

Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

www.ZAPYard.com

ZAPYARD

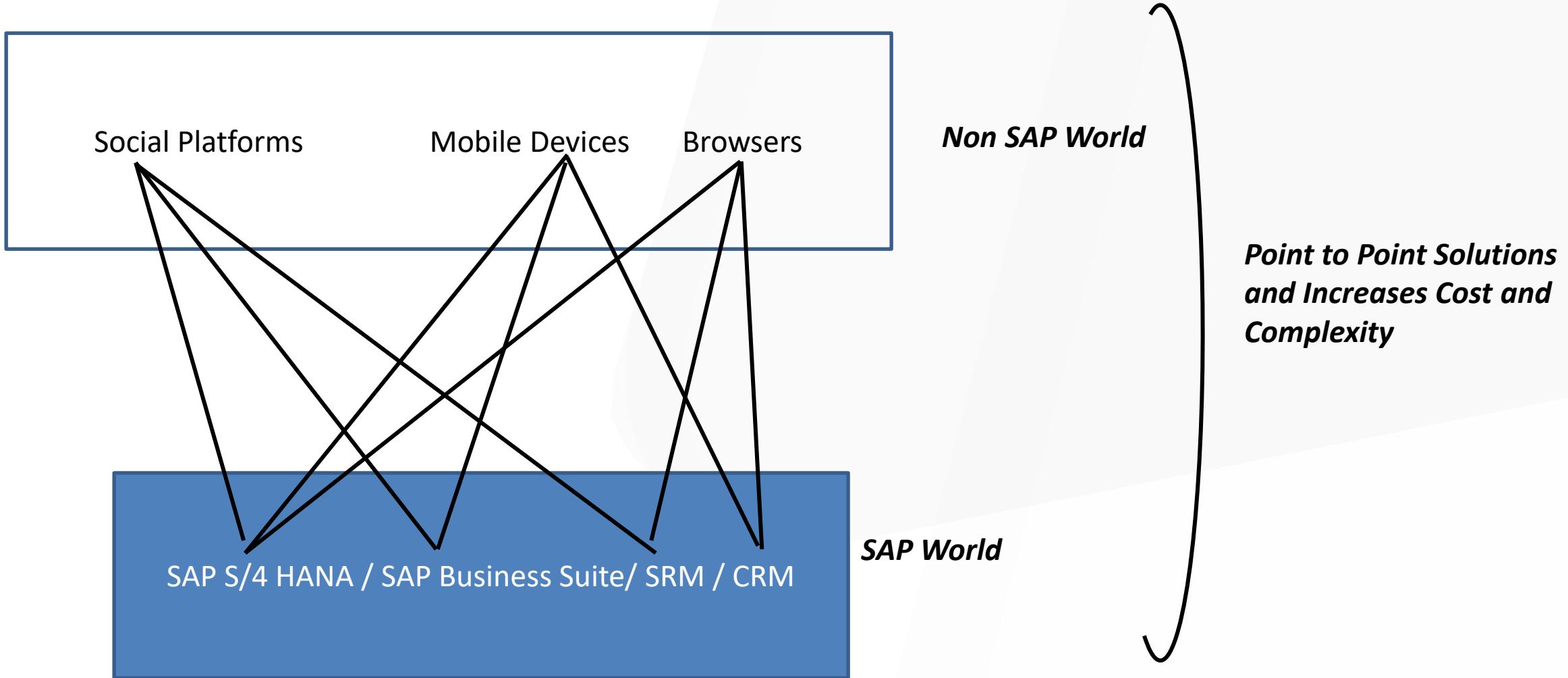
Agenda

- Understanding the SAP Gateway and the OData Protocol
- Understanding the SAP Gateway Deployment Options
- Basic Understanding of OData and EDM Model and different ways to create an OData Service
- Explore OData.org Website

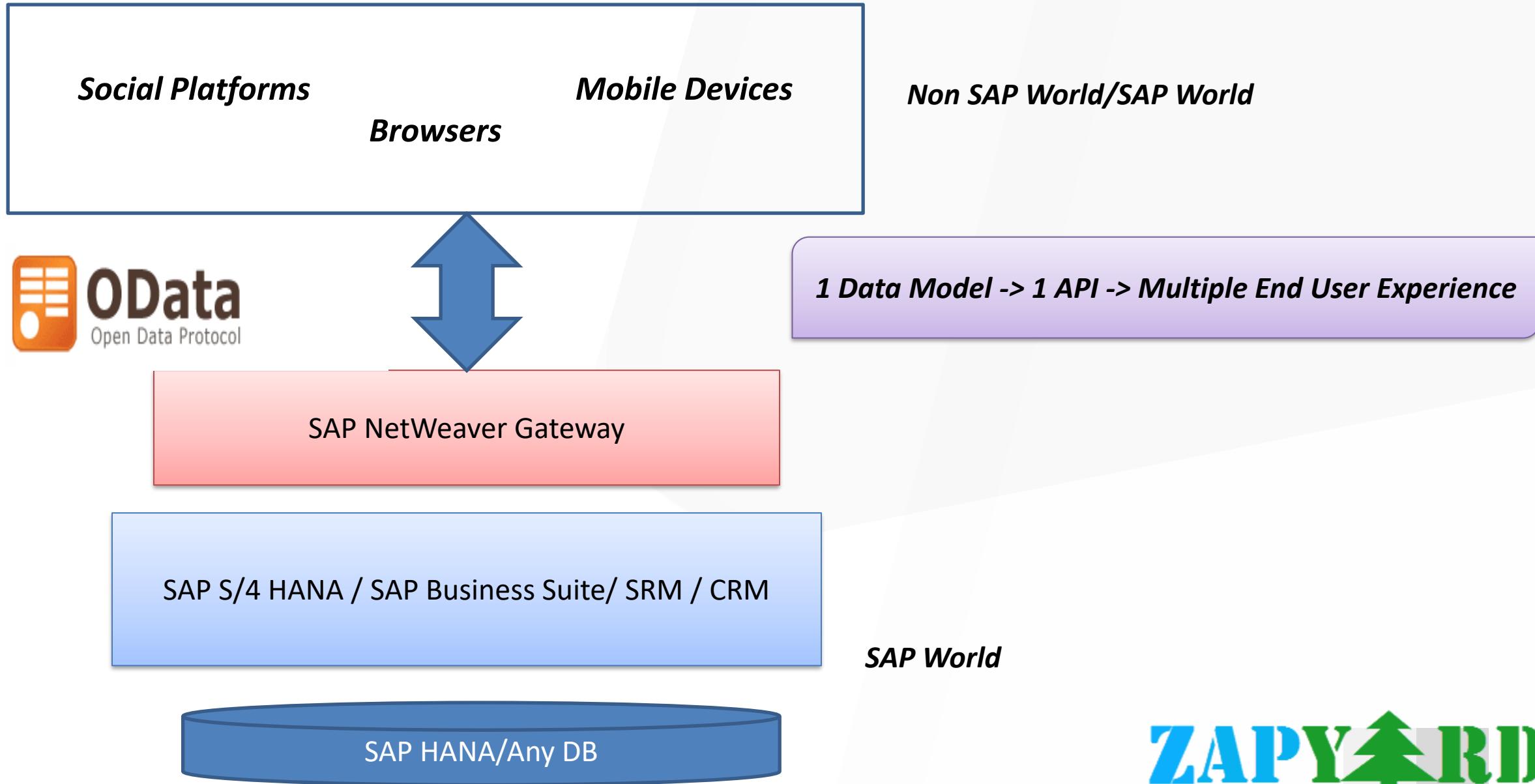


OData - the best way to REST

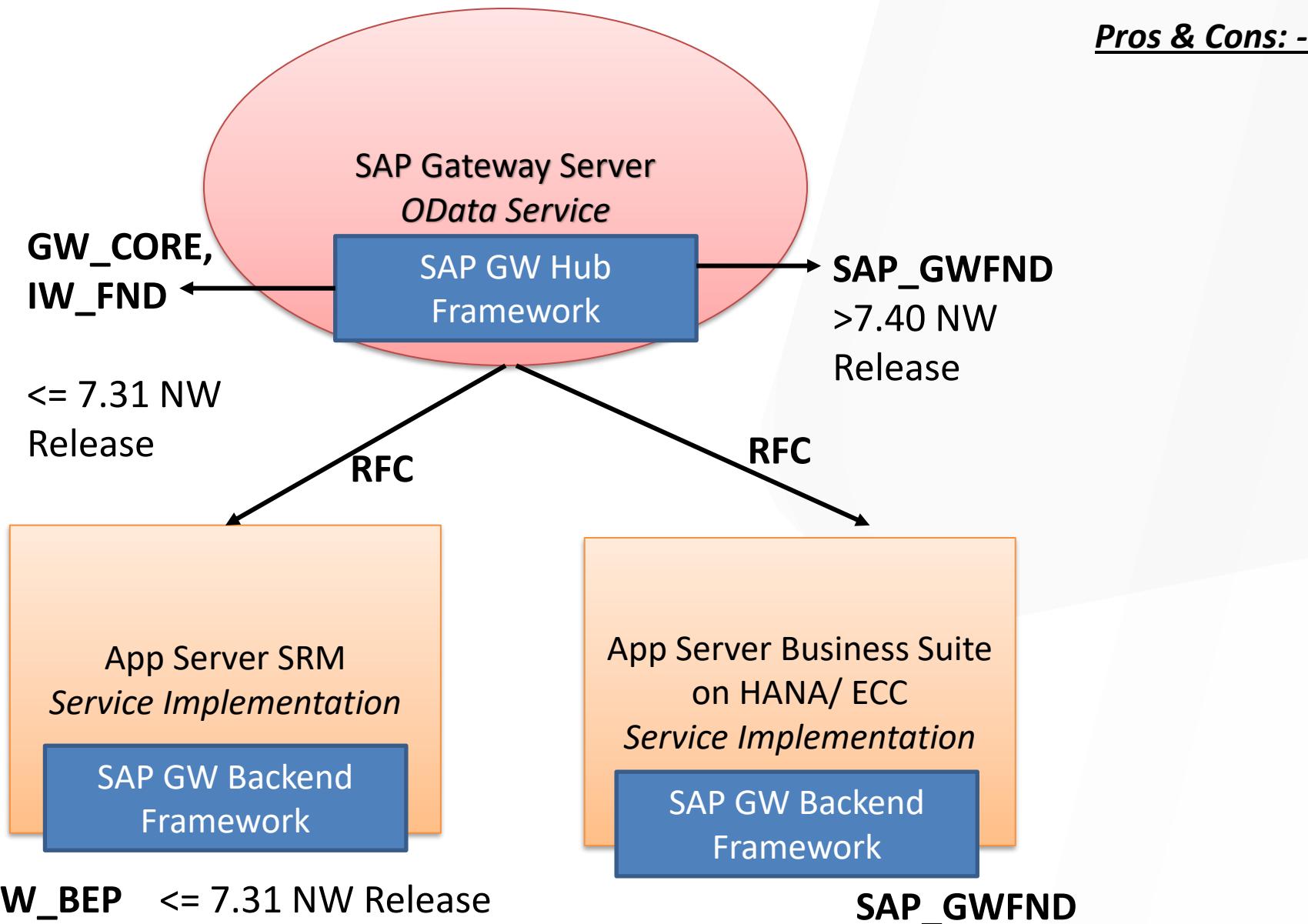
Understanding the SAP Gateway and the OData Protocol



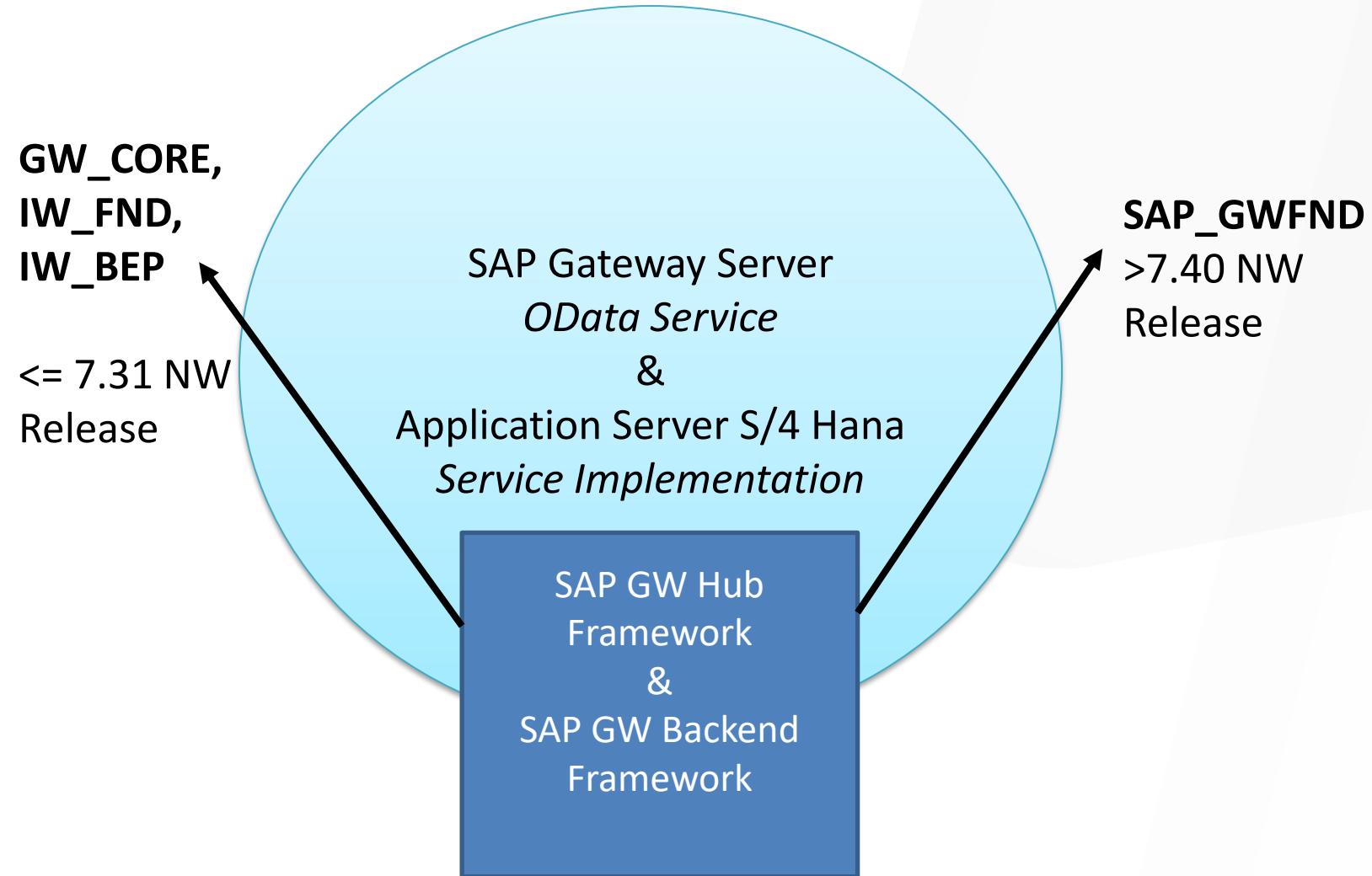
What's the solution to the earlier Problem:- OData



