenario	SAMPLEFLIGHT	/IVEND/
BEP	/IVEND/SG_SAMPLE_USER	1 Sample User Processing	SampleUserProcessing	/IVEND/
BEP	/IVEND/SG_SUBSCRIPTIONMANAGEMENT	2 MDC enabled Subscription Management Service	SUBSCRIPTIONMANAGEMENT	/IVBEP/
BEP	/IVEND/SG_SUBSTITUTION	1 Substitution service document	SUBSTITUTIONPROCESSING	/IVC/ITSG/
BEP	/IVEND/SG_TEST_APPLICATION	1 Test Application	TeaProcess	/IVEND/
BEP	/IVEND/SG_TEST_APPLICATION_STC	1 Test Application	TestProc	/IVEND/
BEP	/IVEND/SG_USER_SERVICE	1 Information Worker - User Service	USERSERVICE	/IVEND/
BEP	/IVC/ITSG_WF_ACTIVITY_TASK	1 Workflow dialog activity processing	WFACTIVITYTASKPROCESSING	
BEP	/IVC/ITSG_WF_DECISION_TASK	1 User Decision Workflow task processing	WDECISIONTASKPROCESSING	
BEP	/IVC/ITSG_WF_GEN_TASK	1 Workflow generic task processing	WFGENTASKPROCESSING	
BEP	/IVC/ITSG_ODM_RMT_WORKFLOW	1 Workflow for minimale Gateway-Servicegruppe	WFODCPPROCESSING	/IVC/ITSG/
BEP	ZGSTS_ITM_SC_PO_DISPLAY_SRV	1 ZCL_ZGSTS_ITM_SC_PO_DL_DPC_EXT	ZGSTS_ITM_SC_PO_DISPLAY_SRV	
BEP	ZGSTS_MAPP_PARTICIPANT_SEARCH_SRV	1 ZCL_ZGSTS_MAPP_PARTICIPANT_DPC_EXT	ZGSTS_MAPP_PARTICIPANT_SEARCH_SRV	

System Aliases

SAP System Alias	Description	User Role	Default	Host Name
SAP_SRM	SYSTEM ALIAS FOR DM1 300			

ICF Nodes and System Aliases are populated. Now the service is activated.

Now we can construct queries in order to retrieve the shopping cart information using this OData service (few examples given below):

[/sap/opu/odata/sap/ZGSTS_ITM_SC_PO_DISPLAY_SRV/SCHeaderCollection?
\\$filter= IN_ShopCartId eq '1000045468' and IN_IsidRequestor eq 'USMANAGER6'](https://sap/opu/odata/sap/ZGSTS_ITM_SC_PO_DISPLAY_SRV/SCHeaderCollection?filter=IN_ShopCartId eq '1000045468' and IN_IsidRequestor eq 'USMANAGER6')

[/sap/opu/odata/sap/ZGSTS_ITM_SC_PO_DISPLAY_SRV/SCHeaderCollection?
\\$format=json&\\$top=10&\\$filter= IN_ShopCartId eq '1000045468'and IN_IsidRequestor eq 'USMANAGER6'](https://sap/opu/odata/sap/ZGSTS_ITM_SC_PO_DISPLAY_SRV/SCHeaderCollection?format=json&$top=10&$filter= IN_ShopCartId eq '1000045468'and IN_IsidRequestor eq 'USMANAGER6')

`/sap/opu/odata/sap/ZGSTS_ITM_SC_PO_DISPLAY_SRV/SCHeaderCollection?
$filter= IN_IsidRequestor eq 'USEMPLOYEE15' and IN_CreateDate eq
datetime'2013-12-01T00:00:00' and IN_CreateDateTo eq
datetime'2013-12-31T00:00:00'`

`/sap/opu/odata/sap/ZGSTS_ITM_SC_PO_DISPLAY_SRV/SCHeaderCollection?
$filter= IN_IsidRequestor eq 'USEMPLOYEE15' and (IN_CategoryId eq
'10100000' or IN_CategoryId eq '43211507')`

How to find Web Service WSDL in SOA Manager - ABAP Connectivity - Support Wiki

Thursday, August 24, 2023 12:28 PM

Clipped from:

<https://wiki scn sap com/wiki/display/ABAPConn/How+to+find+Web+Service+WSDL+in+SOA+Manager>

SAP Support Wiki Spaces Search Log in

Pages / ABAP Connectivity Home / ABAP Connectivity - Web Services ABAP

How to find Web Service WSDL in SOA Manager

Created by Harish Mehta, last modified by Felipe Zanettini on Mar 30, 2018

Contents

- How to find Web Service WSDL in SOA Manager
 - Prerequisites
 - Procedure

How to find Web Service WSDL in SOA Manager

This article describes how to find WSDL for a Web Service in ABAP System via SOA Manager.

Prerequisites

Web Service should be active in the ABAP Workbench (SE80) before it's WSDL is retrieved in SOA Manager.

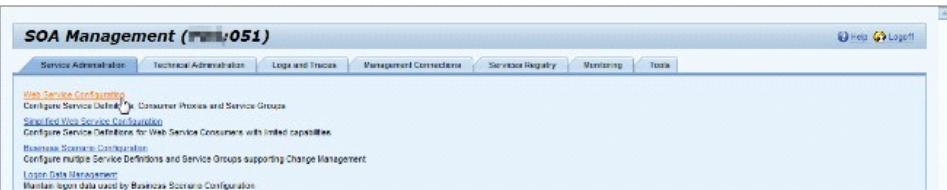
WSDL containing bindings is available only if the respective Web Service is first configured in SOA Manager.

You need to have the following roles to be able to extract WSDL for a Web Service:
SAP_BC_WEBSERVICE_CONFIGURATOR
More information: SAP Note 1318883 (Introduction of new authorizations in Web Service)

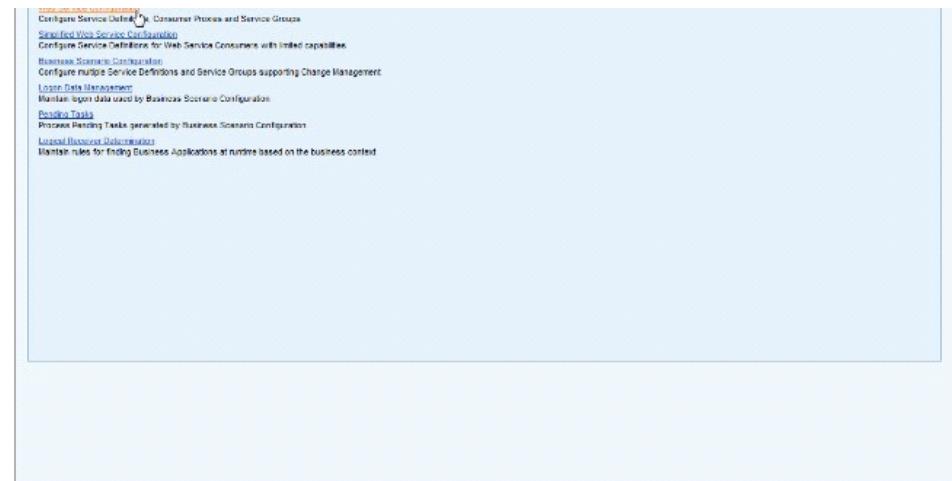
[Back to top](#)

Procedure

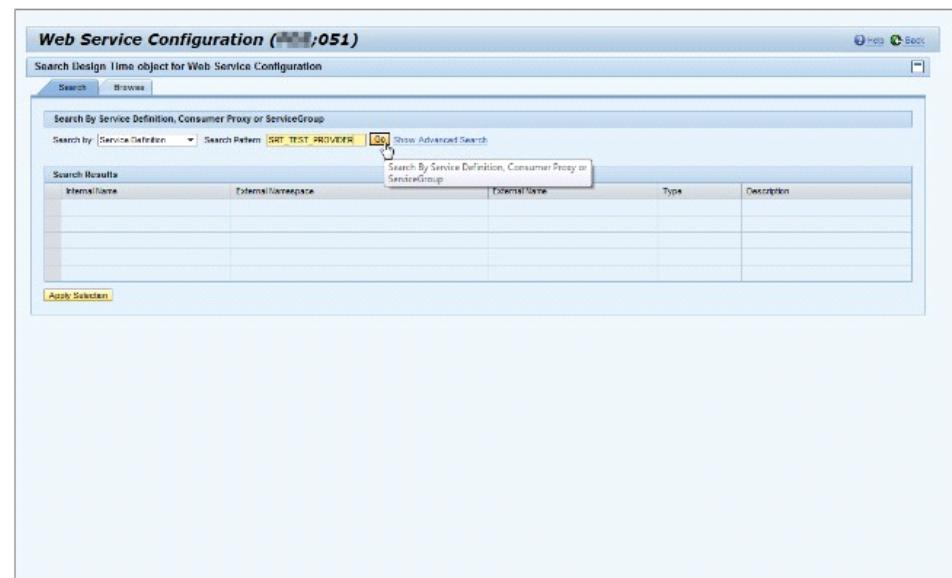
1. Start SOA Manager.
Use transaction SOAMANAGER. SOA Manager is opened in a new window.
- The SOA Manager is an ABAP WebDynpro application. It is like each regular WD application using an ICF node for accessing it. You may use transaction SICF to check that the ICF node for SOA Manager transaction (/sap/bc/webdynpro/sap/appl_soap_management) is active.
More information about this subject: SAP Note 1124553 and 1088717.
2. Go to the **Service Administration** tab.
3. Select **Web Service Configuration**.