SAP Gateway Deployment Options:- Central Hub Deployment

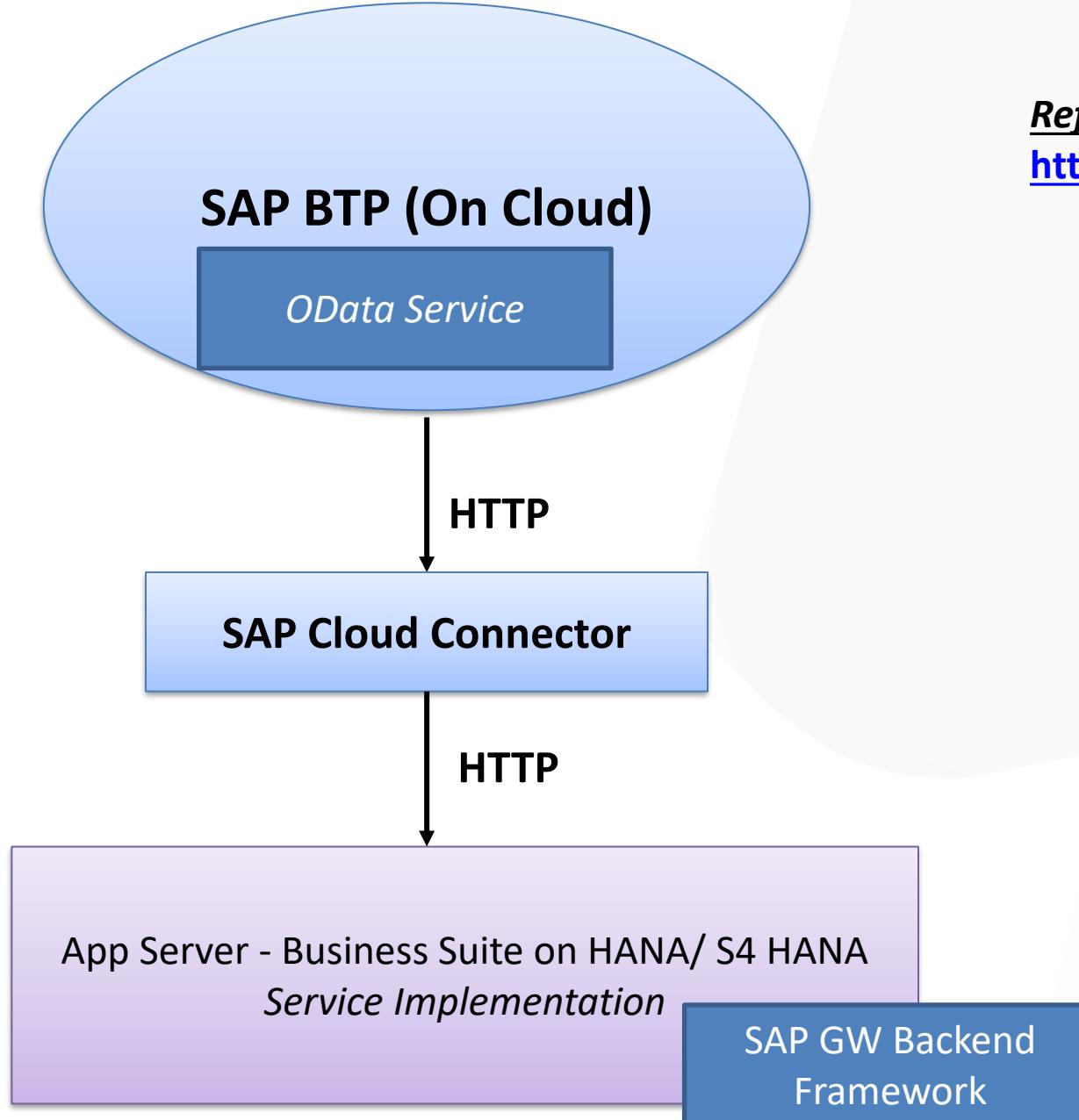


SAP Gateway Deployment Options:- Embedded Deployment



Pros & Cons: -

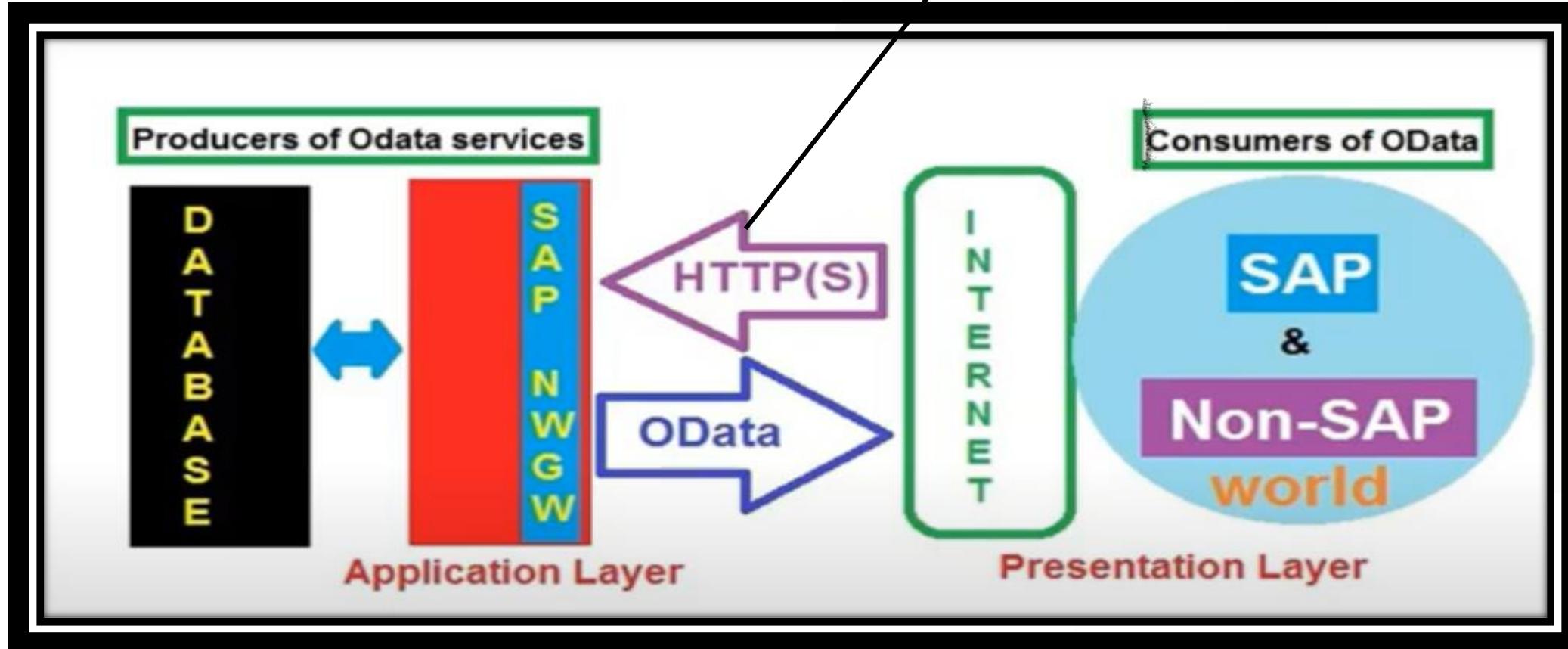
SAP Gateway Deployment Options:- OData Provisioning via SAP BTP



Reference Link w.r.t WebIDE Setup:-

<https://sap.github.io/cloud-s4ext/week-3/unit-2/>

OData Overview and Understanding REST



OData is a REST-inspired Technology for reading, writing and modifying information.

→ Representational State Transfer

OData Service Structure:- *Each OData Service is represented by a URI*

There are 2 types of document associated with each OData Service:-

- ❖ The service document
- ❖ The service Metadata Document

Information of the Data about the Data

```
{  
  "d": {  
    "EntitySets": [  
      "BusinessPartnerSet",  
      "ProductSet",  
      "SalesOrderSet",  
      "SalesOrderLineItemSet",  
      "ContactSet",  
      "VH_SexSet",  
      "VH_CountrySet",  
      "VH_AddressTypeSet",  
      "VH_CategorySet",  
      "VH_CurrencySet",  
      "VH_UnitQuantitySet",  
      "VH_UnitWeightSet",  
      "VH_UnitLengthSet",  
      "VH_ProductTypeCodeSet",  
      "VH_BPRoleSet",  
      "VH_LanguageSet"  
    ]  
  }  
}
```

[http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/ + \\$metadata](http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/ + $metadata)

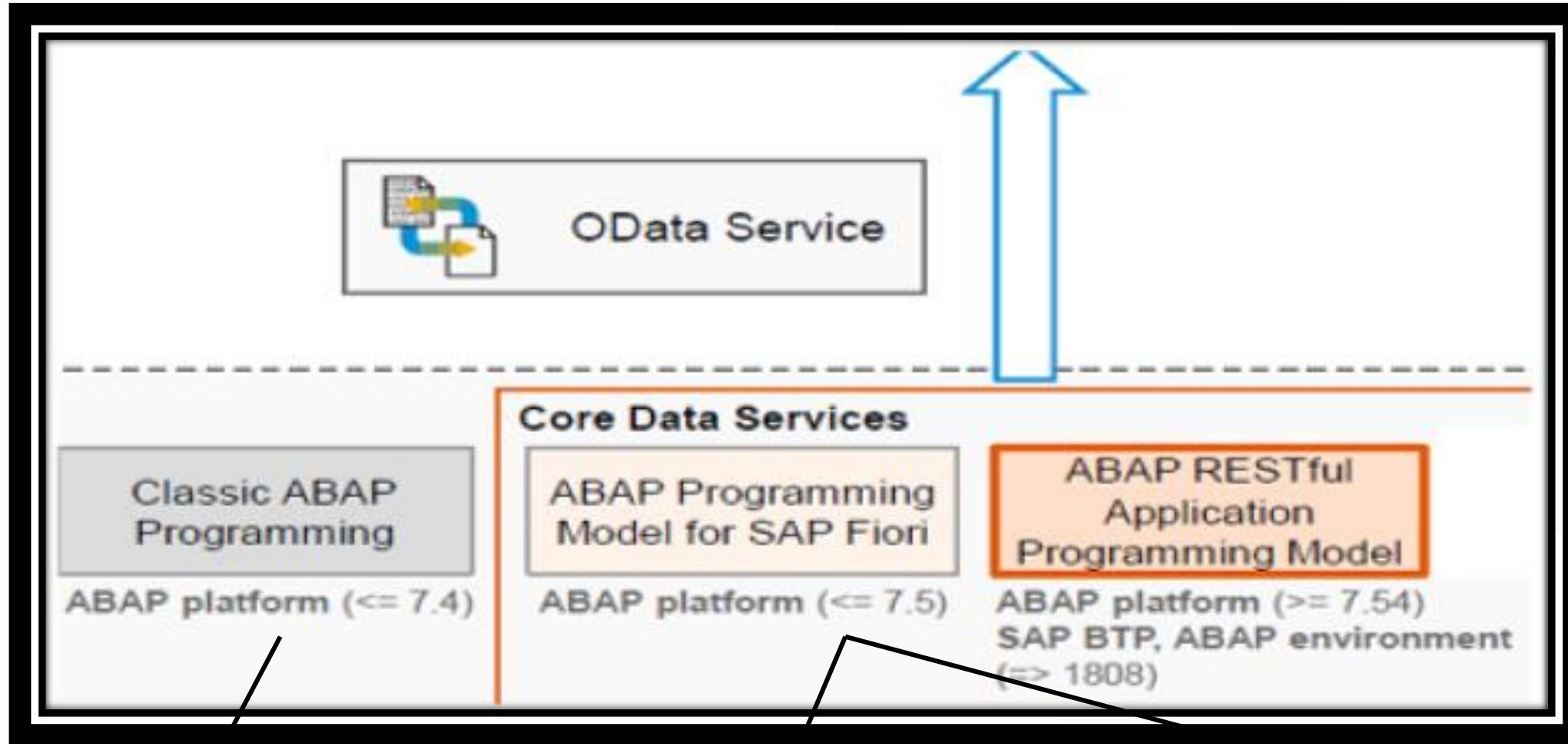
[http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/\\$metadata](http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/$metadata)

Service MetaData and the Entity Data Model:-

The diagram illustrates the Service MetaData XML structure for the 'BusinessPartner' entity type. A large blue arrow on the left points from the XML code towards the annotations. A black arrow points from the text 'Key Property' to the opening tag of the 'Key' element. A curly brace on the right groups the 'Key' element and the 'Property' and 'NavigationProperty' elements under the heading 'Normal Property and Navigation Property'. A small watermark 'Snipping Tool' is visible in the center of the XML code area.

```
<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="GWSAMPLE_BASIC" xml:lang="en" sap:schema-version="1">
      <EntityType Name="BusinessPartner" sap:content-version="1">
        <Key>
          <PropertyRef Name="BusinessPartnerID"/>
        </Key>
        <Property Name="Address" Type="GWSAMPLE_BASIC.CT_Address" Nullable="false"/>
        <Property Name="BusinessPartnerID" Type="Edm.String" Nullable="false" MaxLength="10" sap:label="Business Partner ID" sap:creatable="false" sap:updatable="false"/>
        <Property Name="CompanyName" Type="Edm.String" Nullable="false" MaxLength="80" sap:label="Company"/>
        <Property Name="WebAddress" Type="Edm.String" sap:label="Web Address" sap:sortable="false" sap:filterable="false" sap:semantics="url"/>
        <Property Name="EmailAddress" Type="Edm.String" Nullable="false" MaxLength="255" sap:label="Email" sap:semantics="email"/>
        <Property Name="PhoneNumber" Type="Edm.String" MaxLength="30" sap:label="Phone" sap:semantics="tel"/>
        <Property Name="FaxNumber" Type="Edm.String" MaxLength="30" sap:label="Fax Number"/>
        <Property Name="LegalForm" Type="Edm.String" MaxLength="10" sap:label="Legal Form"/>
        <Property Name="CurrencyCode" Type="Edm.String" Nullable="false" MaxLength="5" sap:label="Currency Code" sap:semantics="currency-code"/>
        <Property Name="BusinessPartnerRole" Type="Edm.String" Nullable="false" MaxLength="3" sap:label="Bus. Part. Role"/>
        <Property Name="CreatedAt" Type="Edm.DateTime" Precision="7" sap:label="Time Stamp" sap:creatable="false" sap:updatable="false"/>
        <Property Name="ChangedAt" Type="Edm.DateTime" Precision="7" ConcurrencyMode="Fixed" sap:label="Time Stamp" sap:creatable="false" sap:updatable="false"/>
        <NavigationProperty Name="ToSalesOrders" Relationship="GWSAMPLE_BASIC.Assoc_BusinessPartner_SalesOrders" FromRole="FromRole_Assoc_BusinessPartner_SalesOrders"
          ToRole="ToRole_Assoc_BusinessPartner_SalesOrders"/>
        <NavigationProperty Name="ToContacts" Relationship="GWSAMPLE_BASIC.Assoc_BusinessPartner_Contacts" FromRole="FromRole_Assoc_BusinessPartner_Contacts"
          ToRole="ToRole_Assoc_BusinessPartner_Contacts"/>
        <NavigationProperty Name="ToProducts" Relationship="GWSAMPLE_BASIC.Assoc_BusinessPartner_Products" FromRole="FromRole_Assoc_BusinessPartner_Products"
          ToRole="ToRole_Assoc_BusinessPartner_Products"/>
      </EntityType>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

Holistic Picture of different ways of Creating and Generating OData Services:-



Importing the ABAP CDS View as a Data Source Reference / CDS Based Annotations

SEGW(Service Gateway Builder Tool)

ZAPYARD

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



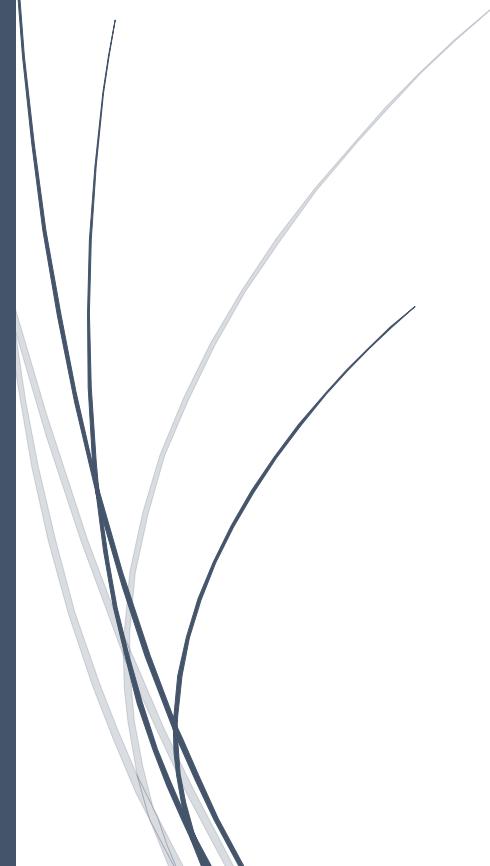
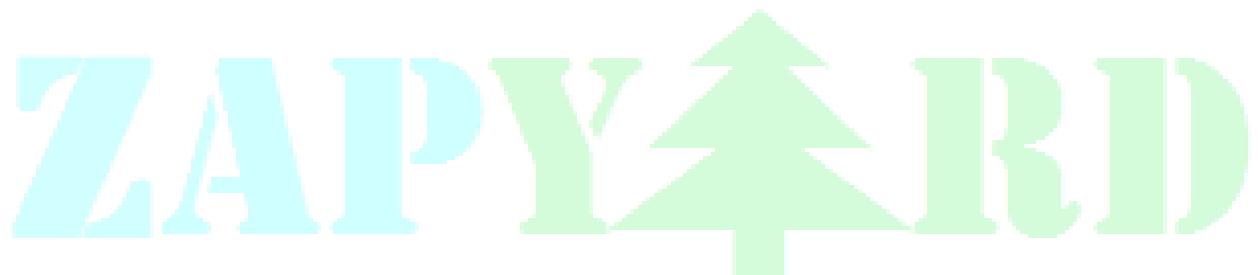
mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



Master SAP OData

Day 1

Issues with Not having SAP Gateway and OData Protocol: -

Poor Scalability, Increased System landscape and increased administration effort

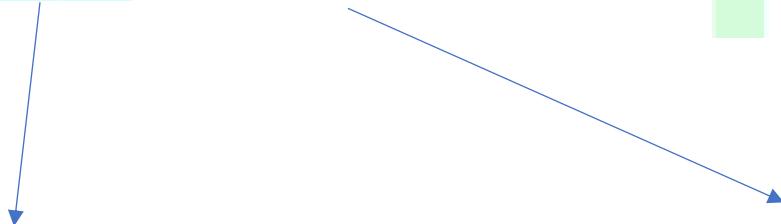
SAP Gateway is more of an infrastructure that allows you to easily develop or generate your own OData services. It's like a window for outside world to peep into SAP and transfer data to/from SAP

OData -> Open Data Protocol based on HTTP

OData basically helps me to develop services with high level of data integration and cross-platform interoperability.

3 different Deployment Options:-

Central Hub/Embedded/Odata Provisioning



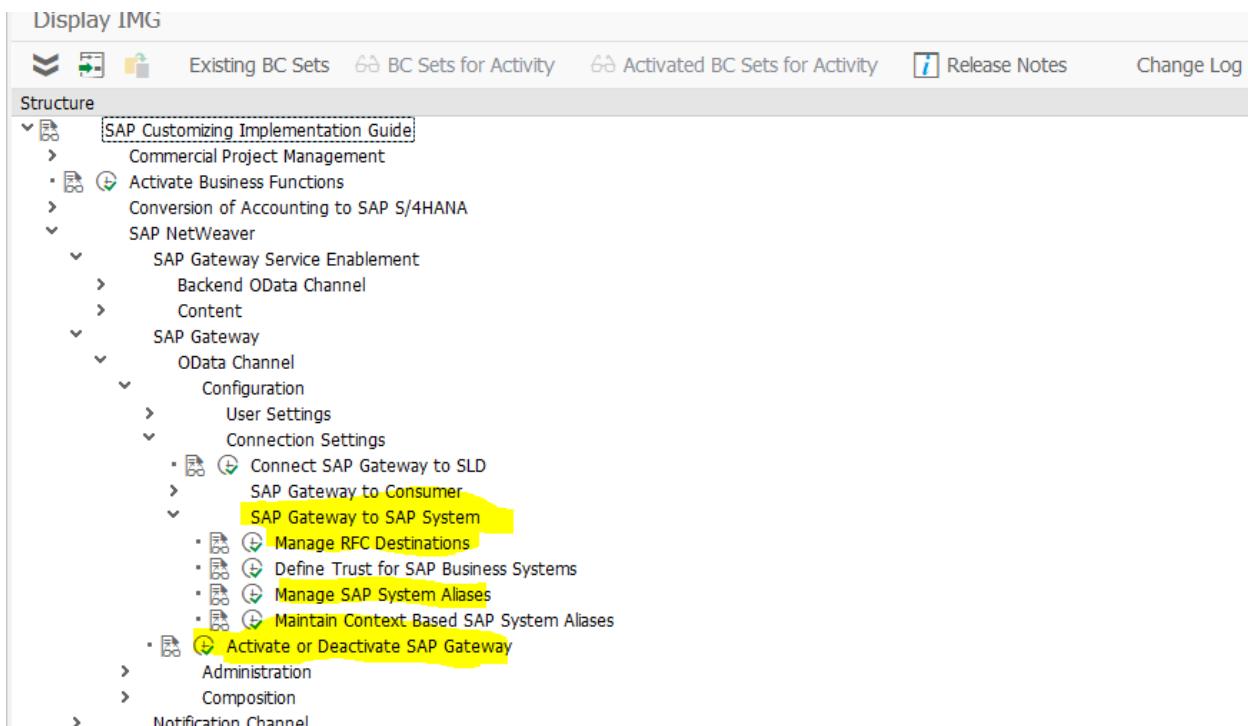
Advantages: -Innovation w/o Disruption, Multiple SAP System and Load Balancing
Disadvantages: - Cost, Security, Slow, Double Maintenance and Confusion

Advantages: - Low Cost, No confusion, No double Maintenance and Good performance
Disadvantages: - Scattered Landscape and Everything will be down

Master SAP OData

The response format of an Odata Service includes both JSON/XML based formats.

SPRO Configuration: -



Design Philosophy of a Odata service is based on REST.

Client-Server/Statelessness/Uniform Interface/Code-On-Demand/Cacheability

URL we also call as URI(Uniform Resource Identifier).

T-codes:-

/n/iwfnd/maint_services

/O/IWFND/GW_CLIENT

Service Document of an OData Service

http://six30.mydomain.com:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/



http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/

JSON Viewer Chrome Extension

Service Metadata Document

[http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/\\$metadata](http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/$metadata)

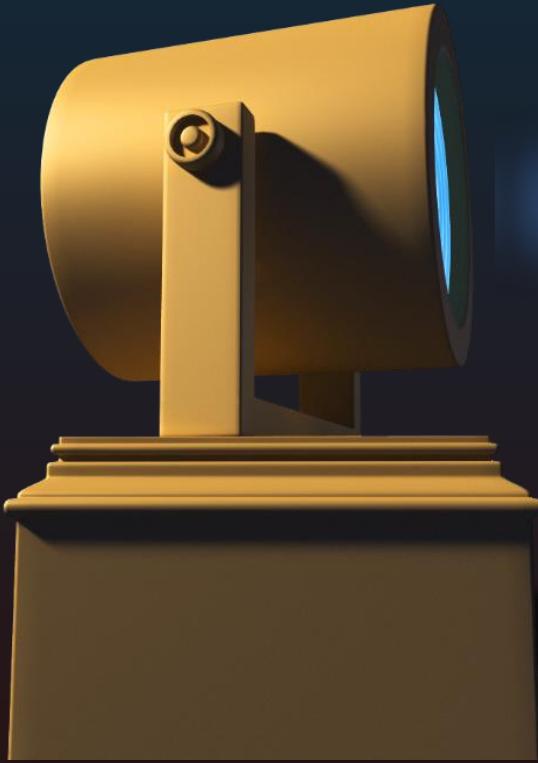


To explore an Entity Set

[http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/BusinessPartnerSet?\\$format=json](http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/BusinessPartnerSet?$format=json)

If we want to have a HTML Page look and all relative links to be enabled:-

http://103.44.1.51:8770/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/BusinessPartnerSet?sap-ds-debug=true



Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

www.ZAPYard.com

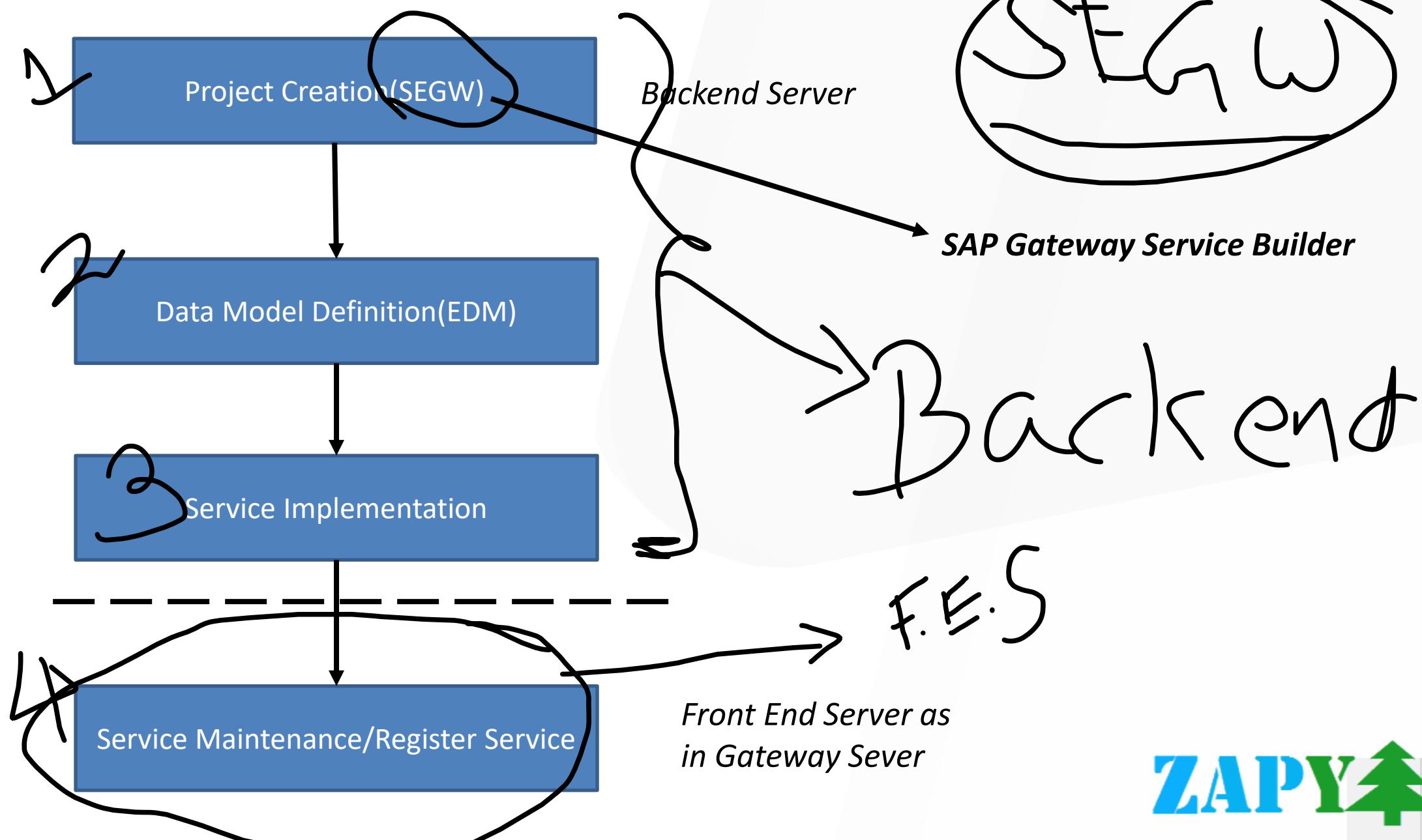
ZAPYARD

Agenda

- Creating first Gateway Service
- Understanding different CRUDQ Operations
- Understanding Service Components and Registering the Service
- Implementing GET Call and its relevant logic
- Understanding the Importance of Key fields in GET Call
- Raising Exceptions as a part of GET Calls



Creating and Implementing our First Gateway Project – Different Phases



Phase 1 and 2 our First Gateway Project – Project Creation and EDM Setup

The screenshot shows the SAP Gateway Service Builder interface. On the left, the project tree displays several projects and services, with **ZLEARN_ODATA** selected. The central pane shows the service details for **odata**, which is described as "Employee Details with Batch Operations". The right pane shows the **Properties** table for the **Entity Types** of the **Header** entity set. The table includes columns for Name, Is Key, Edm Core Type, Prec., Scale, and Max The properties listed are Id (Is Key, Edm.String), Name (Edm.String), Dept (Edm.String), and CreatedOn (Edm.DateTime). A black arrow points from the text in the blue box below to the **CreatedOn** row in the table.

Name	Is Key	Edm Core Type	Prec.	Scale	Max ...
Id	<input checked="" type="checkbox"/>	Edm.String	0	0	3
Name	<input type="checkbox"/>	Edm.String	0	0	10
Dept	<input type="checkbox"/>	Edm.String	0	0	10
CreatedOn	<input type="checkbox"/>	Edm.DateTime	8	0	0

ZLEARN_ODATA

- » Data Model
- » Service Implementation
- » Runtime Artifacts
- » Service Maintenance

project odata

Employee Information testing

Properties

Name	Is Key	Edm Core Type	Prec.	Scale	Max ...
Id	<input checked="" type="checkbox"/>	Edm.String	0	0	3
Name	<input type="checkbox"/>	Edm.String	0	0	10
Dept	<input type="checkbox"/>	Edm.String	0	0	10
CreatedOn	<input type="checkbox"/>	Edm.DateTime	8	0	0

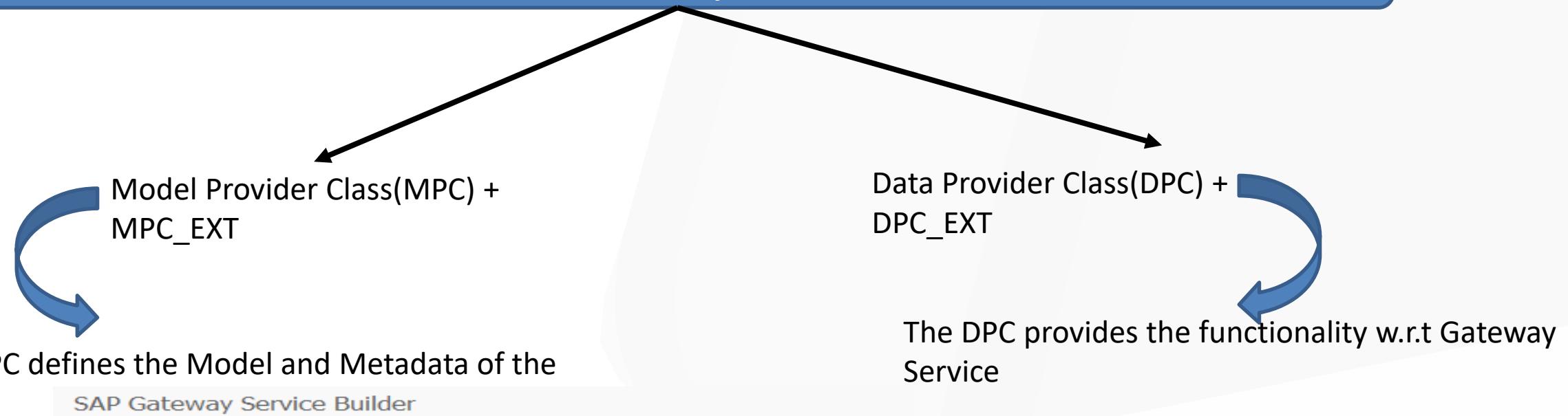
ZLEARN_ODATA

- » Data Model
- » Entity Types
 - » Header
 - » Properties
 - Id
 - Name
 - Dept
 - CreatedOn
 - » Navigation Properties
 - Associations
 - Entity Sets
 - » Service Implementation
 - » Runtime Artifacts
 - » Service Maintenance

While building the OData Model in the Service we can reuse existing data structures or manually create Model Components

Phase 3 of First Gateway Project – Service Implementation

OData service is based on 2 regular ABAP Classes + 2 EXT Classes which gets generated on Generate RunTime Objects :-



The MPC defines the Model and Metadata of the Service

SAP Gateway Service Builder

The screenshot shows the SAP Gateway Service Builder interface. On the left, a tree view lists various services: ZDEMO_ODATA_363, ZEMPLOYEE_BATCH, ZG08_EMP_INFO, ZGW_363_DEMO, ZGW_BILL_INFO, ZGW_PO, ZGW_PO_DEMO, ZGW_PO032, ZGW_PO1, and ZLEARN_ODATA. The ZLEARN_ODATA node is expanded, showing its sub-components: Data Model, Service Implementation, Runtime Artifacts, and Service Maintenance. On the right, detailed information for the ZLEARN_ODATA service is provided, including its name, description ("Employee Details with Batch Operations"), and a note about being a demo of odata.

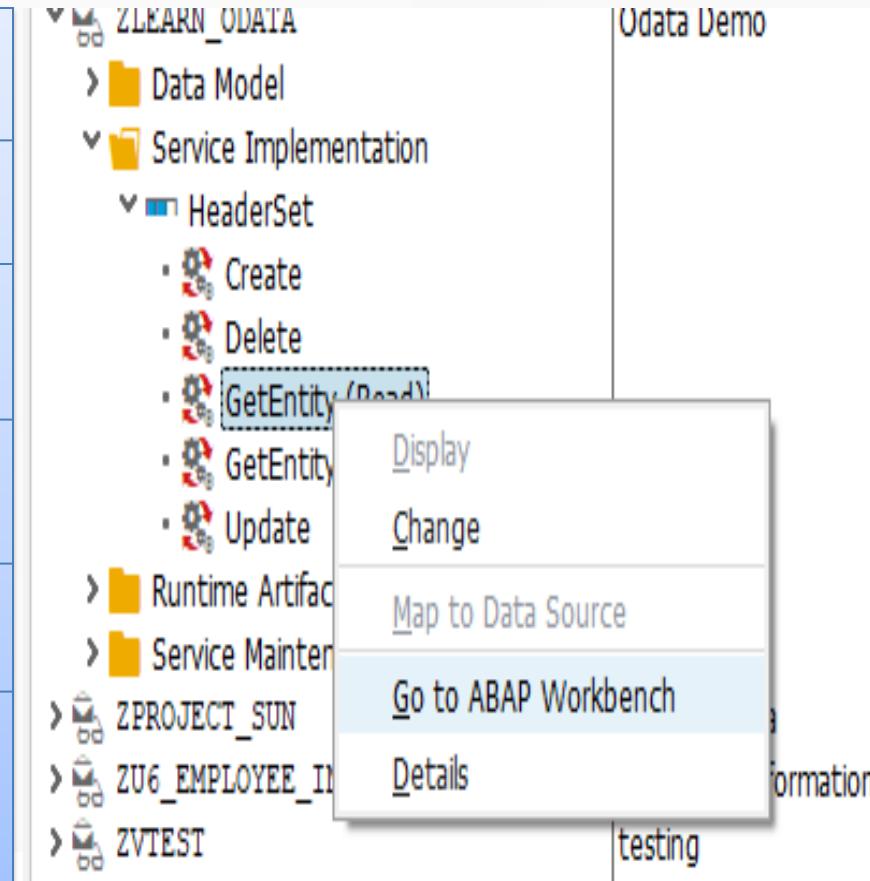
Service	Description
ZDEMO_ODATA_363	Employee Details with Batch Operations
ZEMPLOYEE_BATCH	Employee Information
ZG08_EMP_INFO	demo of odata
ZGW_363_DEMO	Billing info service
ZGW_BILL_INFO	Service for PO
ZGW_PO	dome of service for PO
ZGW_PO_DEMO	Odata Demo
ZGW_PO032	odata
ZGW_PO1	Odata Demo
ZLEARN_ODATA	<ul style="list-style-type: none">Data ModelService ImplementationRuntime ArtifactsService Maintenance

Understanding CRUD Operations in OData

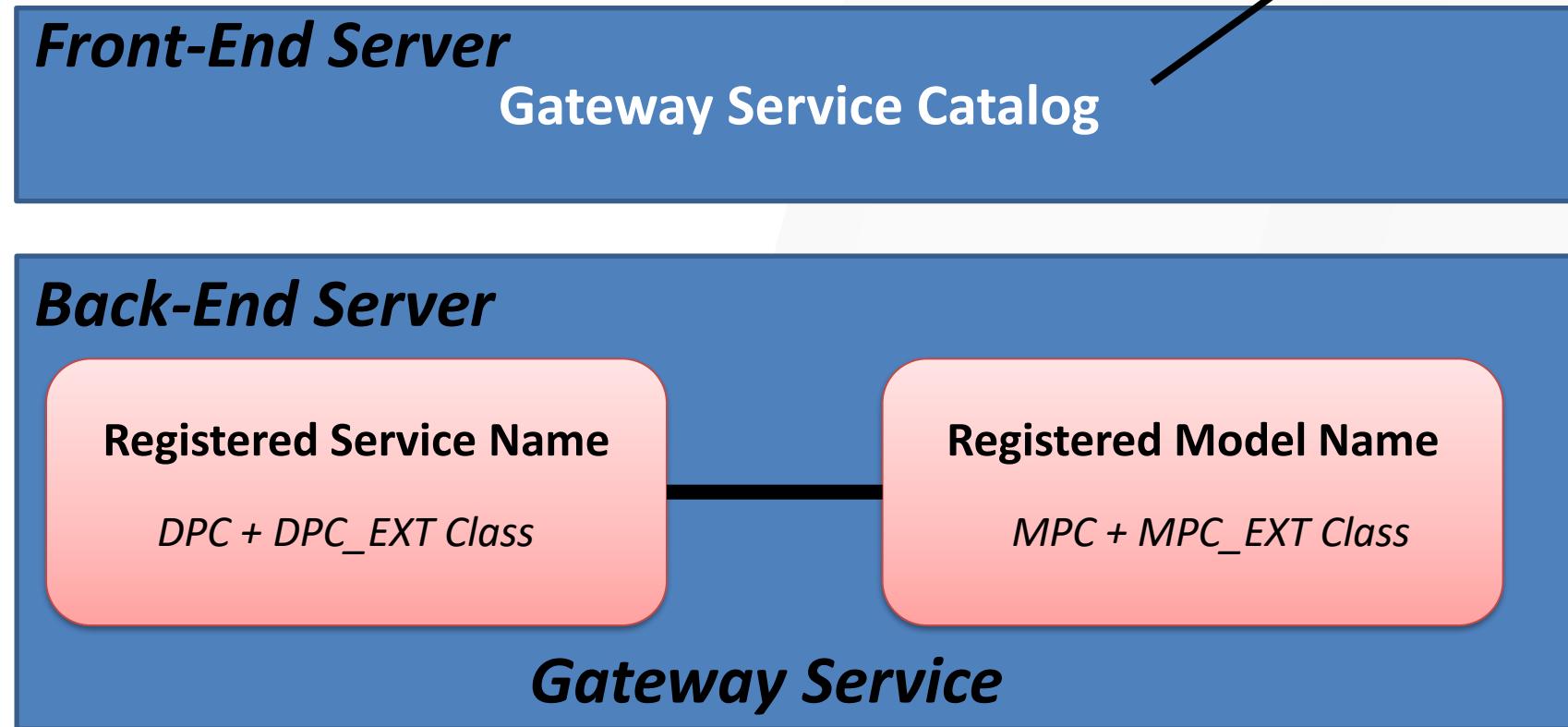
Operation on Resource	HTTP Method	
CREATE	POST	Insert <table> from <workarea>
READ	GET	Select Single * from <table> into <wa>
UPDATE	PUT/PATCH	Modify <table> from <wa> OR Update <table> set.....
DELETE	DELETE	Delete from <table>
QUERY	GET	Select * From <table> Into Table based on WHERE(Conditions derived by Query)

Linking CRUDQ Operations with methods of DPC_EXT Class

Operation on URI	DPC_EXT Methods
CREATE	<Entity_Set>_CREATE_ENTITY
READ	<Entity_Set>_GET_ENTITY
UPDATE	<Entity_Set>_UPDATE_ENTITY
DELETE	<Entity_Set>_DELETE_ENTITY
QUERY	<Entity_Set>_GET_ENTITYSET



Gateway Service Registration and Testing



Registering OData Service and ensuring SICF Node and System Alias Maintained

Activate and Maintain Services

Service Catalog

Type	Technical Service Name	Ver...	Service Description	External Service Name	Namespace	OAuth...	Soft State Status	Processing Mode
BEP	ZLEARN ODATA SRV	1	Odata Demo	ZLEARN ODATA SRV		<input type="checkbox"/>	Not Supported	Routing-based

ICF Node must be active. If it gets deactivated we need to activate it.

In case if System Alias is not maintained we need to add it manually with the help of Technical Service Identifier

ICF Nodes

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

System Aliases

SAP System Alias	Description	Default System	Metadata
LOCAL	Local System Alias	<input checked="" type="checkbox"/>	<input type="checkbox"/>

How to check all details once a service has been registered - /iwbep/reg_service

Display Service

Cleanup Cache Configuration

Service Information

Technical Service Name	ZLEARN_ODATA_SRV
Service Version	1
Description	Odata Demo
External Service Name	ZLEARN_ODATA_SRV
Namespace	
Data Provider Class	ZCL_ZLEARN_ODATA_DPC_EXT
Created By	SYARD27
Changed By	
Package	\$TMP

Extension for Service

Technical Service Name	
Service Version	0

Model Information

Technical Model Name	ZLEARN_ODATA_MDL
Model Version	1
Description	Odata Demo
Model Provider Class	ZCL_ZLEARN_ODATA_MPC_EXT
Created By	SYARD27
Changed By	
Package	\$TMP

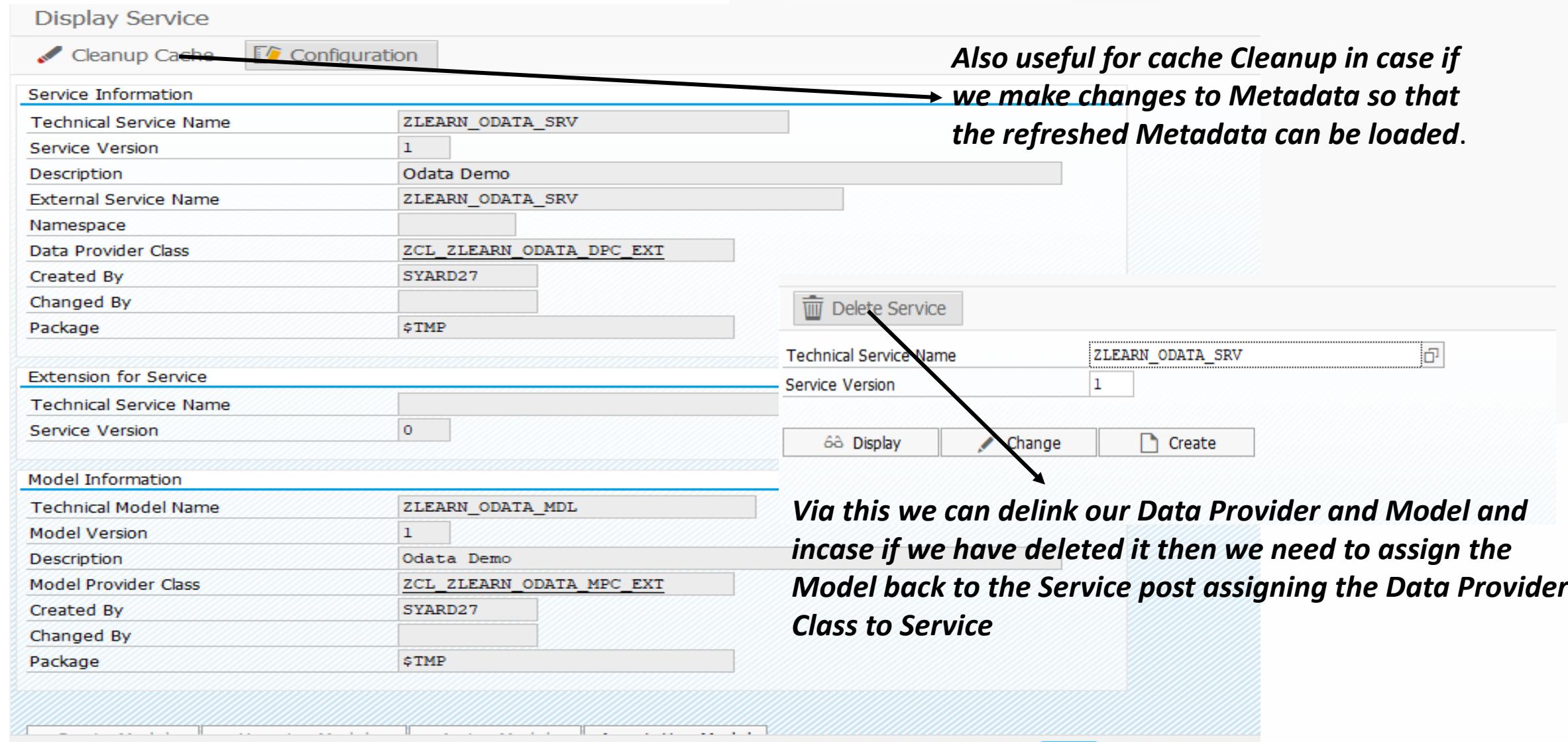
Also useful for cache Cleanup in case if we make changes to Metadata so that the refreshed Metadata can be loaded.

Delete Service

Technical Service Name: ZLEARN_ODATA_SRV
Service Version: 1

Display Change Create

Via this we can delink our Data Provider and Model and incase if we have deleted it then we need to assign the Model back to the Service post assigning the Data Provider Class to Service



Implementing GET Call w.r.t both EntitySet and Entity

The screenshot shows the SAP Gateway Client interface. The top navigation bar includes 'Execute', 'Select', 'Service Administration', 'EntitySets', and 'Add URI Option'. The 'HTTP Method' dropdown is set to 'GET'. The 'Request URI' field contains '/sap/opu/odata/sap/ZLEARN_ODATA_SRV/HeaderSet(Id='100')'. The 'Protocol' is set to 'HTTP'. The 'Test Group' tab is selected, and the 'Test Case' tab is active. The 'HTTP Response' section displays the following details:

Header Name	Value
~status_code	501
~status_reason	Not Implemented

The XML error message is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
- <error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <code>/IWBEPMGW_RT/021</code>
  <message xml:lang="en">Method 'HEADERSET_GET_ENTITY' not implemented in data provider
    class</message>
- <innererror>
  - <application>
    <component_id/>
    <service_namespace>/SAP/</service_namespace>
    <service_id>ZLEARN_ODATA_SRV</service_id>
    <service_version>0001</service_version>
  </application>
<transactionid>7CC752319F2A0050F0060C9656C9F8A9</transactionid>
```