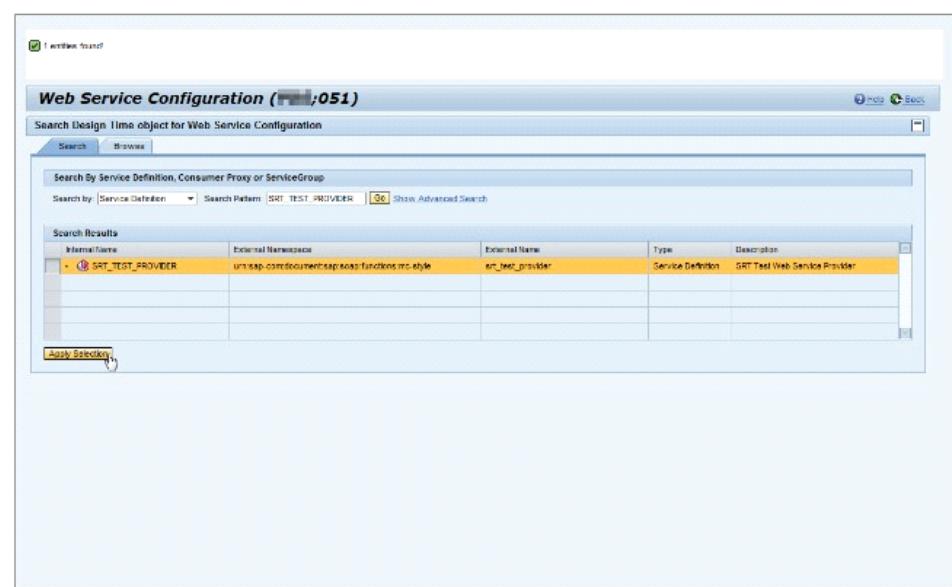
- How to set up ABAP Services Registry
- How to set up a Connection and Function
- How to use Logical Receiver Definitions
- How to use Technical Receiver Definitions
- Migration of Web Service Configuration
- Plain SOAP
- Proxy Features Downport 2011
- SOAManager Documentation via SAP Help
- Technical and Logical Receiver Definitions
- Technical Setup and Reset of the Web Services
- Troubleshooting Guide - Web Services
- Using Reverse Proxies
- Web Service Runtime Monitor
- Web Service Testing Using the J2EE Container
- BAdI Builder
- Basic Knowledge Sharing in APJ
- Causes of translation issue in Fiori Launchpad
- CMS Server Is not coming up for Data Services
- Create Currency Conversion issue at the Fiori Launchpad
- Enable In-House Cash Bank Payments
- File lists
- Home office network prerequisites configuration
- how to create users for testing with Fiori Launchpad
- How to get the start up information
- Layout Test
- Omnidfilter for tables
- Possible to print header/footer on each page
- SMTP Enhancements in 4.2 SP04
- Support Information index in local language
- Test Space
- Wiki Image Page for BYD Comms
- ABAP Connectivity - Application Integration
- PostgreSQL ODBC with EnterpriseConnect
- Other Pages



4. Select option **Service Definition** from dropdown **Search by** and Specify the Web Service name. Press **Go** to search for the Web Services.



5. Select the desired Web Service in the result set and choose **Apply Selection**.



6. Go to the **Overview** tab

6. Go to the **Overview** tab.

7. Select **Open Design Time WSDL document** to generate Design time WSDL for this Web Service. New browser window will be opened and fresh WSDL will be generated.

ⓘ This WSDL does not contain any Bindings. This Web Service WSDL should be identical to the WSDL seen in the Web Service development workbench (SE80).

The screenshot shows the SAP Web Service Configuration interface. In the search results table, there is one entry for 'SRT_TEST_PROVIDER'. The details view for this provider shows the 'General Attributes' section. Under 'Select Binding', the dropdown menu is open, listing several options including 'HTTP', 'HTTPS', 'A', and 'HTTP/HTTPS'. Below the dropdown, there is a note: 'Binding WSDL is generated only if the Web Service is already configured and has at least one binding.' There is also a link to 'How to configure a Service Provider'.

8. a. Under release SAP_BASIS 740 choose the desired binding from the dropdown **Select Binding** and select **Open WSDL document for selected binding or service** to generate binding WSDL for the selected binding.

This will generate a Monolithic WSDL containing policies for the selected binding.

⚠ Binding WSDL is generated only if the Web Service is already configured and has at least one binding.

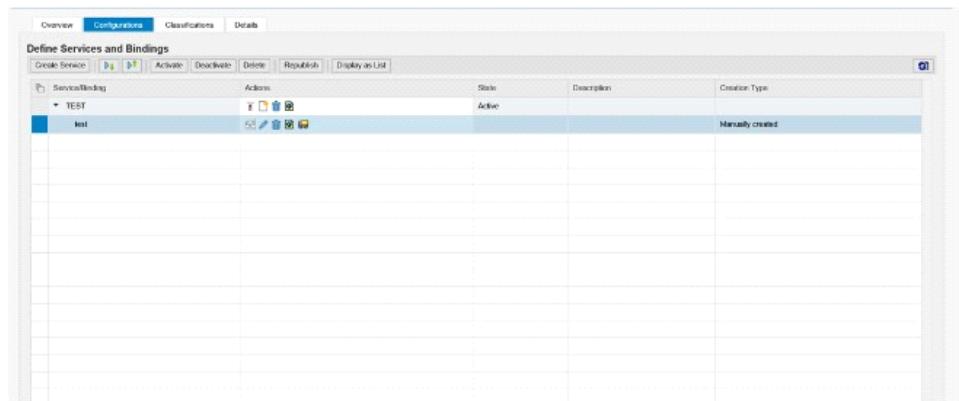
More Information : [How to configure a Service Provider](#)

The screenshot shows the SAP Web Service Configuration interface. In the search results table, there is one entry for 'SRT_TEST_PROVIDER'. The details view for this provider shows the 'General Attributes' section. Under 'Select Binding', the dropdown menu is open, listing several options including 'HTTP', 'HTTPS', 'A', and 'HTTP/HTTPS'. Below the dropdown, there is a note: 'Binding WSDL is generated only if the Web Service is already configured and has at least one binding.' There is also a link to 'How to configure a Service Provider'.

b. From release SAP_BASIS 740 the layout looks different in SOAMANAGER

Select the icon to open the WSDL generation wizard of the binding WSDL file

Select the  icon to open the WSDL generation wizard of the binding WSDL file



Choose the  icon to generate the WSDL file

WSDL Generation for Binding: test

WSDL Flavours

Flavour:	All possible combinations are allowed
SAP Assertions:	All
Security Assertions:	AHAP 700/710/740 and higher
WSDL Section:	AllInOne
WSDL Version:	11
WSP Version:	1.2
WSP Style:	single-Binding
SOAP Version:	SOAP 1.1 and SOAP 1.2
SOAP Action:	With
SOAP Style:	Document

Options for WSDL Access and URLs

Standard
 Alternative URL
 Alternate Host:
 Meta Data Protocol:
 Alt Port (http):
 Alt Port (https):

WSDL Generation

WSDL URL for Binding: [https:///sap/bc/srt/wsdl/fly_10002A111AD1/bndg_urtsap/bc/srt/xip/sap/zzzsv000/test/test?sap-client=000](https://<your_host>/sap/bc/srt/wsdl/fly_10002A111AD1/bndg_urtsap/bc/srt/xip/sap/zzzsv000/test/test?sap-client=000)

9. More specific WSDL can be generated by expanding the tray **WSDL Generation**.

Web Service Configuration (SRT-051)

Search Design Time object for Web Service Configuration

Search By Service Definition, Consumer Proxy or Service Group

Search by: Service Definition | Search Pattern: SRT test provider | Show Advanced Search

Search Results

Internal Name	External Namespace	External Name	Type	Description
SRT_TEST_PROVIDER	urn:sap:com:document:sap:functions:mc-style	ext_web_provider	Service Definition	SRT Test Web Service Provider

Details of Service Definition: SRT_TEST_PROVIDER

General Attributes

Configuration Status: Enabled | Bindings: 2
 External Namespace: urn:sap:com:document:sap:functions:mc-style
 External Name: ext_web_provider
 Internal Name: SRT_TEST_PROVIDER
 Internal Namespace: urn:sap:com:document:sap:functions:mc-style
 Package Name: SD4I_CMCIC
 Software Component: G01_PMSR
 Application Component: BC4EWWSABA
 Selected Binding: [SRT Test provider \(HTTP, document\)](#)
[Open WSDL document for selected binding or service](#)
[Open Web Service navigator for selected binding](#)
[Open Test Documenter](#)
[Show / Hide selected Endpoint or Services WSDL URL](#)

WSDL Generation

Choose the desired WSDL option from this tray and select **Open WSDL document for selected binding or service** again to regenerate the WSDL.

a. Choose option **Monolithic WSDL** to **False** if fragmented WSDL is desired

Choose the desired WSDL option from this tray and select **Open WSDL document for selected binding or service** again to regenerate the WSDL.

- a. Choose option **Monolithic WSDL** to **False** if fragmented WSDL is desired.
- b. Choose option **WSDL Format** to **Standard** if WSDL without policies is desired.
- c. Choose option **WSDL style** to **RPC** if RPC style WSDL is desired.
- d. Choose option **WSDL Document Type** to **All bindings of Service** if WSDL containing all bindings associated to a WSDL service is desired.
- e. Choose option **Binding SOAP Version** to control the SOAP version of WSDL. This setting takes effect if **WSDL Document Type** is set to **Selected Binding Only**.
- f. Choose **Options for WSDL Access and URLs** to influence the WSDL and Binding URL i.e. host, port, protocol and path prefix. This can be desired if a Reverse Proxy is in place during the Web Service communication.

ⓘ Content of WSDL files are generated dynamically, depending on the parameters in the WSDL's URL path. Changing above settings will change these parameters. If you set the desired settings, then save or publish the generated WSDL URL, the same content will be returned each time this address is called - independently from the SOAMANAGER settings.

No labels

«

[Privacy](#) [Terms of Use](#) [Legal Disclosure](#) [Copyright](#) [Trademark](#) [Cookie Preferences](#)

OData V4 code based implementation – Overview | SAP Blogs

Sunday, October 9, 2022 9:25 AM

Clipped from: <https://blogs.sap.com/2017/12/12/odata-v4-code-based-implementation-overview/>

This blog is meant as an introduction of a series of blogs in which I will explain the use of the new SAP Gateway V4 framework.

It is meant for those readers that must create OData V4 series now and that cannot wait until an end-to-end support for OData V4 will be available through the new ABAP programming model.

Before starting code based OData V4 development you should check my blog [OData service development options](#) where I outline in more detail what the recommended options for OData development are right now.

Updates

- - 13.12.2017 – added link to the first how to guide and the blog that explains the OData service development options in more detail
 - 19.05.2021 – published the source code on [GitHub](#)

Table of contents

This blog is part of blog series about OData V4 code based development

[OData V4 code based implementation – Overview](#)

[OData V4 code based implementation I \(basic interface, read access\)](#)

[OData V4 code based implementation I \(basic interface, create&update\)](#)

Demo system ES5

The service has been deployed in the ES5 demo system

[https://sapes5.sapdevcenter.com/sap/opu/odata4/sap/ze2e001/default/sap/ze2e001_salesorder/0001/\\$metadata?sap-client=002&sap-language=EN](https://sapes5.sapdevcenter.com/sap/opu/odata4/sap/ze2e001/default/sap/ze2e001_salesorder/0001/$metadata?sap-client=002&sap-language=EN)

In order to access the service you have to

[Sign up for a demo account on ES5 here](#)

More details about the ES5 demo system as such you will find in my following blog

[New SAP Gateway Demo System available](#)

Source code

The source code can be found on GitHub.

[Data provider class – zcl_e2e001_odata_v4_so_data](#)

[Model provider class – zcl_e2e001_odata_v4_so_model](#)

[Exception class – zcx_e2e001_odata_v4_so](#)

[Interface – zif_e2e001_odata_v4_so_types](#)

[Consumption view – sales order – ze2e001_c_salesorder](#)

[Consumption view sales order items – ze2e001_c_salesorderitem](#)

[Interface view – ze2e001_i_salesorderitem_e](#)

[Consumption view – ze2e001_c_salesorder](#)

What's new in the protocol?

The main paradigm: Reduction of data

The main paradigm of the OData V4 paradigm is the reduction of data. This reduction is achieved through a more powerful query language and a new optimized JSON protocol. At the same time it is possible to leverage richer metadata as compared to OData V2.

New JSON format

The OData V4 protocol comes with a very lean JSON protocol. The response payload now basically only contains name value pairs. The metadata has been reduced to a single line

"[@odata.context](#)" : "\$metadata#SalesOrder/\$entity"

as opposed to the more richer metadata information in the V2 response payload, both highlighted in blue in the following figure.

V2 PAYLOAD PRODUCT DETAILS

```

d: {
  __metadata: {
    id: "https://1dailigly.wdf.sap.corp:44356/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/ProductSet('HT-1000')",
    uri: "https://1dailigly.wdf.sap.corp:44356/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/ProductSet('HT-1000')",
    type: "IWBEP/GWSAMPLE_BASIC.Product",
    etag: "W/\"datetime'2015-10-22T09:43:25.000000Z\""
  },
  ProductID: "HT-1000",
  TypeCode: "PC",
  Category: "Notebooks",
  Name: "Notebook Basic 15",
  Description: "Notebook Basic 15 with 2,00 GHz quad core, 15'' LCD, 4 GB DDR3 RAM, 500 GB Hard Disc, Windows 8 Pro",
  DescriptionLanguage: "EN",
  SupplierID: "0100000000",
  SupplierName: "SAP",
  TaxTariffCode: 1,
  MeasureUnit: "EA",
  WeightMeasure: "4.200",
  WeightUnit: "KG",
  Length: "20.000",
  Price: "994.000",
  Width: "30.000",
  Depth: "18.000",
  Height: "2.000",
  DiaIncht: "0.00",
  CreatedAt: "Date(1445507005000)/",
  ChangedAt: "Date(1445507005000)/",
  ToSalesOrderLineItems: [
    __deferred: {
      uri: "https://1dailigly.wdf.sap.corp:44356/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/ProductSet('HT-1000')/ToSalesOrderLineItems"
    }
  ],
  ToSupplier: [
    __deferred: {
      uri: "https://1dailigly.wdf.sap.corp:44356/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/ProductSet('HT-1000')/ToSupplier"
    }
  ]
}

```

© 2017 SAP SE or an SAP affiliate company. All rights reserved. | EXTERNAL

```

@odata.context: "metadata#ProductList/$entity",
ProductID: "HT-1000",
TypeCode: "PC",
Category: "Notebooks",
Name: "Notebook Basic 15",
DescriptionLanguage: "EN",
Description: "Notebook Basic 15 with 2,00 GHz quad core, 15'' LCD, 4 GB DDR3 RAM, 500 GB Hard Disc, Windows 8 Pro",
SupplierID: "1000000000",
SupplierName: "SAP",
TaxTariffCode: 1,
MeasureUnit: "EA",
WeightMeasure: 4.2,
Width: "30.000",
Depth: "18.000",
Height: 3,
DiaIncht: "0.00",
CreatedAt: "2015-10-22T09:43:25Z",
ChangedAt: "2015-10-22T09:43:25Z"

```

V4 PAYLOAD PRODUCT LIST

Cross service references

Cross service navigation enables inter communication of services. With this, navigation properties of entities of one service can reach entities of another service in a service group. With the support of cross service navigation several requirements of SAP Fiori like applications can be addressed.

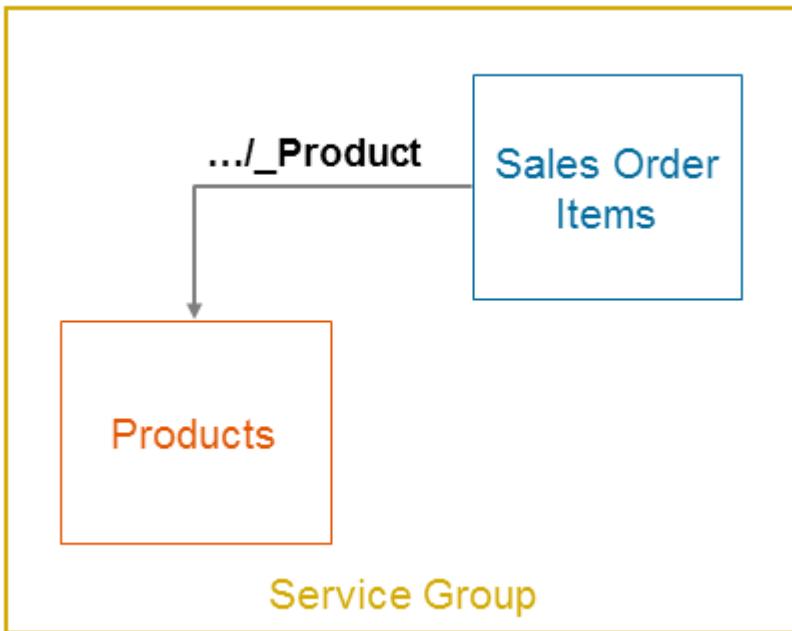
1.) The rich metadata can be leveraged but at the same time not the complete data model has to be loaded at startup time of an application. There is rather the option to have a lazy loading of parts of a service model on demand.