GET Calls for fetching either a Single Record based on Key field or Multiple Data will not happen until and unless the corresponding methods are implemented

In GET Calls w.r.t Single Entity => Key field plays a vital role

The screenshot shows a SAP GUI interface for testing OData services. The top navigation bar has radio buttons for GET, POST, PUT, PATCH, MERGE, DELETE, and HEAD. The 'DELETE' button is highlighted with a yellow box and an arrow pointing to it. The URL entered is `/sap/opu/odata/sap/ZLEARN_ODATA_SRV/HeaderSet(ID='101')`. Below the URL, there are tabs for Test Group and Test Case, with 'Test Group' selected. The main area displays the 'HTTP Response - Processing Time = 47 ms'. It includes a table of headers and their values:

Header Name	Value
~status_code	400
~status_reason	Bad Request

Below the table is the XML error response:

```
<?xml version="1.0" encoding="UTF-8"?>
- <error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
    <code>005056A509B11EE1B9A8FEC11C22578E</code>
    <message xml:lang="en">Invalid key name in key predicate. Expected name is 'Id'</message>
    - <innererror xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
        <transactionid>7CC752319E2A0020E0060AFCAFC97C5C</transactionid>
        <timestampl>2021-06-24T15:21:17.660Z</timestampl>
```

If the key is not provided properly or if all the Key fields are not specified incase if we have multiple Key fields we will get Error in our Response.

Date Representation in XML / JSON Format

```
....  
    <link title="HeaderSet" href="HeaderSet" rel="self"/>  
    - <entry>  
        <id>http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ ('100')</id>  
        <title type="text">HeaderSet('100')</title>  
        <updated>2021-06-23T09:46:36Z</updated>  
        <category scheme="http://schemas.microsoft.com/ado/2007/08/data: term="ZLEARN_ODATA_SRV.Header"/>  
        <link title="Header" href="HeaderSet('100')" rel="self"/>  
        - <content type="application/xml">  
            - <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/ xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservice"  
                <d:Id>100</d:Id>  
                <d:Name>SAM</d:Name>  
                <d:Dept>DEPT1</d:Dept>  
                <d:CreatedOn>2021-04-03T00:00:00</d:CreatedOn>  
            </m:properties>  
        </content>  
    </entry>
```

```
{  
    "_metadata": {  
        "id" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ ('100')",  
        "uri" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ ('100')",  
        "type" : "ZLEARN_ODATA_SRV.Header"  
    },  
    "Id" : "100",  
    "Name" : "SAM",  
    "Dept" : "DEPT1",  
    "CreatedOn" : "\/Date(1617408000000)\/"  
},
```

XML	yyyy-mm-ddThh:mm:ss	Eg:- 2021-03-08T12:34:56
JSON	“\Date(<ticks>)”<ticks> = number of milliseconds since midnight Jan 1, 1970	

Handling Success Message as a part of Response Header

HTTP Method GET POST PUT PATCH MERGE DELETE HEAD

Request URI /sap/opu/odata/sap/ZLEARN_ODATA_SRV/HeaderSet

Protocol HTTP HTTPS Test Group

HTTP Response - Processing Time = 39588 ms

Header Name	Value
sap-message	<notification xmlns:sap="http://www.sap.com/Protocols/SAPData"><code>ZMSG_DEMO/002</code><message>2 rows fetched successfully</message>
sap-perf-fesrec	39580542.000000

HTTP Request

The Message can also sent as a part of Response Header when we are doing a HTTP GET call

```
<?xml version="1.0"?>
- <feed xml:base="http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ODATA_SRV/" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns="http://www.w3.org/2005/Atom">
  <id>http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ODATA_SRV/HeaderSet</id>
  <title type="text">HeaderSet</title>
  <updated>2021-06-23T16:45:50Z</updated>
  - <author>
    <name/>
  </author>
  <link title="HeaderSet" rel="self" href="HeaderSet"/>
  - <entry>
    <id>http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ODATA_SRV/HeaderSet('100')</id>
    <title type="text">HeaderSet('100')</title>
    <updated>2021-06-23T16:45:50Z</updated>
    <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZLEARN_ODATA_SRV.Header"/>
    <link title="Header" rel="self" href="HeaderSet('100')"/>
    - <content type="application/xml">
```

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



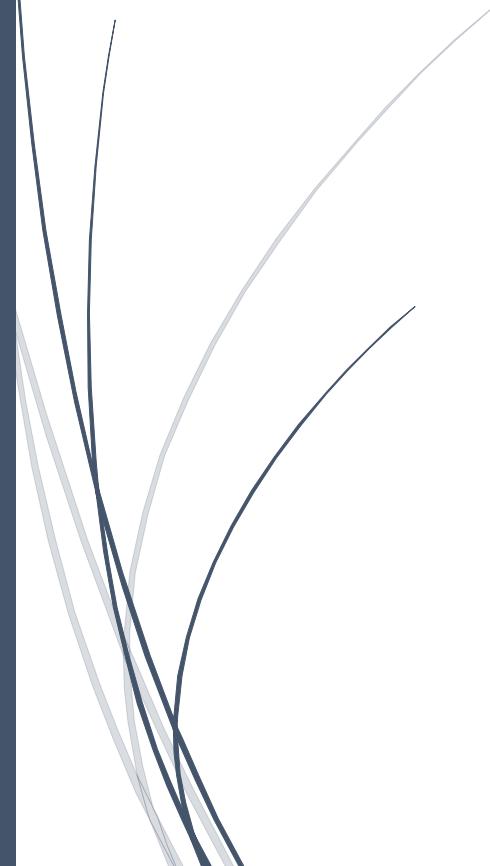
mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



Master SAP OData

Day 2

OData is a way of performing database style create, read, update and delete operations on the resources by using HTTP Verbs.

B.ES => Phase 1 -3

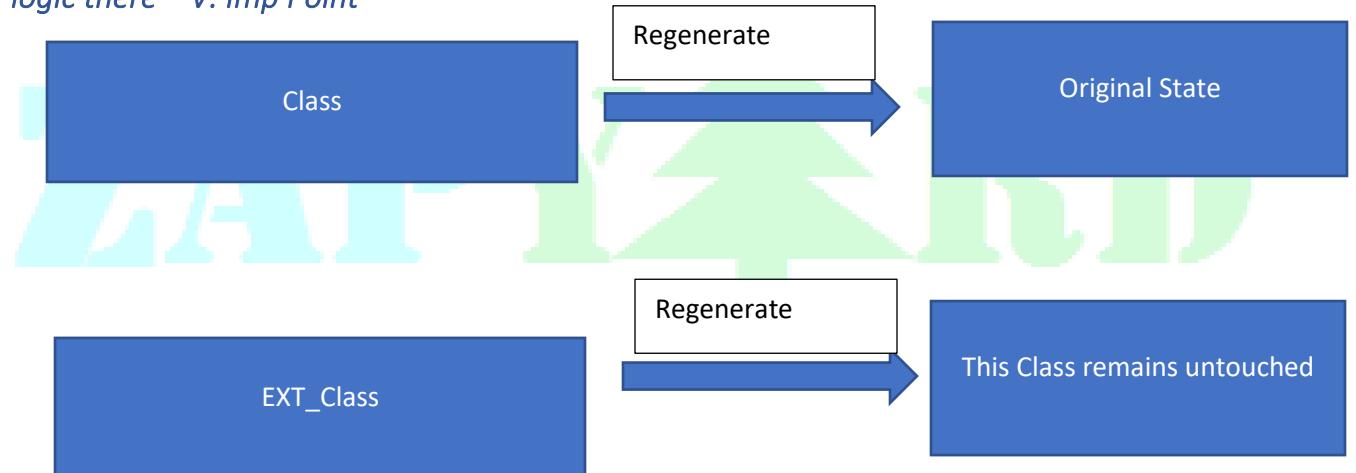
F.ES. => Phase 4

DPC => It will help you to achieve all your CRUDQ Operations on Endpoint -> Entityset.