2.) Services can be reused more easily since services can be partitioned without the loss of navigation.

Examples of such services that can be reused in various SAP Fiori applications are Users, Attachments, Conditions, Addresses, ...

Please note:

Cross service references are only possible within one service group.



If a request like the following is issued:

```
.../SalesOrderItems(SalesOrder='500000000', SalesOrderItem='10')/
 _Product
```

one would receive a response like the following where it is indicated via the @odata.context annotation that this response stems from another service.

```
{
  "@odata.context" : ".../sap/Product/0001/$metadata#Product/$entity",
  "value" : [
    {
      "Product" : "HT-1000",
      "Currency" : "EUR",
      ...
    }
  ]
}
```

Support for Any and All

New is the support of the query options any and all.

With these it is now possible to find all sales orders where at least one of the items contains a particular product

```
.../SalesOrders?$filter = _Item/any(d:d/Product eq 'HT-1007')
```

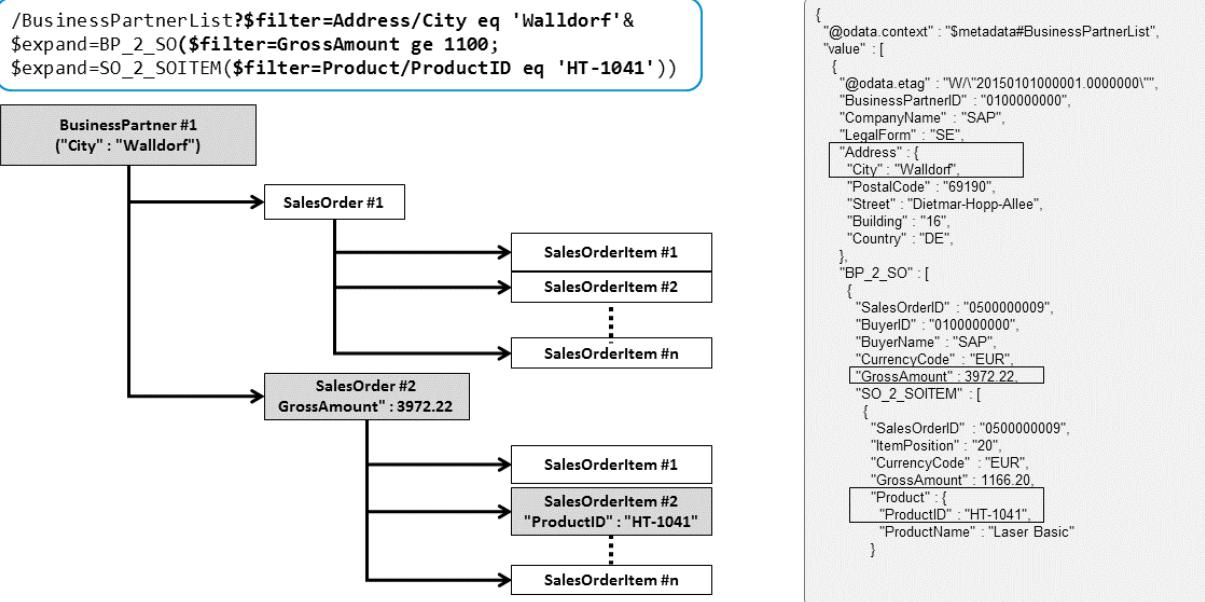
or it is possible to find all sales orders where every item has a price greater than \$100

```
.../SalesOrders?$filter = _Item/all(d:d/Price ge 100)
```

Filter on expanded result sets

New in OData V4 is the option for filtering on each level of an expanded entity set. In the following figure you see a request that

1. reads all business partners that are located in 'Walldorf'
2. reads all sales orders of those business partners where the gross amount exceeds 1100 Euro
3. reads only those items that contain the product HT-1041.



What's new in the framework?

Advanced, intermediate and basic interfaces

The API that is used to develop OData V4 services has significantly changed compared to the API that is used to develop OData V2 services.

When you implement the methods of the basic interface you will get a working OData V4 service that will satisfy most requests. Complex requests such as \$expand are then handled by the framework that will call the methods for the basic interface in the correct order.

The implementation of the intermediate or advanced interface should however also be taken into account if your service implementation would be able to handle specific requests such as specific \$expand or navigation calls more efficient than by calling the methods of the basic interface recursively.

Name	Purpose
/IWBEPIF_V4 _DP_BASIC	<ul style="list-style-type: none">• Methods provide basic functionality• (Create, Update, Delete, Navigation, ...)• When being implemented à Working OData service

	supporting most requests
/IWBEPIF_V4 _DP_INTERMEDIATE	<ul style="list-style-type: none"> • Medium complex functionality • eTag handling, PATCH, \$expand • Contains generic calls to other (especially the basic) interfaces
/IWBEPIF_V4 _DP_ADVANCED	<ul style="list-style-type: none"> • Always called first by the framework • Contains generic calls to the other (especially the basic) interfaces • Will for example be overwritten by the new RESTful ABAP Programming model (planned)
/IWBEPIF_V4 _DP_BATCH	<ul style="list-style-type: none"> • \$batch. Generic \$batch and changeset
/IWBEPIF_V4 _DP_PROCESS_STEPS	<ul style="list-style-type: none"> • Transaction and lifecycle handling

[io_request and io_response](#)

All interface methods have an import parameter called **io_request**. It can be used to retrieve all information you need to handle the request in your service implementation.

A UPDATE_ENTITY method for example will have the following methods

- GET_BUSI_DATA to retrieve entity data from the request, for example the payload of the incoming request.
- GET_ENTITY_SET to retrieve the entity set of the processed entity. So we can switch to entity set specific methods

The corresponding parameter ip_response is used to return business data to the SAP Gateway framework and to tell the framework which processing steps the service implementation has handled itself (see todo and done flags below).

[Generic framework support – example \\$expand](#)

As already mentioned you will get a working OData V4 service by only implementing the methods of the basic interface. If a client calls the following URL:

```
GET .../ze2e001_salesorder/0001/SalesOrder('50000000')?  
$select=Salesorder,Customer&  
$expand=_Item($select=Salesorderitem,Product,Grossamountintransaccurre  
ncy,Transactioncurrency)
```

the SAP Gateway framework will call the following basic methods in your service implementation:

#	Method call	Purpose
---	-------------	---------

- 1 ..._BASIC~READ_E This method will read the data of the sales order
NTITY header
- 2 ..._BASIC~READ_R This method will read the key fields of the items
EF_TARGET_KEY_D that can be used by the READ_ENTITY_LIST
ATA method as a \$filter statement
- 3 ..._BASIC~READ_E This method reads the items.
NTITY_LIST

and would finally return the following data

```
{
  "@odata.context" : "$metadata#SalesOrder(Salesorder,Customer,
  _Item(Salesorderitem,Product,Grossamountintransaccurrency,Transactionc
  urrency,Salesorder))/$entity",
  "Salesorder" : "500000000",
  "Customer" : "100000000",
  "_Item" : [
    {
      "Salesorder" : "500000000",
      "Salesorderitem" : "10",
      "Product" : "HT-1000",
      "Transactioncurrency" : "EUR",
      "Grossamountintransaccurrency" : 1137.64
    },
    {
      "Salesorder" : "500000000",
      "Salesorderitem" : "20",
      "Product" : "HT-1001",
      "Transactioncurrency" : "EUR",
      "Grossamountintransaccurrency" : 2972.62
    },
    ...
  ]
}
```

Please note:

With OData V4 now query options are supported on all levels of an \$expand statement.

[ToDo and Done-Flags](#)

The SAP Gateway V4 framework has introduced so called ToDo-Flags which provide a hint for the application developer what his implementations has to do. Depending on the query options that have been used in the request you will get simple list with boolean values for the following flags:

deltatoken, select, filter, skip, orderby, skiptoken, search, top, ...

Done-Flags confirm that the response fits to the request. They allow

the application developer to inform the framework to handle feature generically e.g., \$top, \$skip, and \$select. Using such flags also allows an implementation to be compatible in the future. Instead of a wrong result an exception will be raised if a done flag is not set.

The list of todo and done flags will vary depending on the method which is called. (READ; READ_LIST, CREATE, ...)

For a simple GET request with a \$filter query option:

```
.../SalesOrder?$filter=Customer eq ,SAP
```

a service implementation would have to look like as follows.

At the beginning of our service implementation we have to check whether we have to handle the filter option. For this we call the method *io_request->get.todos*. Then we have to check whether the flag *ls_todo_list-process-filter* is set. If yes, the filter string is requested via the method *io_request->get_filter_osql_where_clause* and the flag that we have handled the filter query option is set in the structure *ls_done_list*. This information is at the end sent back to the framework via the method *io_response->set_is_done* that takes the done-list as a parameter.

```
io_request->get.todos( importing es_todo_list = ls_todo_list ).  
...  
if ls_todo_list-process-filter = abap_true.  
    io_request->get_filter_osql_where_clause( importing  
    ev_osql_where_clause = lv_where_clause ).  
    ls_done_list-filter = abap_true.  
endif.  
...  
" Report list of request options handled by application  
io_response->set_is_done( ls_done_list ).
```

Supported OData System Query Options | SAP Help Portal

Sunday, October 9, 2022 9:30 AM

Clipped from:

https://help.sap.com/docs/SAP_HANA_PLATFORM/691cb949c1034198800afde3e5be6570/54fad3bd725141d083d2a48b674bdd86.html?version=2.0.03

The screenshot shows a web page from the SAP HANA Search Developer Guide. At the top right, there are navigation links for 'Previous', 'Favorite', 'Download PDF', 'Share', and 'Next'. Below the header, the page title is 'Developer Guide 2.0 SPS 03▼'. On the left, a sidebar lists various search-related topics. The main content area is titled 'Supported OData System Query Options'. It contains a table with two columns: 'System Query Option' and 'Description'. The first row shows '\$orderby' with a note explaining it can only be used if the search is limited to a single view. The second row shows '\$select' with a note explaining it defines the columns returned in the JSON response. Both rows have a note at the bottom.

System Query Option	Description
\$orderby	<p>Sort the response by any column in an ascending or descending order.</p> <p>Example: \$orderby=Search.score() desc</p> <p>\$orderby=CityName</p> <p>\$orderby=CityName asc, Name1 desc</p> <p>\$orderby=CityName, Search.score() desc</p> <p>Note \$orderby=Search.score() can only be used if the search is limited to a single view.</p>
\$select	<p>Defines the columns that are returned in the JSON response. These have to be a subset of these columns.</p> <p>Note \$orderby can only be used if the search is limited to a single view.</p>

Definition of the User Language
Writing the Search Response to a Table
Defining a Search Timeout
□ Full-Text Search with SQL
Creating Search UIs With SAPUI5
SAP File Processing for SAP HANA
SAP HANA File Loader
Important Disclaimer for Features in SAP HANA Platform

have to be a subset of these columns.

□ Note

`$select` restricts the search scope to views fails with an error.

Example: `$select=ID,Name1,Name2`

`$filter`

□ Note

A subset of the OData `$filter` expressions parameter to the `$filter` option. Instead of expression.

Filter condition supports only a subset of the Boolean operators:

- logical operators (eq, ne, gt, ge, lt, le, and, or)
- primitive literals as operands: property name, value
- search specific functions: `Search.search()`, `Search.filter()`

Arithmetic operators, canonical functions, path expressions, and other operators are not supported.

Example:

Preferred to have everything in a query language.

Not recommended, but also working: `$filter=someValue`

`$top`

Defines the number of items returned in the response.

Example:

`$top=10`

`$skip`

Defines how many items are skipped in the response.

Example:

`$skip=20`

`$count`

If `$count=true`, the total number of matching items is returned in the response. This is a custom query option `estimate`. When the `@Ent` extension is used, it counts the rows in the database view.

Example:

`$count=true&estimate=true`

`$apply`

`$apply` can be used instead of `$filter` to define a group operation.

It allows to define a `groupby` operation to group items by a specific property. It is also possible to get the number of distinct anchor items.

The query options `$top` and `$skip` are applied to the result of the `$apply` operation.

Examples:

□ Sample Code

Sample Code

```
/* $apply with a filter only */  
  
/$all?$top=3&$apply=filter(Search
```

Sample Code

```
/* this gives the same results as  
  
/$all?$top=3&$filter=Search.search
```

Sample Code

```
/* $apply with groupby */  
  
/$all?$top=3&$apply=filter(Search  
  
/$all?$top=3&$apply=filter(Search
```

Sample Code

```
/* groupby with orderby */  
  
/$all?$top=3&$orderby=instCity de:
```

Sample Code

```
/* $apply with groupby and aggregate */  
  
/$all?$top=3&$apply=filter(Search  
  
/$all?$top=3&$apply=filter(Search
```

Note

Limitations of \$apply

\$apply with groupby can only be used if the query is mapped to a single view (in the query language to a single view). A federated query is not supported.

When \$apply is used, the following query options are not supported:

- \$select. The columns returned are defined by the \$apply operation.
- \$filter. The filter is already given in \$apply.
- facets
- facetlimit
- wherefound
- whyfound

It is also not possible to order the result set by \$orderby.

■ whyfound

It is also not possible to order the result set by

□ Note

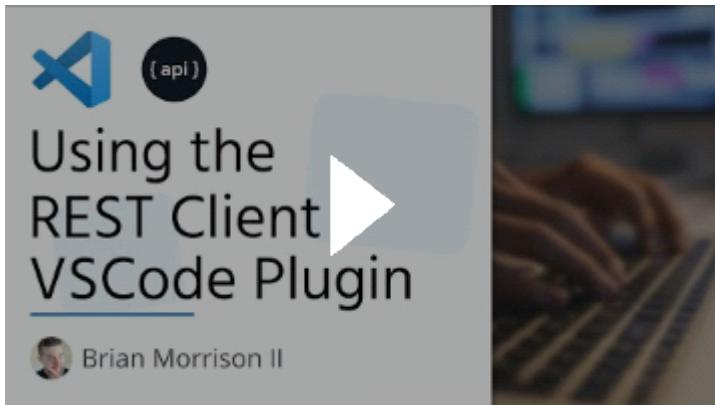
The following OData v4 system query options are not supported:

- \$expand: Associations to other entity sets are not available, so there is nothing to be expanded.
- \$format: The JSON format is always returned.
- \$search: The `search.search()` function is used instead.

REST client in VSCode

Thursday, October 20, 2022 7:01 AM

[Using the REST Client VSCode Plugin](#)



A Step by Step process to create Odata services in SAP / SAP HANA system | SAP Blogs

Thursday, April 13, 2023 4:49 PM

Clipped from: <https://blogs.sap.com/2021/05/06/a-step-by-step-process-to-create-odata-services-in-sap-sap-hana-system/>

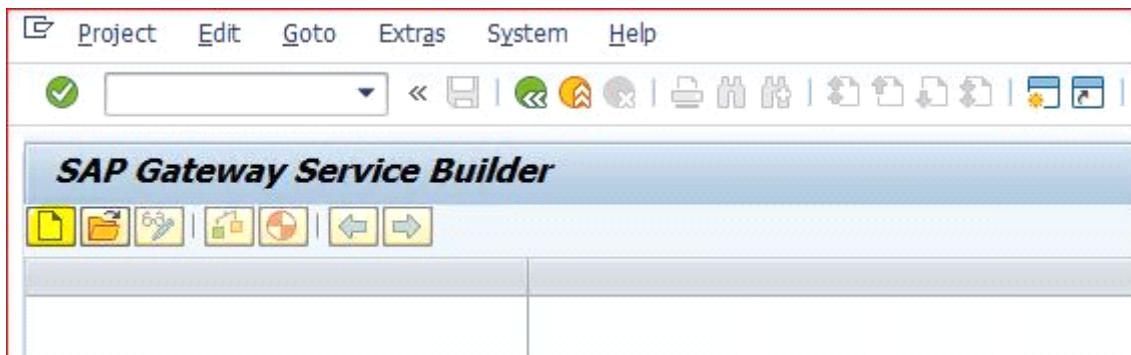
Hello,

While creating Odata services, I had gone through multiple blogs post and found that there are few steps which are not self explanatory and hence at the end of creating an odata service we get bad HTTP response . This is mainly due to missing explanation for how to load meta data and how to test the entity set data .