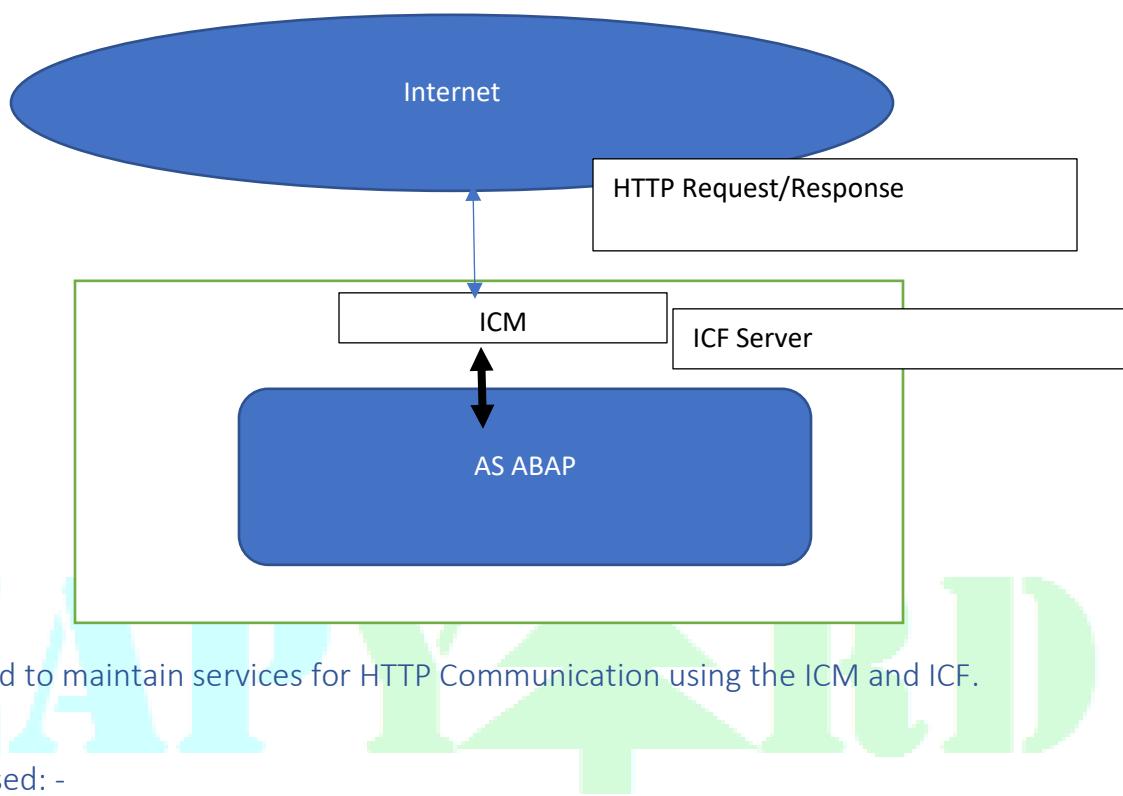
MPC => It helps to provide the Metadata information via my EDM.

<Odata_Project_Name>_SRV => Technical Service Name

*(DPC or MPC) EXT classes remains untouched on regeneration => Hence we write all our logic there * V. Imp Point*



Architecture of ICF and ICM

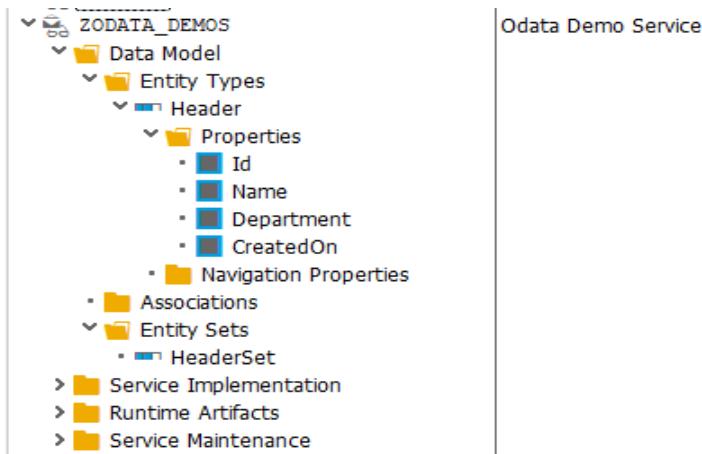


/O/IWBEP/REG_SERVICE => B.E.S

/n/iwbep/cache_cleanup => B.E.S

/n/iwfnd/cache_cleanup => F.E.S

Gateway Service Project:-



Code Backup: -

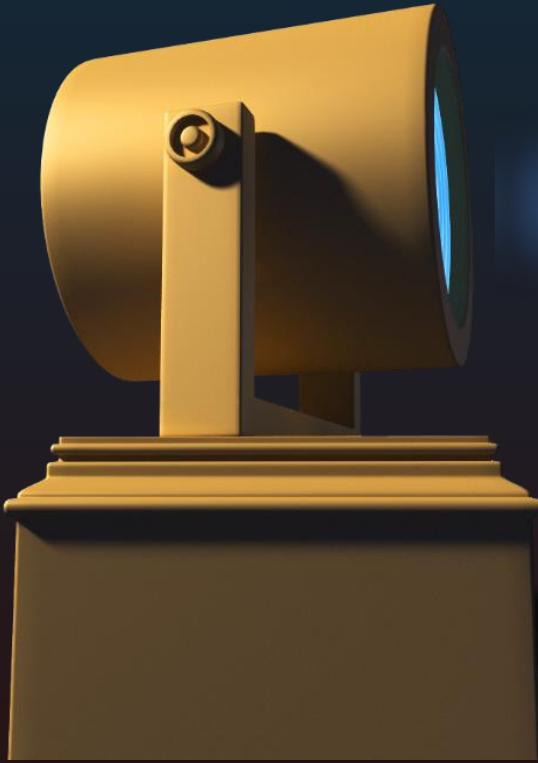
```

METHOD headerset_get_entityset.
    SELECT FROM zheader FIELDS *
    INTO CORRESPONDING FIELDS OF TABLE @et_entityset.
    IF sy-subrc <> 0.
        RAISE EXCEPTION TYPE /iwbep/cx_mgw_tech_exception
        EXPORTING
        textid = /iwbep/cx_mgw_tech_exception=>data_provider_not_found
        RAISE EXCEPTION TYPE /iwbep/cx_mgw_busi_exception
        EXPORTING
        textid = /iwbep/cx_mgw_busi_exception=>resource_not_found.
    ELSE.
        ENDIF.
    ENDMETHOD.

METHOD headerset_get_entity.
    DATA(lv_id) = VALUE #( it_key_tab[ name = 'Id' ]-value OPTIONAL ).
    SELECT SINGLE FROM zheader FIELDS *
    WHERE id = @lv_id
    INTO CORRESPONDING FIELDS OF @er_entity.

ENDMETHOD.

```



Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

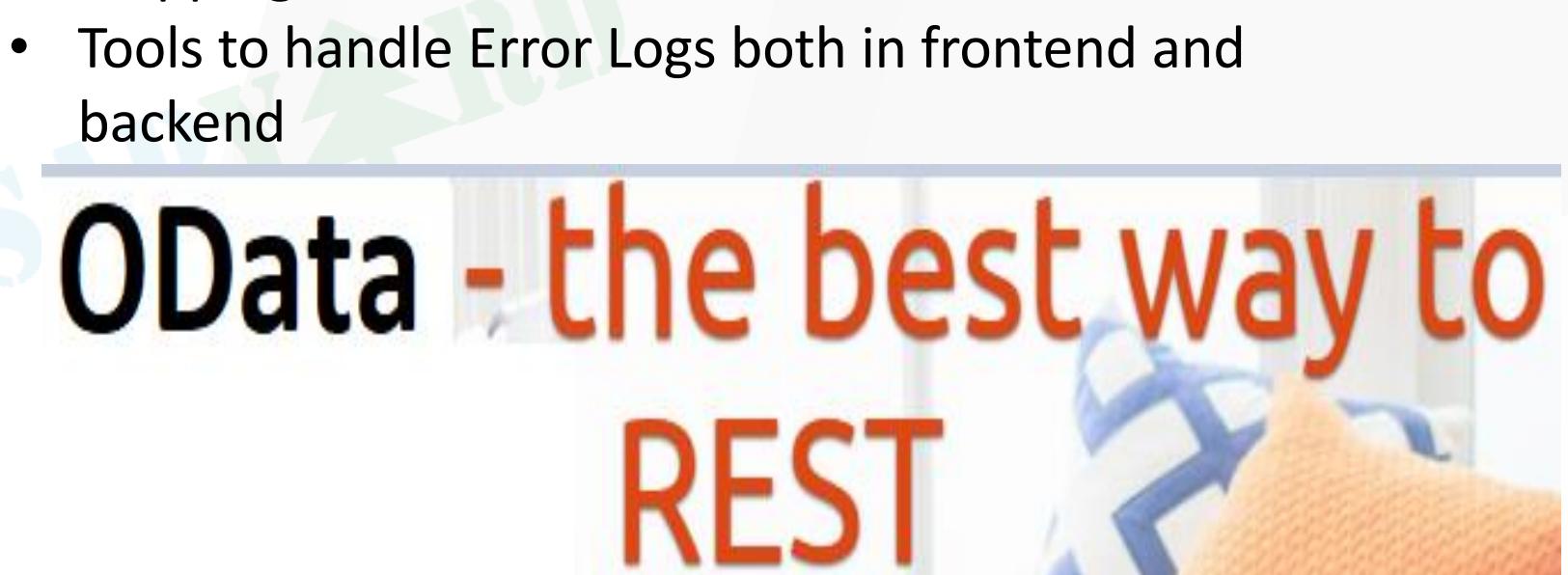
www.ZAPYard.com

ZAPYARD

Agenda

- Implementing
POST,
PUT/PATCH/MERGE,
DELETE Calls
- Creating a Data Model via RFC FM and
implementing GET Calls and understanding the
mapping associated with it
- Tools to handle Error Logs both in frontend and
backend

OData - the best way to REST



Implementing POST Call to Create a New Entity

Properties													
	Name	Is Key	Edm Core Type	Prec.	Scale	Max ...	Unit Prop.	Creat...	Upda...	Sorta...	Nulla...	Filt.	Label
	Id	<input checked="" type="checkbox"/>	Edm.String	0	0	3		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Id
	Name	<input type="checkbox"/>	Edm.String	0	0	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Name
	Dept	<input type="checkbox"/>	Edm.String	0	0	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Department
	CreatedOn	<input type="checkbox"/>	Edm.DateTime	7	0	0		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Created on

Entity Sets

Name	Entity Type Name	Label	La...	Semantics	Creat...	Upda...
HeaderSet	Header	T			<input checked="" type="checkbox"/>	<input type="checkbox"/>

```
version="1" Namespace="ZLEARN_ODATA_SRV">
- <EntityType sap:content-version="1" Name="Header">
  - <Key>
    <PropertyRef Name="Id"/>
  </Key>
  <Property sap:updatable="false" Name="Id" sap:filterable="false" sap:sortable="false"
    sap:label="Id" sap:unicode="false" MaxLength="3" Nullable="false" Type="Edm.String"/>
  <Property sap:updatable="false" Name="Name" sap:filterable="false" sap:sortable="false"
    sap:label="Name" sap:unicode="false" MaxLength="10" Nullable="false" Type="Edm.String"/>
  <Property sap:updatable="false" Name="Dept" sap:filterable="false" sap:sortable="false"
    sap:label="Department" sap:unicode="false" MaxLength="10" Type="Edm.String"/>
  <Property sap:updatable="false" Name="CreatedOn" sap:filterable="false" sap:sortable="false"
    sap:label="Created on" sap:unicode="false" Nullable="false" Type="Edm.DateTime"
    Precision="7"/>
</EntityType>
```

If we enable the checkbox w.r.t Creatable for both Entity and Entity Set we can analyze from the Service Metadata and the Service Document by going through its properties

Implementing POST Call to Create a New Entity – Logic + Testing

First we need to do a GET Call so that we can understand what properties we need to send for creation of entity and also the same response can be used as Request

SAP Gateway Client

HTTP Method: POST (highlighted)

Request URI: /sap/opu/odata/SAP/ZLEARN_ODATA_SRV/HeaderSet

Protocol: HTTP

Test Group: Test Case

HTTP Response - Processing Time = 48032 ms

Header Name	Value
~status_code	201
~status_reason	Created
sap-processing-info	ODataBEP=,crp=,st=,MedCacheHub=SHM,codeployed=X,softstate=

HTTP Request

Header Name	Value

XML Request:

```
<?xml version="1.0" encoding="utf-8"?>
<entry xml:base="http://Six30.mydomain.com:8770/sap/opu/odata/SAP/ZLEARN_ODATA_SRV/HeaderSet('101')">
  <id>http://Six30.mydomain.com:8770/sap/opu/odata/SAP/ZLEARN_ODATA_SRV/HeaderSet('101')</id>
  <title type="text">HeaderSet('101')</title>
  <updated>2021-06-25T17:30:06Z</updated>
  <category term="ZLEARN_ODATA_SRV.Header" scheme="">
    <link href="HeaderSet('101')" rel="self" title="" type="application/xml"/>
  </category>
  <content type="application/xml">
    <m:properties>
      <d:Id>102</d:Id>
      <d:Name>Jimmy</d:Name>
      <d:Dept>Dept3</d:Dept>
    </m:properties>
  </content>
</entry>
```

XML Response:

```
<?xml version="1.0" encoding="UTF-8"?>
- <entry xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xml:base="http://Six30.mydomain.com:8770/sap/opu/odata/SAP/ZLEARN_ODATA_SRV/"
  xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
  xmlns="http://www.w3.org/2005/Atom">
  <id>http://Six30.mydomain.com:8770/sap/opu/odata/SAP/ZLEARN_ODATA_SRV/HeaderSet('102')</id>
  <title type="text">HeaderSet('102')</title>
  <updated>2021-06-25T17:34:38Z</updated>
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"
    term="ZLEARN_ODATA_SRV.Header"/>
  <link title="Header" href="HeaderSet('102')" rel="self"/>
  <content type="application/xml">
    - <m:properties>
      <d:Id>102</d:Id>
      <d:Name>Jimmy</d:Name>
      <d:Dept>Dept3</d:Dept>
      <d:CreatedOn>2021-06-25T00:00:00</d:CreatedOn>
    </m:properties>
```

The request can be sent in both XML and JSON format

Implementing DELETE call to Delete an already Created Entity

The screenshot shows the SAP OData Test Client interface. At the top, there is a table titled "Entity Sets" with columns: Name, Entity Type Name, Label, La..., Semantics, Creat..., Upda..., Delete..., and Page. A row for "HeaderSet" is selected, showing "Header" in the Entity Type Name column and a checkmark in the Delete... column. Below this, the main area has tabs for "HTTP Method" (with "DELETE" selected), "Request URI" (/sap/opu/odata/SAP/ZLEARN_ODATA_SRV/HeaderSet('103')), "Protocol" (HTTP selected), and "Test Group" (selected). The "Test Case" tab is active. On the left, there are buttons for "Add File" and "Remove File". On the right, there are buttons for "Response in Browser", "Error Log", "HTTP Header", "Use as Request", "Data Explorer", and "Copy". The "HTTP Response" section shows a processing time of 39505 ms. It lists headers: ~status_code (Value: 204) and ~status_reason (Value: No Content). Below the response, the XML body is shown: <?xml version="1.0"?> <RESPONSE_BODY/>. A large black arrow points from the text "If the data gets deleted successfully then we get 204 as the HTTP Status Response otherwise we will get Error" down to the "Value: 204" entry in the response table.

*If the data gets deleted successfully then we get 204 as the HTTP Status Response
otherwise we will get Error*

Implementing PUT call to update an entity

The screenshot shows the SAP OData Test Client interface. At the top, there are buttons for GET, POST, PUT, PATCH, MERGE, DELETE, and HEAD, with PUT being the selected method. Below the buttons is a URL input field containing `/sap/opu/odata/sap/ZLEARN_ODATA_SRV/HeaderSet`. Underneath the URL, there are tabs for HTTP and HTTPS, with HTTP selected. To the right of the URL is a "Test Case" button. A large arrow points from the title "Implementing PUT call to update an entity" down to the "PUT" button.

The main area is divided into several sections:

- Request:** On the left, it shows the JSON payload:

```
"d" : {  
  "_metadata" : {  
    "id" : "http://Six30.mydomain.com:8770/sap/  
    "uri" : "http://Six30.mydomain.com:8770/sap/  
    "type" : "ZLEARN_ODATA_SRV.Header"  
  },  
  "Id" : "102",  
  "Dept" : "Dept_3"  
}
```
- HTTP Response - Processing Time = 54 ms:** This section displays the response headers and the XML error message.

Header Name	Value
~status_code	405
~status_reason	Method Not Allowed

The XML error message is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>  
- <error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">  
  <code>005056A509B11ED199D8826D151FC0FE</code>  
  <message xml:lang="en">The specified HTTP method is not allowed for the resource identified by the Data  
    Service Request URI</message>  
  - <innererror xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">  
    <transactionid>7CC752319E2A0070E0060AFCAFCE07A</transactionid>  
    <timestamp>20210630153801.0194350</timestamp>  
    - <ErrorResolution>
```

Even for PUT Call we need to ensure that we pass the Key property in the Request URI and also keep a close watch on the payload that we are sending as request because PUT indicates a replacement update.

Implementing PATCH/MERGE call to modify an entity

The screenshot shows a SAP GUI test environment. The top navigation bar has radio buttons for GET, POST, PUT, PATCH, MERGE, DELETE, and HEAD, with PATCH and MERGE selected. The URL is /sap/opu/odata/sap/ZLEARN_ODATA_SRV/HeaderSet('102'). Below the URL, it says HTTP is selected. The main area is divided into Request and Response sections. The Request section shows a JSON patch:

```
{  
  "d" : {  
    "__metadata" : {  
      "id" : "http://Six30.mydomain.com:8770/sap/  
      "uri" : "http://Six30.mydomain.com:8770/sap/  
      "type" : "ZLEARN_ODATA_SRV.Header"  
    },  
    "Dept" : "Dept_3"  
  }  
}
```

The Response section shows the following details:

HTTP Response - Processing Time = 66 ms

Header Name	Value
~status_code	204
~status_reason	No Content

Response Body (highlighted in red):

```
<?xml version="1.0"?>  
<RESPONSE_BODY/>
```

IN PATCH/MERGE Call, the beauty is even if we pass only the data that needs to be updated since either PATCH or MERGE indicates a different update hence only the relevant data gets modified and rest of the data remains as it is.

Creating a Data Model by mapping RFC FMs and Service Implementation

Operations of entity set ProductSet							
Operation Name	Data Source Group	Data Source Name	Data Source Type	RFC Destination	Implementation Class	Method Name	Link to mapping node
Create					ZCL_ZLEARN_ODAT...	PRODUCTSET_CREA...	
Delete					ZCL_ZLEARN_ODAT...	PRODUCTSET_DELETE...	
GetEntity (Read)	SEPM_PRODUCT_BAPI	BAPI_EPM_PRODUC...	Remote Function Call		ZCL_ZLEARN_ODAT...	<u>PRODUCTSET_GET...</u>	<u>Mapping</u>
GetEntitySet (Query)	SEPM_PRODUCT_BAPI	BAPI_EPM_PRODUC...	Remote Function Call		ZCL_ZLEARN_ODAT...	<u>PRODUCTSET_GET...</u>	<u>Mapping</u>
Update					ZCL_ZLEARN_ODAT...	PRODUCTSET_UPDA...	

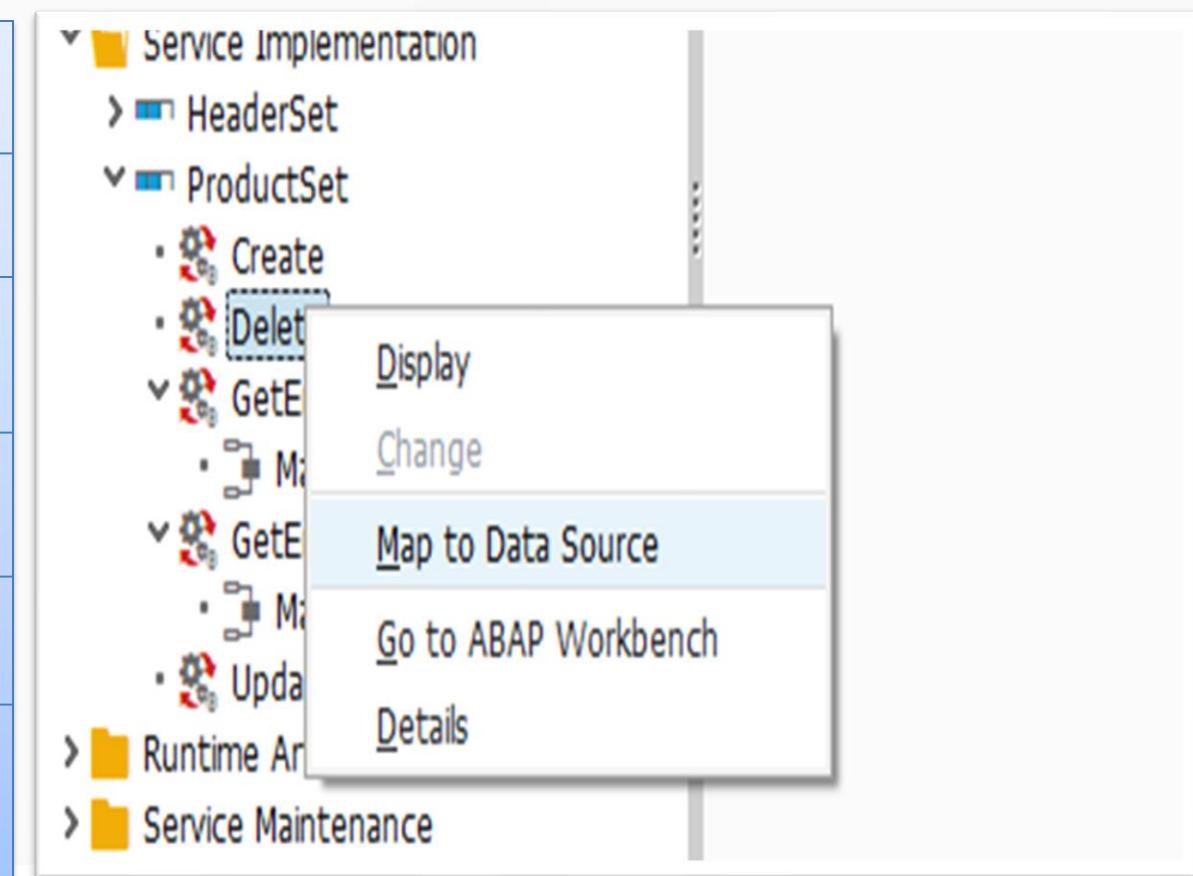
Mapping can be automated by Propose Mapping

Mapping of Operation GetEntitySet (Query) for ProductSet	
Pr... Entity Set property	Constant Value
ProductId	HEADERDATA\PRODUCT_ID
TypeCode	HEADERDATA\TYPE_CODE
Category	HEADERDATA\CATEGORY
Name	HEADERDATA\NAME
Description	HEADERDATA\DESCRIPTION
SupplierName	HEADERDATA\SUPPLIER_NAME

If we map the RFC FM for the Service Implementation Logic we need to regenerate the Runtime Objects so that our FM and its respective logic automatically gets handled in the DPC Class.

Linking CRUDQ Operations with methods of DPC by mapping of RFC FM as Data Source

Operation on URI	DPC Methods
CREATE	<Entity_Set>_CREATE_ENTITY
READ	<Entity_Set>_GET_ENTITY
UPDATE	<Entity_Set>_UPDATE_ENTITY
DELETE	<Entity_Set>_DELETE_ENTITY
QUERY	<Entity_Set>_GET_ENTITYSET



Once the mapping is completed, generate run time objects and go to your Data Provider Class(DPC) to find the ABAP coding has been generated automatically.

Handling error logs in both F.E.S and B.E.S

The screenshot shows the SAP GUI interface with the 'Error Log' tab selected. The title bar indicates 'HTTP Response - Processing Time = 83 ms'. The error details are as follows:

Header Name	Value
~status_code	500
~status_reason	Internal Server Error

The main area displays an XML error response:

```
<?xml version="1.0" encoding="UTF-8"?>
- <error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <code>0050569259751EE4BA9710043F8A5115</code>
  <message xml:lang="en">In the context of Data Services an unknown internal server error occurred</message>
- <innererror xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <transactionid>7CC752319E2A0030E0060C5960394B00</transactionid>
  <timestam...>20210630161140.9817110</timestam...
```

*This is where we can access the Error Log if any w.r.t accessing the Service URI
OR*

We can use the T-codes Directly as well

Front-End Server

/n/iwfnd/error_log

Back-End Server

/n/iwbep/error_log

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



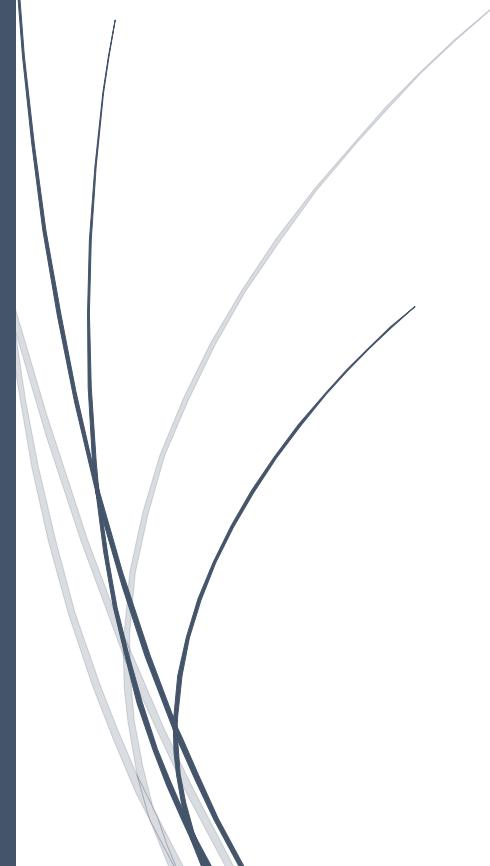
mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



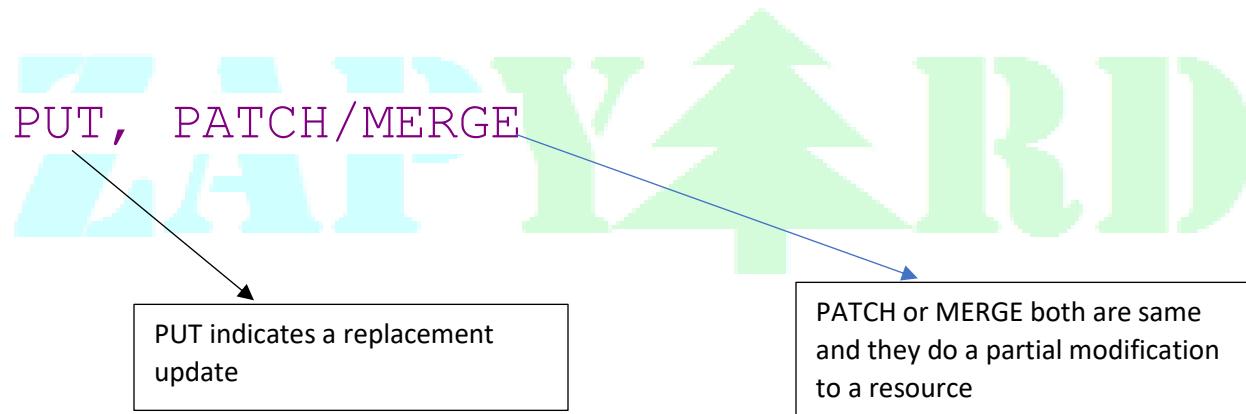
Day 3

"To get the Key-Value Pairs: - (You can use it key tab as well)

```
DATA:lt_keys TYPE /iwbep/t_mgw_tech_pairs.  
lt_keys = io_technology_request_context->get_keys( ).
```

For GET Single Entity, UPDATE and DELETE => We need to pass Key field in the URI otherwise Error will be raised from the F.E.S

For CREATE & GET Entity Set=> We will not pass any Key field along with Value in then URI



Logic for CREATE (POST) : -

```
METHOD header_set_create_entity.  
DATA:ls_input_data TYPE zcl_zodata_demos_mpc->ts_header.  
io_data_provider->read_entry_data( IMPORTING es_data = ls_input_data ).  
ls_input_data-created_on = sy-datum.  
INSERT zheader FROM ls_input_data.  
IF sy-subrc EQ 0.  
    er_entity = ls_input_data.  
ELSE.  
    RAISE EXCEPTION TYPE /iwbep/cx_mgw_busi_exception  
    EXPORTING
```

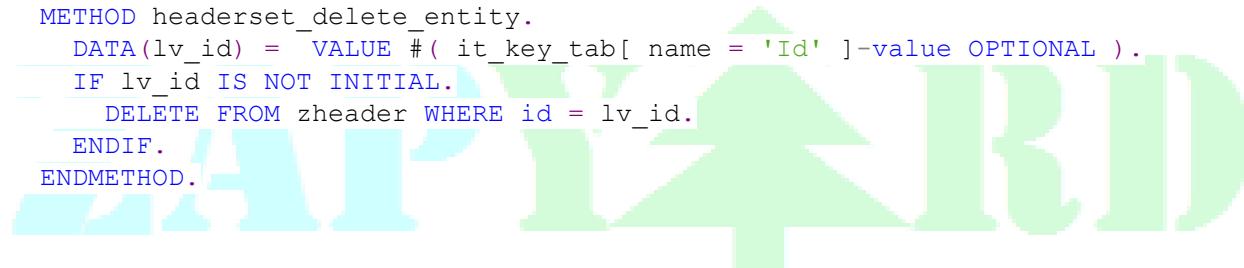
```
textid  = /iwbp/cx_mgw_busi_exception=>business_error_unlimited  
message = 'Error occurred while creating a new record'.  
ENDIF.  
ENDMETHOD.
```

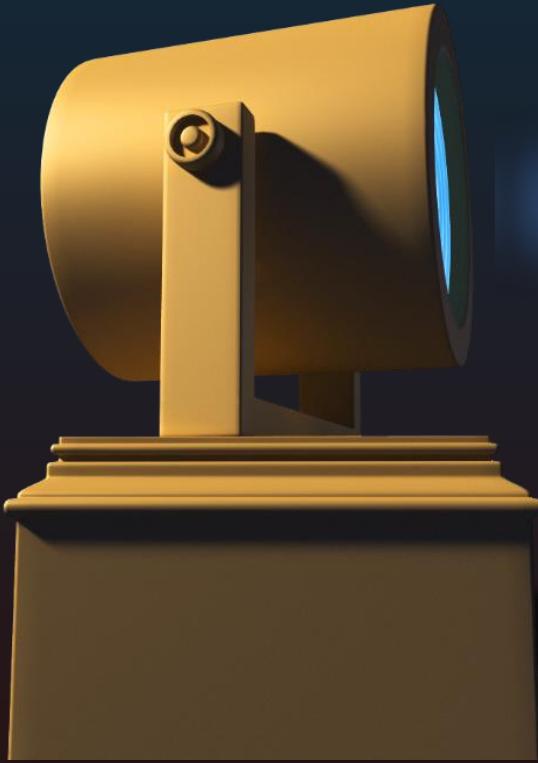
Logic for UPDATE (PUT/PATCH/MERGE) :-

```
METHOD headerset_update_entity.  
DATA:ls_input_data TYPE zcl_zodata_demos_mpc=>ts_header.  
io_data_provider->read_entry_data( IMPORTING es_data = ls_input_data ).  
MODIFY zheader FROM ls_input_data.  
IF sy-subrc EQ 0.  
    er_entity = ls_input_data.  
ENDIF.  
ENDMETHOD.
```

Logic for DELETE (DELETE) : -