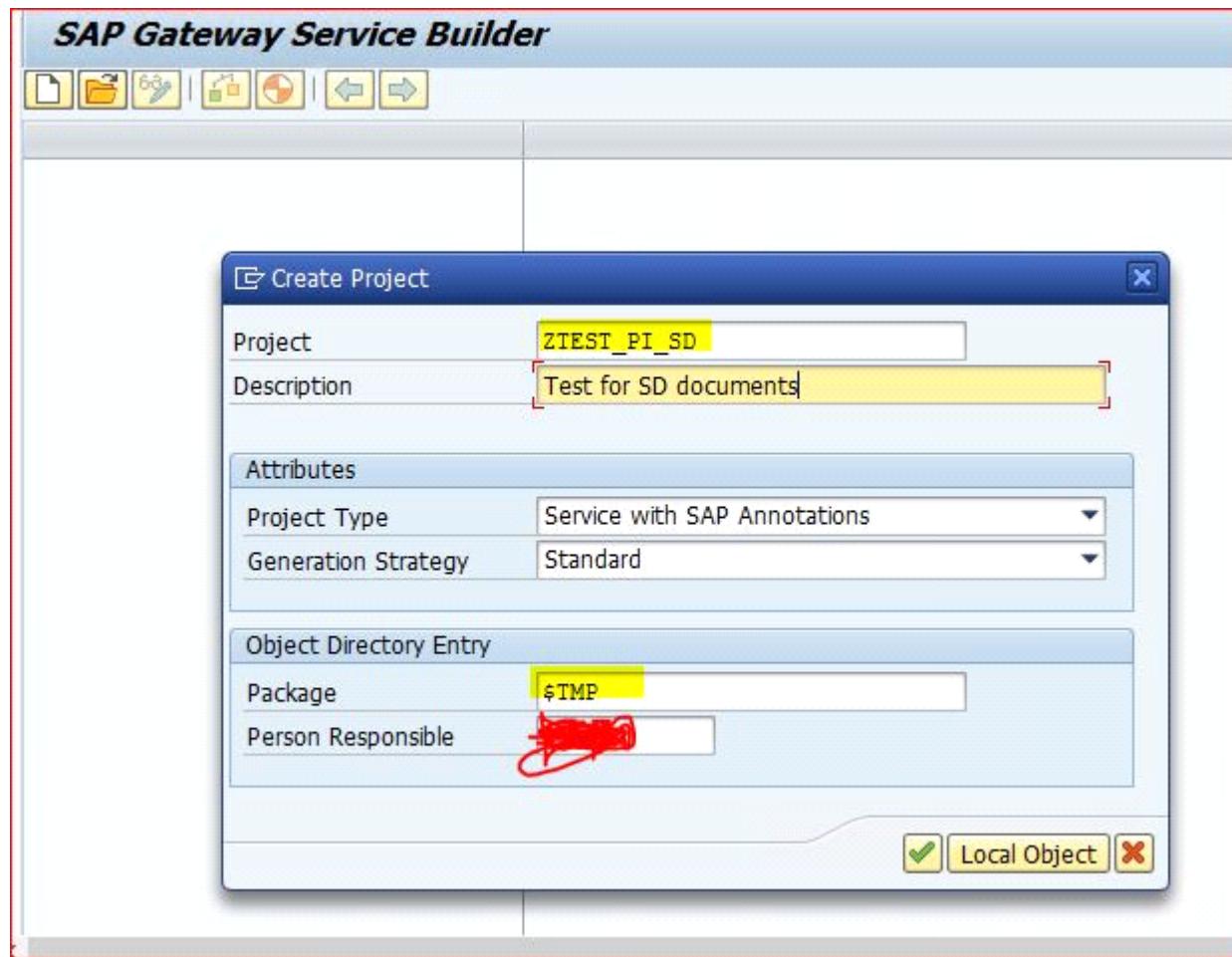
In this blog post, I am going to tell you step by step process to create Odata services for SAP /SAP HANA system and how to test it.

I tried to explain the whole process in 4 steps.

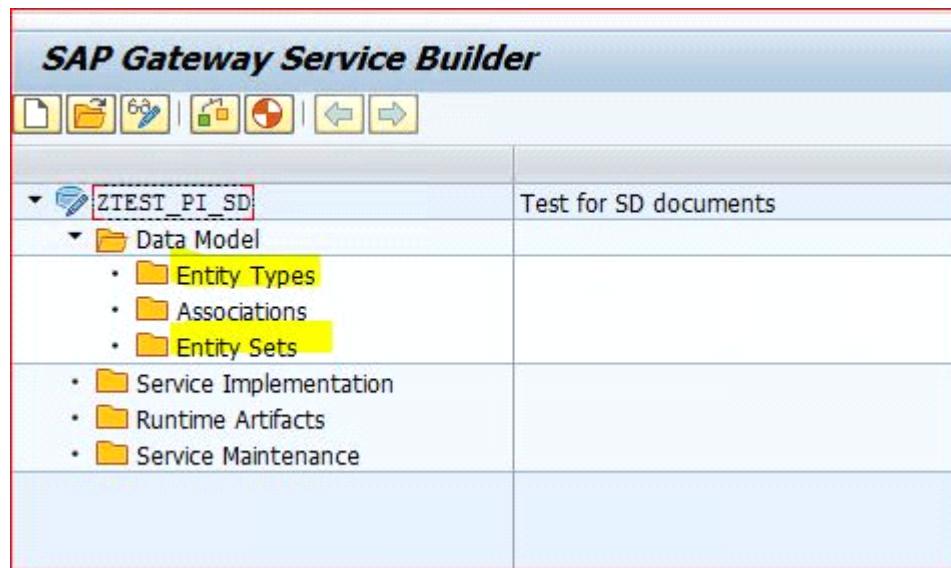
Step 1. Go to transaction code – SEGW .



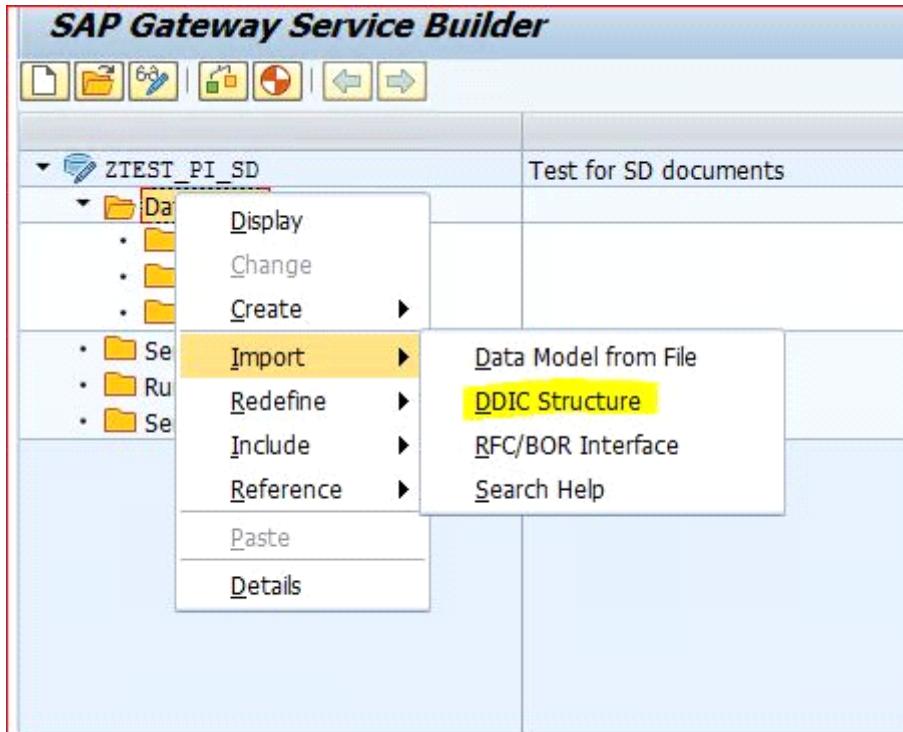
Click on Icon Create. A pop window will appear , Fill the details as per below mention in screen shot and click on check icon or enter.



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.

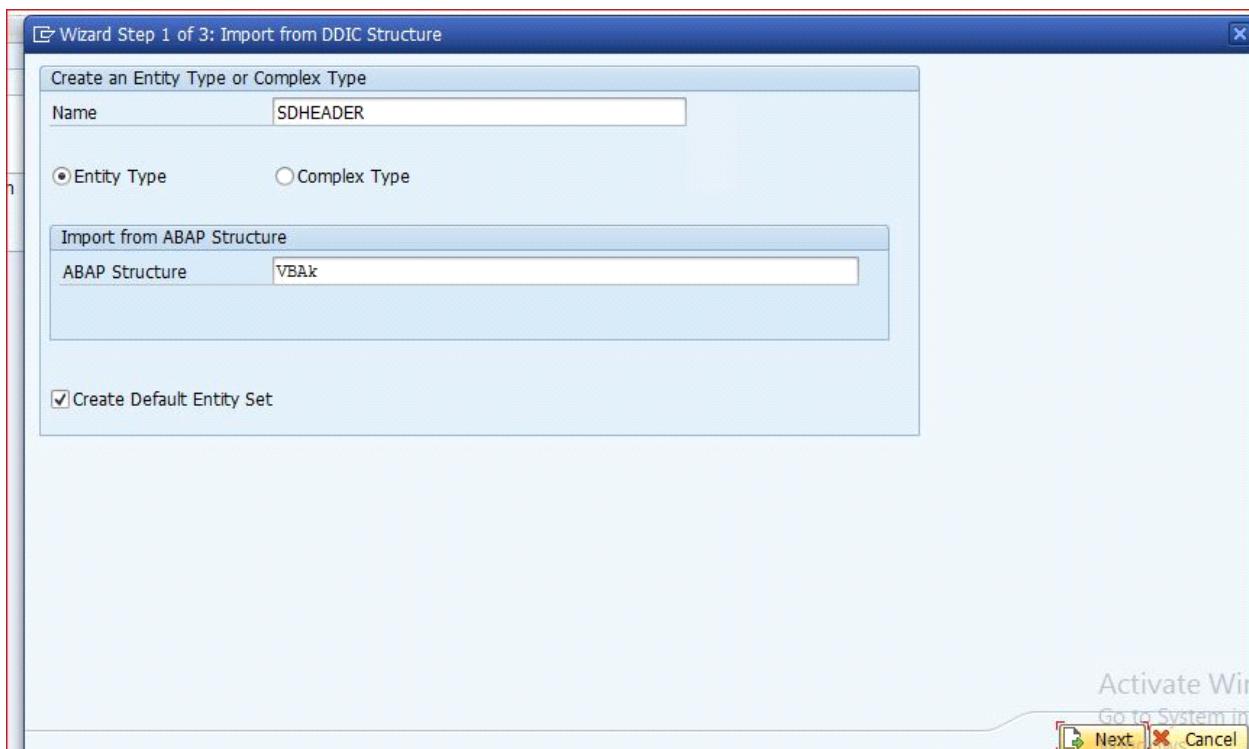


Now we are going to define structure of work area and internal table , Right click on Data model select import and select DDIC structure .