```
METHOD headerset_delete_entity.  
DATA(lv_id) = VALUE #( it_key_tab[ name = 'Id' ]-value OPTIONAL ).  
IF lv_id IS NOT INITIAL.  
    DELETE FROM zheader WHERE id = lv_id.  
ENDIF.  
ENDMETHOD.
```





Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

www.ZAPYard.com

ZAPYARD

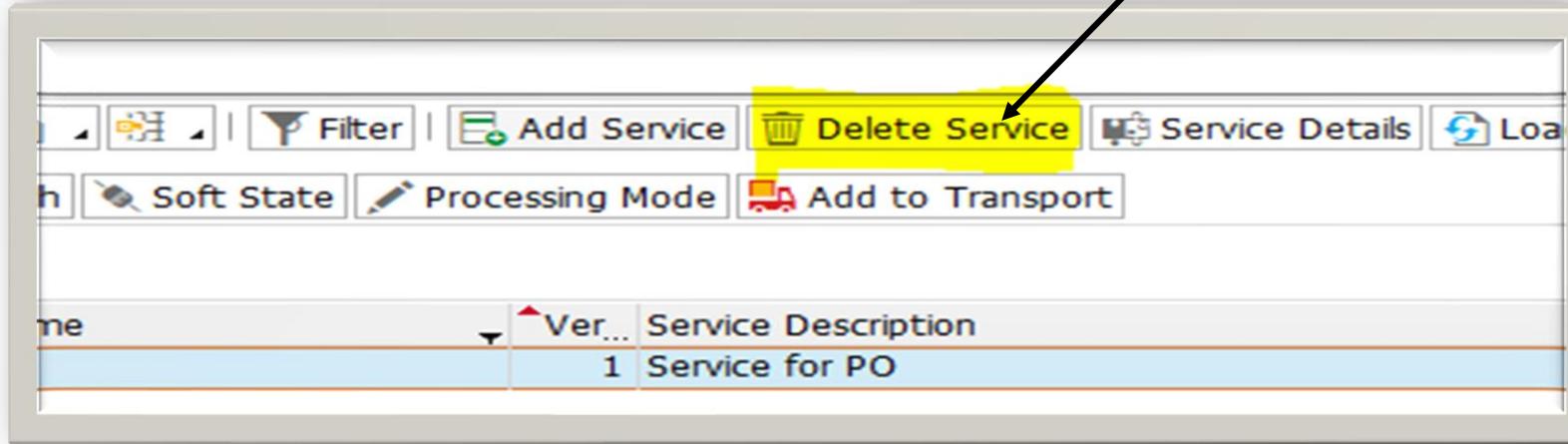
Agenda

- Deletion of an OData Service
- Authentication Options for an OData Service
- Understanding different Queries w.r.t GET Calls
- Handling:-
 - ❖ Filtering=>\$filter



Deletion of an OData Service via Activate and Maintain Services on Gateway

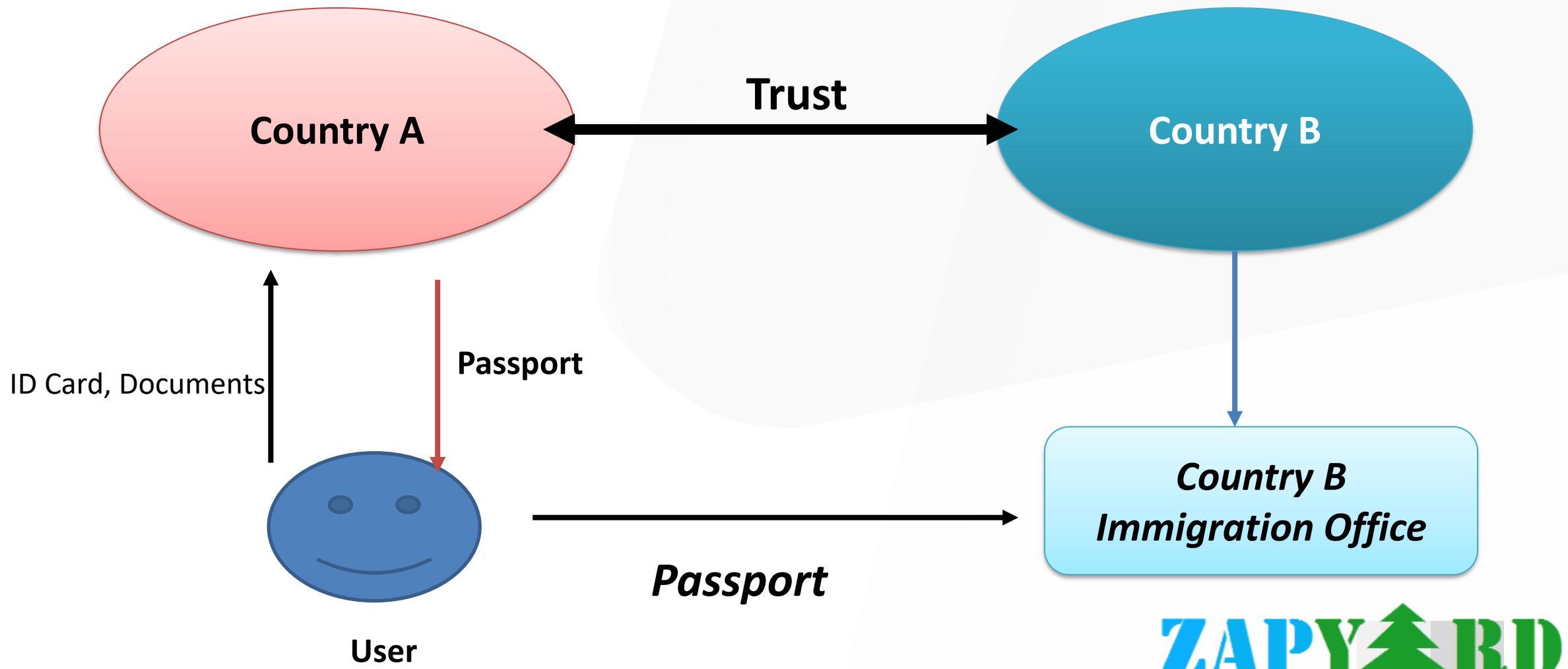
Is it so easy to delete a Service by the Click of Delete??



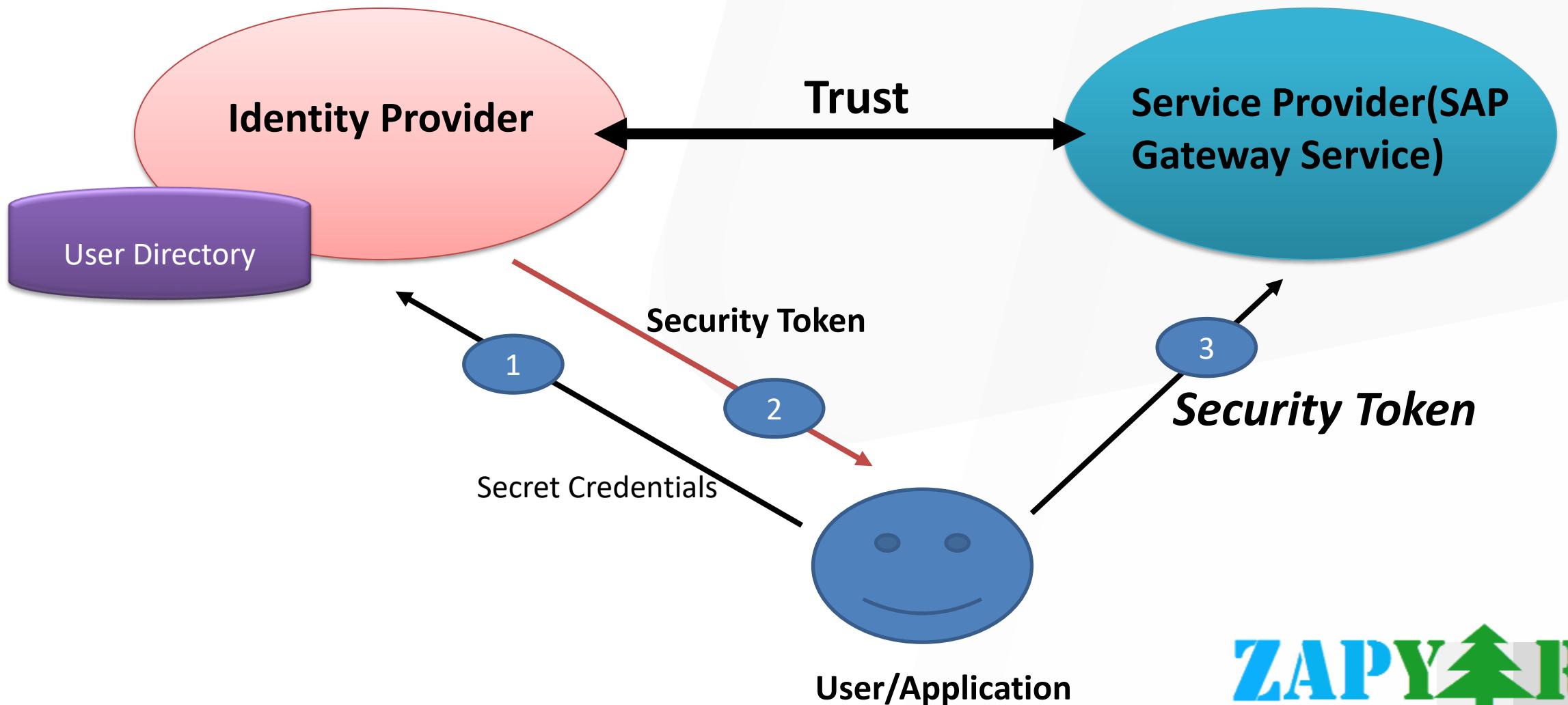
3 Steps to delete a Service:-

1. Remove System Alias
2. Delete ICF Node
3. Delete the Service finally

Enabling Authentication for a OData Service



Enabling Authentication for a OData Service – Fundamentals



Authentication Options Supported by SAP Gateway Hub

- ❖ SAML 2.0 Browser SSO(Security Assertion Markup Language)
- ❖ Basic(Username and Password)
- ❖ Kerberos Tickets(Works only in Intranet Scenarios)
- ❖ OAuth
- ❖ SAP Logon Tickets
- ❖ X.509 Client Certificate

For extranet scenarios, SAML 2.0 Browser Protocol or OAuth are recommended

Enabling OAuth2 Scope for accessing a service with OAuth 2.0

CREATION INFORMATION

Package Assignment Local Object

ICF Node

Standard Mode None

Set Current Client as Default Client in ICF Node

OAuth enablement

Enable OAuth for Service



This checkbox needs to be enabled while registering the service on Gateway

For deleting an OAuth Scope we need to execute the report:- /IWFND/R_OAUTH_SCOPES

Handling Filtering w.r.t GET Calls

Query Parameter:- \$filter = <fieldname> <filter_operator> <value>

Single Filter Condition

Multiple Filter Condition =>

Where every condition is differentiated by => and

Valid values can be of string, integer, float, true , false, null, date, timestamp..

*Any of the property of EDM on which filter condition is applied.
Any invalid property name leads to a bad request.*

We can group Multiple filter conditions using the separators ‘and’ and ‘or’ and also make use of parenthesis to group the filter conditions to execute them in a particular order of sequence.

IV_FILTER_STRING and IT_FILTER_SELECT_OPTIONS=> Two Parameters which can be used w.r.t Filtering

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



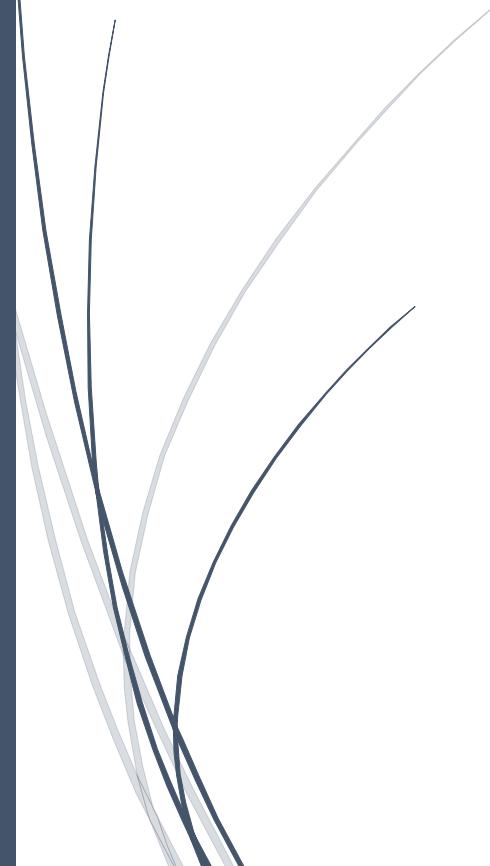
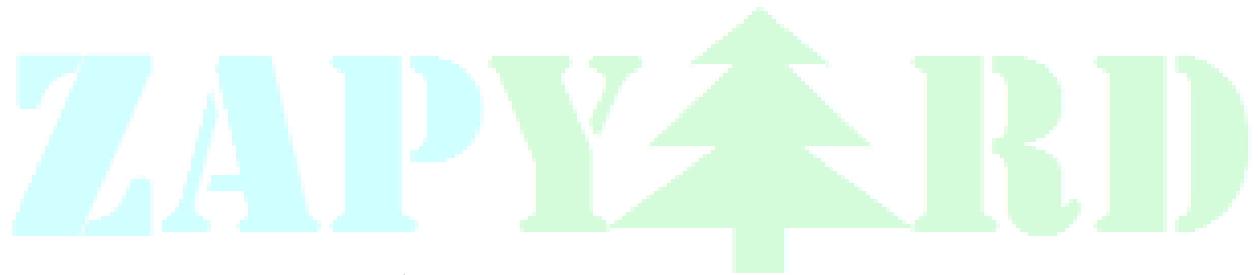
mailZAPYard@gmail.com



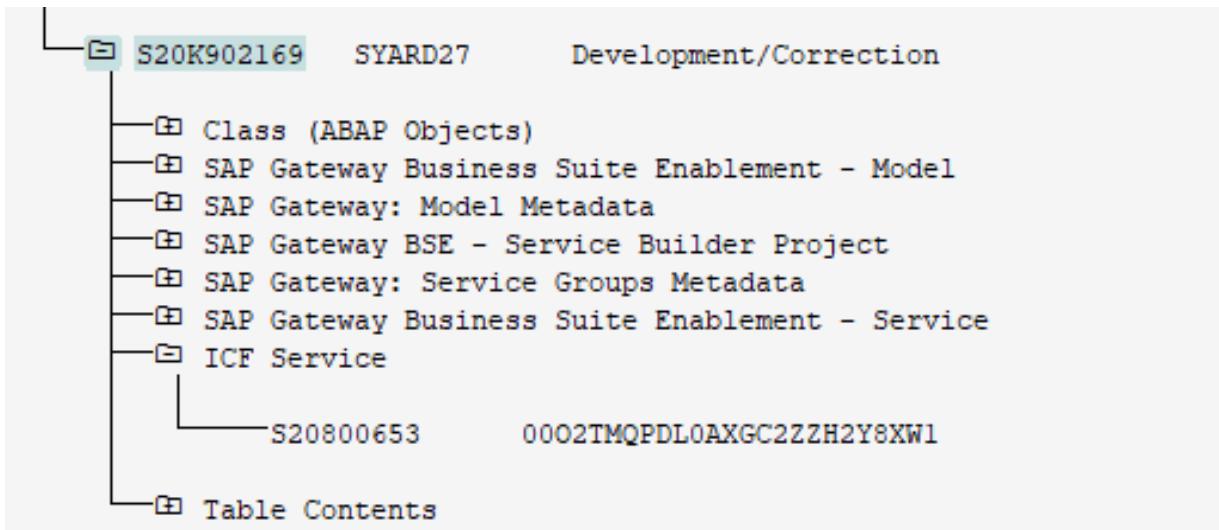
+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



Day 4



Imp :- Move the TR with the deleted entries to delete the same from your Dev and QA Landscape as well



Authentication w.r.t OData Services:-

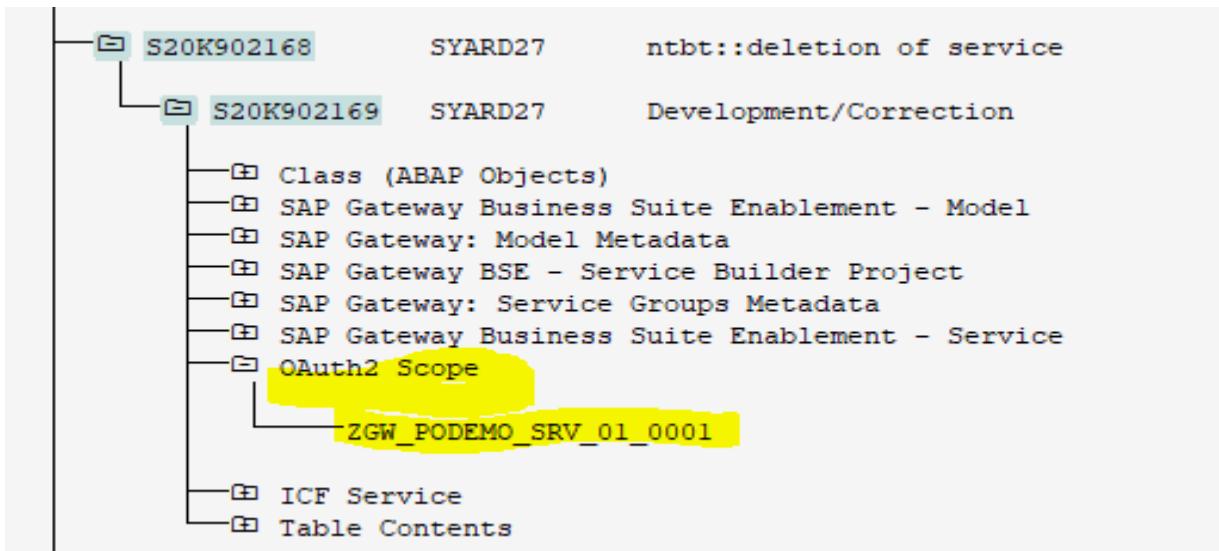
SSO is the most efficient way to facilitate user authentication in an enterprise.

SAP Gateway is responsible for the following:

OData Support, Authentication and Protection of business functions.

Master SAP OData

Post Enabling OAuth:-



Report for Creation or Deletion of OAuth scope:-

The screenshot shows the SAP report interface for creating or deleting OAuth scopes. The title bar says 'Program /IWFND/R_OAUTH_SCOPES'. A 'Doc Identifier' field is highlighted with a yellow box and has an arrow pointing to it from the previous screenshot. The 'Service Details' section contains a 'Service Doc. Identifier' field with the value 'ZGW_PODEMO_SRV_01_0001'. The 'Scope' section includes a 'Description of the Scope' field and two radio buttons: 'Create Oauth scope for service' (unchecked) and 'Delete Oauth scope of service' (checked). The 'Additional parameters' section contains 'Transport request' set to 'S20K902168' and 'Package assignment' set to 'ZABAP_TRNG'.

Filtering in SAP OData:-

The filter operator => eq, ne, gt, lt, ge, le

/sap/opu/odata/sap/ZODATA_DEMOS_SRV/Product_DetailSet?\$filter=ProductId eq 'HT-1000' and Category eq 'Notebooks'&\$format=json

/sap/opu/odata/sap/ZODATA_DEMOS_SRV/Product_DetailSet?\$filter=ProductId eq 'HT-1000' or ProductId eq 'HT-1001'&\$format=json



"Code w.r.t Filter String: -

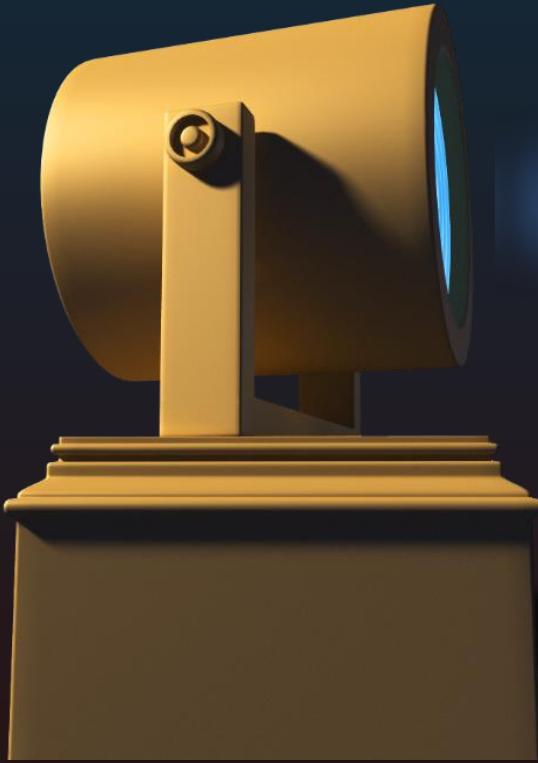
"The Alternative to get filter String:-

```
DATA(lv_string) = io_tech_request_context->get_filter( )-
>get_filter_string( ).
```

```
IF iv_filter_string IS NOT INITIAL.
  SELECT FROM zheader FIELDS *
    WHERE (iv_filter_string)
    INTO CORRESPONDING FIELDS OF TABLE @et_entityset.
ELSE.
  SELECT FROM zheader FIELDS *
    INTO CORRESPONDING FIELDS OF TABLE @et_entityset.
ENDIF.
```

Code w.r.t IT FILTER SELECT OPTIONS:-

```
METHOD productdetailset_get_entityset.  
  DATA: lt_headerdata TYPE STANDARD TABLE OF bapi_epm_product_header,  
        lt_product_id TYPE TABLE OF bapi_epm_product_id_range,  
        ls_product_id TYPE bapi_epm_product_id_range,  
        lt_category    TYPE STANDARD TABLE OF bapi_epm_product_categ_range.  
  
  DATA(lt_filters) = io_tech_request_context->get_filter( )-  
>get_filter_select_options( ).  
  