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



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	<input type="checkbox"/>		VBAK	
• MANDT	<input type="checkbox"/>	Client	CLNT	3
• VBELN	<input checked="" type="checkbox"/>	Sales Document	CHAR	10
• ERDAT	<input checked="" type="checkbox"/>	Created on	DATS	8
• ERZET	<input checked="" type="checkbox"/>	Time	TIMS	6
• ERNAM	<input checked="" type="checkbox"/>	Created by	CHAR	12
• ANGDT	<input checked="" type="checkbox"/>	Valid From	DATS	8
• BNDDT	<input type="checkbox"/>	Valid To	DATS	8
• AUDAT	<input checked="" type="checkbox"/>	Document Date	DATS	8
• VBTYP	<input type="checkbox"/>	Document Cat.	CHAR	4
• TRVOG	<input type="checkbox"/>	Transact. Group	CHAR	1
• AUART	<input type="checkbox"/>	Sales Doc. Type	CHAR	4
• AUGRU	<input type="checkbox"/>	Order Reason	CHAR	3
• GWLDT	<input type="checkbox"/>	Begin guarantee	DATS	8
• SUBMI	<input type="checkbox"/>	Collective No.	CHAR	10
• LIFSK	<input type="checkbox"/>	Delivery Block	CHAR	2
• FAKSK	<input type="checkbox"/>	Billing Block	CHAR	2
• NETWR	<input type="checkbox"/>	Net Value	CURR	15
• WAERK	<input type="checkbox"/>	Doc. Currency	CURR	5

Activate Window Go to System Info Back Next Cancel

project5

Other window will appear where we have to 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
<input checked="" type="checkbox"/>	SDHEADER	VBELN	<input checked="" type="checkbox"/>	CHAR	Vbeln	Sales Document
<input checked="" type="checkbox"/>	SDHEADER	ERDAT	<input type="checkbox"/>	DATS	Erdat	Created on
<input checked="" type="checkbox"/>	SDHEADER	ERZET	<input type="checkbox"/>	TIMS	Erzet	Time
<input checked="" type="checkbox"/>	SDHEADER	ERNAM	<input type="checkbox"/>	CHAR	Ernam	Created by
<input checked="" type="checkbox"/>	SDHEADER	ANGDT	<input type="checkbox"/>	DATS	Angdt	Valid From
<input checked="" type="checkbox"/>	SDHEADER	BNDDT	<input type="checkbox"/>	DATS	Bnddt	Valid To
<input checked="" type="checkbox"/>	SDHEADER	AUDAT	<input type="checkbox"/>	DATS	Audat	Document Date

Activate Window Go to System Info Back Finish Cancel

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.

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

Similarly Create Structure for SD Item following same steps

Wizard Step 1 of 3: Import from DDIC Structure

Create an Entity Type or Complex Type

Name: SDITEM

Entity Type: Entity Type Complex Type

Import from 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	<input type="checkbox"/>	VBAP				
MANDT	<input type="checkbox"/>	Client	CLNT	3		
VBELN	<input checked="" type="checkbox"/>	Sales Document	CHAR	10		
POSNR	<input checked="" type="checkbox"/>	Item	NUMC	6		
MATNR	<input checked="" type="checkbox"/>	Material	CHAR	40		
MATWA	<input checked="" type="checkbox"/>	MaterialEntered	CHAR	40		
PMATN	<input checked="" type="checkbox"/>	Pr. Ref. Matl	CHAR	40		
CHARG	<input checked="" type="checkbox"/>	Batch	CHAR	10		
MATKL	<input checked="" type="checkbox"/>	Material Group	CHAR	9		
ARKTX	<input checked="" type="checkbox"/>	Item Descr.	CHAR	40		
PSTYV	<input type="checkbox"/>	Item category	CHAR	4		
POSAR	<input type="checkbox"/>	Item Type	CHAR	1		
LFREL	<input type="checkbox"/>	Item Delf.Div.	CHAR	1		
FKREL	<input type="checkbox"/>	Relev.for Bill.	CHAR	1		
UEPOS	<input type="checkbox"/>	Higher-Lev.Item	NUMC	6		
GRPOS	<input type="checkbox"/>	Altern. to Item	NUMC	6		
ABGRU	<input type="checkbox"/>	RejectionReason	CHAR	2		
PRODH	<input type="checkbox"/>	Prod.hierarchy	CHAR	18		
PRODUNIT	<input type="checkbox"/>	Node ID	CHAR	40		

Wizard Step 3 of 3: Import from DDIC Structure

Modify Entity Type

IEn...	Complex/Entity Type	ABAP Name	Is K...	Type	Name	Label
<input checked="" type="checkbox"/>	SDITEM	VBELN	<input checked="" type="checkbox"/>	CHAR	Vbeln	Sales Document
<input checked="" type="checkbox"/>	SDITEM	POSNR	<input checked="" type="checkbox"/>	NUMC	Posnr	Item
<input checked="" type="checkbox"/>	SDITEM	MATNR	<input type="checkbox"/>	CHAR	Matnr	Material
<input checked="" type="checkbox"/>	SDITEM	MATWA	<input type="checkbox"/>	CHAR	Matwa	MaterialEntered
<input checked="" type="checkbox"/>	SDITEM	PMATN	<input type="checkbox"/>	CHAR	Pmatn	Pr. Ref. Matl
<input checked="" type="checkbox"/>	SDITEM	CHARG	<input type="checkbox"/>	CHAR	Charg	Batch
<input checked="" type="checkbox"/>	SDITEM	MATKL	<input type="checkbox"/>	CHAR	Matkl	Material Group
<input checked="" type="checkbox"/>	SDITEM	ARKTX	<input type="checkbox"/>	CHAR	Arktx	Item Descr.

Activate Window

Go to System Information

Back Finish Cancel

Properties

Name	Key	Edm Type	Prec.	Scale	Max.	Unit Prop.	Cre...	Upd...	Sort...	Null...	Filt...	Label	L...	Comp. Type	ABAP Field	A...	Semantics
Vbeln	<input checked="" type="checkbox"/>	Edm.String	0	0	10		<input checked="" type="checkbox"/>	Sales Docu...	T	VBELN	<input type="checkbox"/>						
Posnr	<input checked="" type="checkbox"/>	Edm.String	0	0	6		<input checked="" type="checkbox"/>	Item	T	POSNR	<input type="checkbox"/>						
Matnr	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	Material	T	MATNR	<input type="checkbox"/>						
Matwa	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	MaterialEnt...	T	MATWA	<input type="checkbox"/>						
Pmatn	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	Pr. Ref. Matl	T	PMATN	<input type="checkbox"/>						
Charg	<input type="checkbox"/>	Edm.String	0	0	10		<input checked="" type="checkbox"/>	Batch	T	CHARG	<input type="checkbox"/>						
Matkl	<input type="checkbox"/>	Edm.String	0	0	9		<input checked="" type="checkbox"/>	Material Gro...	T	MATKL	<input type="checkbox"/>						
Arktx	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	Item Descr.	T	ARKTX	<input type="checkbox"/>						

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 details and proceed further . These are also knowns as runtime artifacts.

Model and Service Definition

Model Provider Class	
Class Name	ZCL_ZTEST_PI_SD_MPC_EXT
Base Class Name	ZCL_ZTEST_PI_SD_MPC
Data Provider Class	
<input checked="" type="checkbox"/> 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

Step 2. Go to transaction code /IWFND/MAINT_SERVICE .

Click on push button Add services .

Activate and Maintain Services

Service Catalog								
Type	Technical Service Name	V...	Service Description	External Service Name	Nsp.	OAuth	Soft State	Processing Mode
BEP	ZABAP REPOSITORY SRV	1	SAPUI5 ABAP Repository services via oData	ABAP REPOSITORY SRV	/UI5/	<input type="checkbox"/>	Not Supported	Routing-based
BEP	ADT	1	Gateway service for ADT	ADT SRV		<input checked="" type="checkbox"/>	Not Supported	Routing-based
BEP	ZAPJ JOB MANAGEMENT SRV	1	Application job management	APJ JOB MANAGEMENT SRV		<input checked="" type="checkbox"/>		Co-deployed only
BEP	ZAPL LOG MANAGEMENT SRV	1	Application Log management	APL LOG MANAGEMENT SRV		<input checked="" type="checkbox"/>		Co-deployed only
BEP	ZAPS CDS GKE SRV	1	Generic CDS key user access	APS CDS GKE SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZAPS CHANGE DOCUMENTS SRV	1	OData Service for Change Documents	APS CHANGE DOCUMENTS SRV		<input checked="" type="checkbox"/>		Co-deployed only
BEP	ZAPS EXT CBO D SRV	1	Custom Business Objects: Draft	APS EXT CBO D SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZAPS EXT CCL SRV	1	Custom Code List	APS EXT CCL SRV		<input type="checkbox"/>	Not Supported	Routing-based
BEP	ZAPS EXT TRC SRV	1	OData Services for Tracing	APS EXT TRC SRV		<input type="checkbox"/>	Not Supported	Routing-based

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

Add Selected Services

Get Services

Filter

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

Add Selected Services

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	

Activate Windows
Go to System in Control
Windows

Now double click on your project name . 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	\$IMP
	Local Object
ICF Node	
<input checked="" type="radio"/> Standard Mode	<input type="radio"/> None
<input checked="" type="checkbox"/> Set Current Client as Default Client in ICF Node	
OAuth enablement	
<input type="checkbox"/> 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

The screenshot shows the SAP Gateway Client interface. At the top, there is a table listing various technical services with their descriptions, external service names, namespaces, OAuth status, soft state, and processing mode. Some entries are highlighted in yellow. Below this is a toolbar with icons for ICF Node, Call Browser, and SAP Gateway Client. A sub-toolbar for 'ICF Nodes' shows one node named 'ODATA' with a status of 'Standard Mode'. To the right, there is a section for 'System Aliases' with a single entry for 'LOCAL' as a 'Local System Alias'. A red box highlights the 'SAP Gateway Client' tab in the toolbar.

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

The screenshot shows the SAP Gateway Client interface with an executed transaction. The 'Request URI' is set to `/sap/opu/odata/sap/ZTEST_PT_SD_SRV/?$format=xml`. The 'HTTP Response' panel shows a successful response with status code 200 and reason OK. The XML content of the response is displayed in the bottom right pane:

```

<?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_PT_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:createable="false">
          <atom:title type="text">SDHEADERSet</atom:title>
        
```

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.

The screenshot shows the SAP ABAP Workbench interface. In the top bar, 'Class/Interface' is set to 'ZCL_ZTEST_PI_SD_DPC_EXT'. Below the toolbar, the object name is 'Object Name' and the description is 'Description'. The 'Methods' section is expanded, showing 'Inherited Methods' which include several methods from the '/IWBEPE/IF_SB_DPC_COMM_SRV' interface. One method, 'SDHEADERSET_GET_ENTITYYS', is highlighted with a yellow box. A context menu is open over this method, with the 'Redefine' option highlighted in yellow. Other options in the menu include 'Change', 'Display', 'Consistency Check', 'Activate', 'Copy...', 'Where-Used List', 'Version Management', and 'Additional Functions'. To the right of the menu, there are two columns of related objects: 'Related EntitySet Name: SDHEA' and 'Related EntitySet Name: SDITEM'.

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

Builder Class ZCL_ZTEST_PI_SD_DPC_EXT Change

```

Type Parameter          Typing
IT_ORDER                TYPE /IWBEPI/T_MGW_SORTING_ORDER
IV_FILTER_STRING         TYPE STRING
IV_SEARCH_STRING         TYPE STRING
IO_TECH_REQUEST_CONTEXT TYPE REF TO /IWBEPI/IF_MGW_REQ_ENTITYSET OPTIONAL
ET_ENTITYSET              TYPE ZCL_ZTEST_PI_SD_MPC->TT_SDHEADER
ES_RESPONSE_CONTEXT      TYPE /IWBEPI/IF_MGW_APPL_SRV_RUNTIME->TY_S_MGW_RESPONSE_C
/IWBEPI/CX_MGW_BUSI_EXCEPTION
/IWBEPI/CX_MGW_TECH_EXCEPTION

Method SDHEADERSET_GET_ENTITYSET active
1   method SDHEADERSET_GET_ENTITYSET.
2
3     Select * from vbak into CORRESPONDING FIELDS OF table ET_ENTITYSET
4       UP TO 5 rows.
5
6   **TRY.

```

Similarly , Write code for GetEntitySet for Item and activate .

```

Type Parameter          Typing
IT_ORDER                TYPE /IWBEPI/T_MGW_SORTING_ORDER
IV_FILTER_STRING         TYPE STRING
IV_SEARCH_STRING         TYPE STRING
IO_TECH_REQUEST_CONTEXT TYPE REF TO /IWBEPI/IF_MGW_REQ_ENTITYSET OPTIONAL
ET_ENTITYSET              TYPE ZCL_ZTEST_PI_SD_MPC->TT_SDITEM
ES_RESPONSE_CONTEXT      TYPE /IWBEPI/IF_MGW_APPL_SRV_RUNTIME->TY_S_MGW_RESPONSE_C
/IWBEPI/CX_MGW_BUSI_EXCEPTION
/IWBEPI/CX_MGW_TECH_EXCEPTION

Method SDITEMSET_GET_ENTITYSET active
1   method SDITEMSET_GET_ENTITYSET.
2     Select * from VBAP into CORRESPONDING FIELDS OF TABLE ET_ENTITYSET
3       UP TO 10 rows.
4
5   **TRY.
6   *CALL METHOD SUPER->SDITEMSET_GET_ENTITYSET
7   + EXPORTING
8     IV_ENTITY_NAME      =
9     IV_ENTITY_SET_NAME  =
10    IV_SOURCE_NAME      =
11    IT_FILTER_SELECT_OPTIONS =
12    IS_PAGING            =

```

Step -4

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

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 .

The screenshot shows the SAP Gateway Client interface. At the top, there's a table titled 'EntitySets' with several entries. Below it are three tabs: 'ICF Nodes', 'System Aliases', and others. The 'ICF Nodes' tab shows a single entry for 'ODATA'. The 'System Aliases' tab shows a single entry for 'LOCAL'.

BEP	ZTEST ODATA_SRV	1 My first Odata project	ZTEST PI_SD_SRV	Not Supported	Routing
BEP	ZTEST PIN PO_SRV	1 test PO Odata	ZTEST PIN PO_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTEST PIN SD_SRV	1 Test for SD documents	ZTEST PIN SD_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTEST SERVICE GENERATION_SRV	1 Odata for Service Generatio technique	ZTEST SERVICE GENERATION_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTEST_SRV	1 test	ZTEST_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTEST TABLE1_SRV	1 test	ZTEST TABLE1_SRV	<input type="checkbox"/>	Not Supported Routing
BEP	ZTRANSLATIONS_SRV	1 cds view translations	ZTRANSLATIONS_SRV	<input type="checkbox"/>	Not Supported Co-dep
BEP	ZX SO CDS	1 Sales Order	ZX SO CDS	<input type="checkbox"/>	Not Supported Routing

Now Click on Entity set.

The screenshot shows the SAP Gateway Client interface. The 'EntitySets' tab is selected, displaying a table with two entries: 'SDHEADERSet' and 'SDITEMSet'. A modal dialog box titled 'EntitySets (1) 2 Entries found' is open, showing the same two entries. The 'HTTP Request' and 'HTTP Response' panes are visible at the bottom.

Select entity for header and enter

The screenshot shows the SAP Gateway Client interface. The 'EntitySets' tab is selected, displaying a table with two entries: 'SDHEADERSet' and 'SDITEMSet'. A modal dialog box titled 'EntitySets (1) 2 Entries found' is open, showing the same two entries. The 'SDHEADERSet' entry is highlighted. The 'HTTP Request' and 'HTTP Response' panes are visible at the bottom.

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

HTTP Method: GET

Request URI: /sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDHEADERSet

Protocol: HTTP

Test Group: Test Case

HTTP Response - Processing Time = 876 ms

Header Name Value

~status_code	200
~status_reason	OK

```

<category schema="http://schemas.microsoft.com/ado/2007/08/dataservices" type="Collection(SDHEADERSet)" title="SDHEADERSet">
  <link href="SDHEADERSet('1')/SDHEADERSet('1')>
  <content type="application/xml">
    <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
      <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:Bndt m:null="true"/>
      <d:Audat>2020-01-14T00:00:00</d:Audat>
    </m:properties>
  </content>
</category>

```

Activate Windows
Go to System in Control Panel to activate

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

HTTP Method: GET

Request URI: /sap/opu/odata/sap/ZTEST_PI_SD_SRV/SDITEMSet

Protocol: HTTP

Test Group: Test Case

HTTP Response - Processing Time = 1167 ms

Header Name Value

~status_code	200
~status_reason	OK

```

<category schema="http://schemas.microsoft.com/ado/2007/08/dataservices" type="Collection(ITEMSet)" title="ITEMSet">
  <link href="ITEMSet('1')/ITEMSet('1')>
  <content type="application/xml">
    <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
      <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>
</category>

```

Here , We had completed our Odata service creation with a Service 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.

Thanks