  READ TABLE it_filter_select_options INTO DATA(ls_filter) WITH KEY property  
y = 'ProductId'.  
  IF sy-subrc EQ 0.  
    LOOP AT ls_filter-select_options INTO DATA(ls_so).  
      MOVE-CORRESPONDING ls_so TO ls_product_id.  
      APPEND ls_product_id TO lt_product_id.  
    ENDLOOP.  
  ENDIF.  
  IF line_exists( lt_filters[ property = 'CATEGORY' ] ).  
    DATA(ls_category) = lt_filters[ property = 'CATEGORY' ]-  
select_options[ 1 ].  
    lt_category = VALUE #( ( sign = ls_category-sign option = ls_category-  
option low = ls_category-low ) ).  
  ENDIF.  
  
  CALL FUNCTION 'BAPI_EPM_PRODUCT_GET_LIST'  
* EXPORTING  
*   MAX_ROWS           =  
*   TABLES  
  headerdata          = lt_headerdata  
  selparamproductid  = lt_product_id  
*   SELPARAMSUPPLIERNAMES      =  
  selparamcategories = lt_category.  
  et_entityset = CORRESPONDING #( lt_headerdata ).  
ENDMETHOD.
```



Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | WhatsApp: +1-251-727-9273

www.ZAPYard.com

ZAPYARD

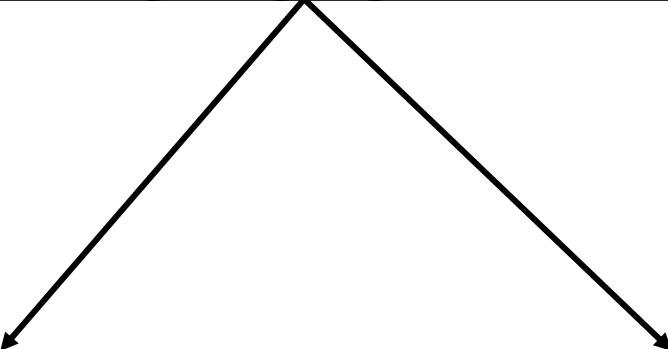
Agenda

- Handling Queries with GET Calls:-
 - ❖ Paging=> \$top, \$skip, \$skiptoken, \$inlinecount
 - ❖ Sorting => \$orderby
 - ❖ Select with GET Calls=>\$select
 - ❖ Search => search
- Analyze SAP OData Services Performance with the help of Traces

**OData - the best way to
REST**

ZAPYARD

Implementing Paging w.r.t GET Calls



Client Side Paging:- \$top and \$skip

IS_PAGING => Parameter to be used

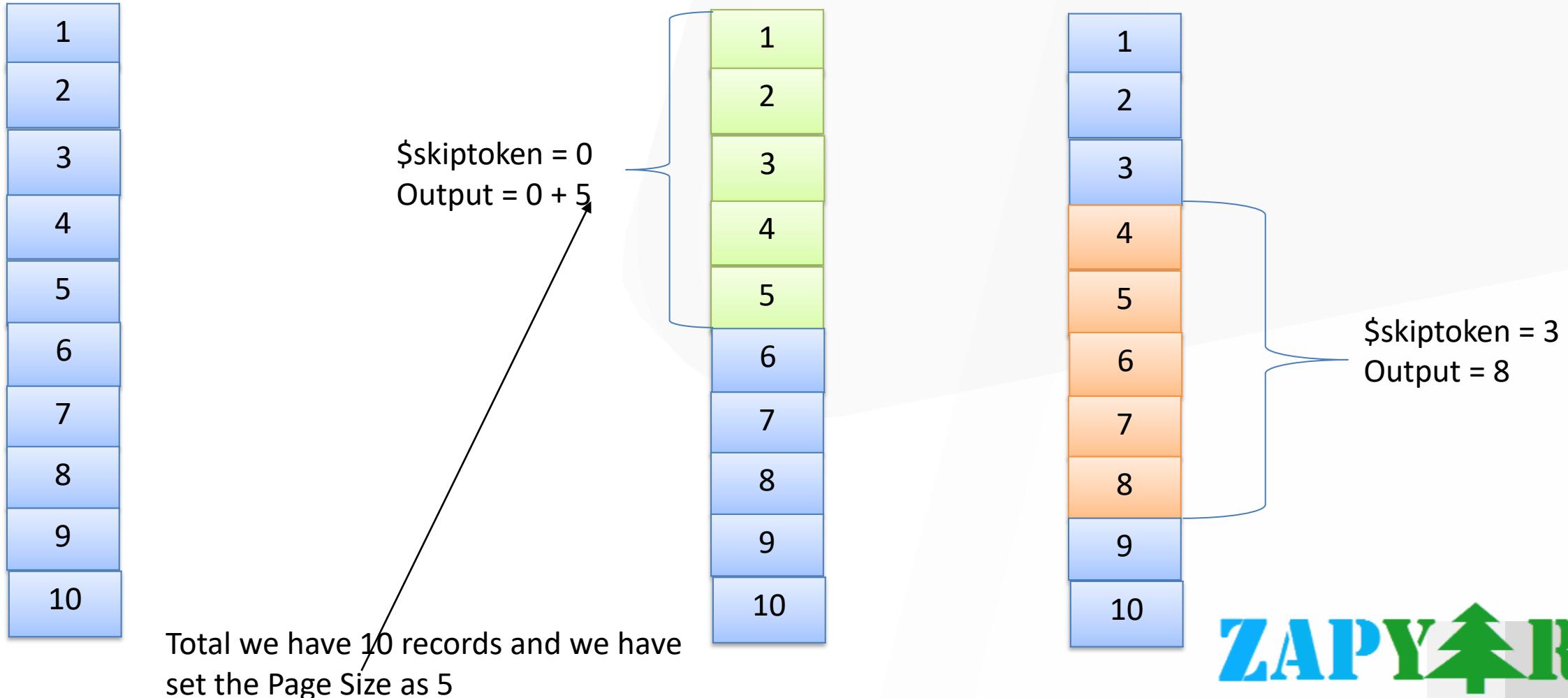
Server Side Paging:- \$skiptoken query

Here the server will only return a portion of results and will provide a “next link” information; which can be followed to get the next set of Information.

```
},  
    ProductId: "HT-1001",  
    Category: "Notebooks",  
    Name: "Notebook Basic 17",  
    Description: "Notebook Basic 17 with 2,80 GHz quad core, 17\" LCD, 4 GB DDR3 RAM, 500 GB Hard Disc, Windows 8 Pro",  
    SupplierId: "100000047",  
    SupplierName: "Becker Berlin",  
    Price: "1249.0000",  
    CurrencyCode: "EUR"  
],  
    }  
],  
    _next: "http://103.44.1.51:8770/sap/opu/odata/sap/ZLEARN\_ODATA\_SRV/Product\_DetailSet?\$filter=ProductId%20eq%20%27HT-1000%27%20or%20ProductId%20eq%20%27HT-1001%27%20or%20ProductId%20eq%20%27HT-1002%27&\$format=json&\$skiptoken=2"  
}
```

Understanding the Concept of \$skiptoken => Query applied to EntitySet Only

- We need to define the Page Size at the server side.
- If the entries in the Entity Set are more than Page Size, we will reduce the entries.
- We will send those reduced entries based on skiptoken value and also the next link for next set of entries if any.



Understanding difference between=> \$inlinecount=allpages and \$count

Query option to get the Count along with all the entries as a part of response

The screenshot shows a ZAP test case interface. The URL bar contains `http://Six30.mydomain.com/Product_DetailSet?$inlinecount=allpages`. The main panel displays an **HTTP Response - Processing Time = 1058 ms**. The response headers show `~status_code: 200` and `~status_reason: OK`. Below the headers, a table lists parameters for the `Product_DetailSet` entity set. One row is highlighted with a yellow background, showing `Parameter Name: count` and `Parameter Value: 123`.

Exp...	Level	Parameter Name	Parameter Value
1	-feed	Product_DetailSet	
2	-link - id	http://Six30.mydomain.com:	
2	-link - self	href="Product_DetailSet" re	
2	count	123	

To get the Count alone as a part of response and its at the end of Entity Set URI

The screenshot shows a ZAP test case interface. The URL bar contains `http://Six30.mydomain.com/Product_DetailSet/$count`. The main panel displays an **HTTP Response - Processing Time = 1058 ms**. The response headers show `~status_code: 200` and `~status_reason: OK`. Below the headers, a table lists parameters for the `Product_DetailSet` entity set. A single row is highlighted with a yellow background, showing `Parameter Name: count` and `Parameter Value: 123`.

Exp...	Level	Parameter Name	Parameter Value
2	count	123	

Handling Sorting and Searching

Query option to sort the entries in either Ascending or Descending order and then display the same as a part of response

Query Parameter => \$orderby

\$orderby=<property_name> <sort_order>

By default its ascending, but we need to explicitly specify desc if we want to sort in Descending Order.

IT_ORDER=> Parameter used for Sorting

To search for a given string along with the help of Wildcard characters

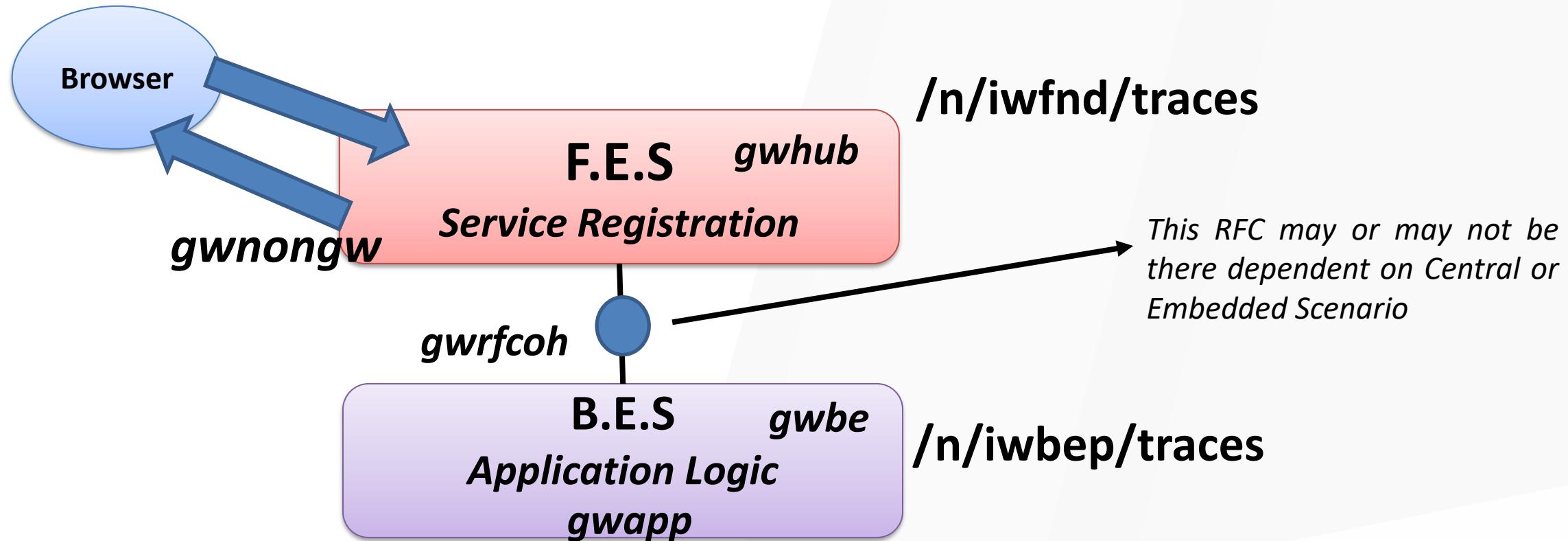
Query Parameter => search and not \$search

IV_SEARCH_STRING=> Parameter used for Search

A brief overview of the OData Queries and the Method Parameters required and if any Alternative way to get the same is available

Query Operation on URI	DPC_EXT Method Parameter	Alternative way available to get the Value	Coding required to implement the Query Operation
\$filter	IT_FILTER_SELECT_OPTIONS and IV_FILTER_STRING	Yes	Yes
\$select	No Method Parameter	Yes	No
\$orderby	IT_ORDER	Yes	Yes
search	IV_SEARCH_STRING	Yes	Yes
\$stop, \$skip	IS_PAGING	Yes	Yes
\$skiptoken, \$inlinecount	No such direct parameters are available as of now	Yes	Yes

Performance Analysis of OData Services



Total time taken by Framework(fw) => gwhub + gwrfcoh + gwbe

Total Time Taken(gwtotal) => fw + gwapp + gwnongw

Each layer will take time, so we need to analyze w.r.t Performance at each Step wr.t. both Frontend and Backend

Gateway Performance Trace w.r.t OData Service Calls

SAP Gateway: Performance Trace Summary

By using the Traces t-code

Average Times (in milliseconds)

No. of Requests	Processing Time	SAP GW Hub Sy...	RFC and Networ...	SAP GW Backen...	Application	Non-GW
1	951	24	0	11	916	0

x Headers Preview Response Initiator Timing Cookies

▼ Response Headers View source

cache-control: no-store, no-cache
content-encoding: gzip
content-length: 7802
content-type: application/json; charset=utf-8
dataserviceversion: 2.0
sap-metadata-last-modified: Fri, 02 Jul 2021 11:07:26 GMT
sap-perf-fesrec: 971827.000000
sap-processing-info: ODataBEP=,crp=,st=X,MedCacheHub=SHM,MedCacheBEP=SHM,codeployed=X,softstate=
sap-statistics: total=951,fw=35,app=916,gwtotal=951,gwfw=35,gwrfcoh=0,gwnongw=0,gwapp=916,gwhub=0,gwbe=35,icftotal=971,icfauth=0,icfext=966,icmtotal=976,icmreqrcv=0,icmext=97
6

All the time w.r.t sap-statistics are in milliseconds

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



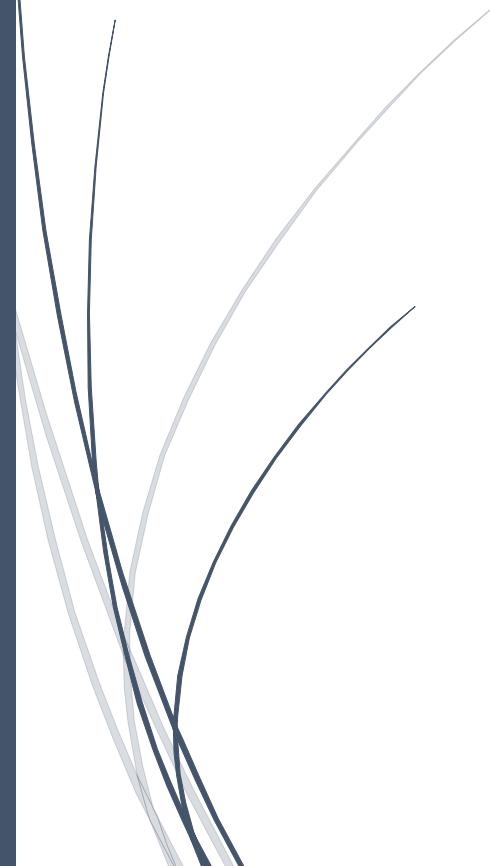
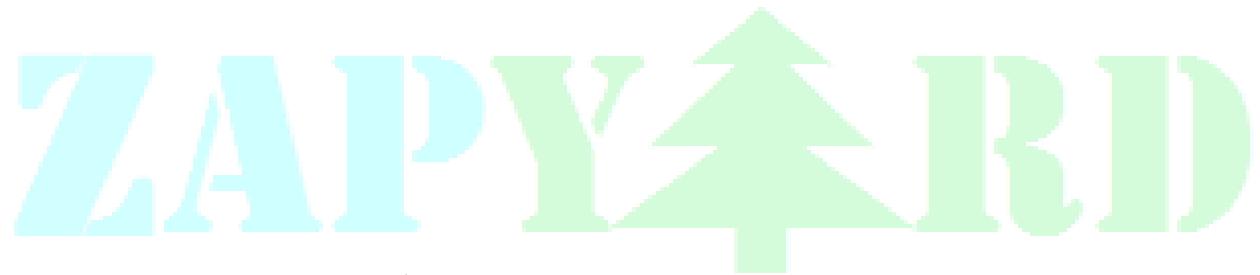
mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



Day 5

Filtering => Single as well as Multiple Parameter

Use OR

Use AND

If you don't pass the value for \$filter and if the URI is correct => Still you will get response.

Date handling in \$filter query: -

/sap/opu/odata/SAP/ZODATA_DEMOS_SRV/HeaderSet?\$filter=CreatedOn eq datetime'2021-04-03T00:00:00'

For a Range of Date: -

/sap/opu/odata/SAP/ZODATA_DEMOS_SRV/HeaderSet?\$filter=CreatedOn ge datetime'2021-04-01T00:00:00' and CreatedOn le datetime'2021-06-10T00:00:00'

Table IT_FILTER_SELECT_OPTIONS[1]-S...				
Attributes Standard [1x4(24)]				
Insert Column Columns ...				
Row	SIGN [C(1)]	OPTION [C(2)]	LOW [CString]	HIGH [CString]
I	BT		20210401	20210610

Code Snippet for Handling Date Filter: -

```
DATA:lr_date TYPE RANGE OF dats.  
      DATA(ls_date_filter) = VALUE #( it_filter_select_options[ 1 ]-  
select_options[ 1 ] OPTIONAL ).  
      lr_date = VALUE #( ( sign = ls_date_filter-sign option = ls_date_filter-  
option_low = ls_date_filter-low high = ls_date_filter-high ) ).  
  
      IF iv_filter_string IS NOT INITIAL.  
          SELECT FROM zheader FIELDS *  
*           WHERE (iv_filter_string)  
           WHERE created_on IN @lr_date  
           INTO CORRESPONDING FIELDS OF TABLE @et_entityset.  
      ELSE.  
          SELECT FROM zheader FIELDS *  
          INTO CORRESPONDING FIELDS OF TABLE @et_entityset.  
      ENDIF.
```

For \$select => No code implementation is required

URI for \$inlinecount=allpages =>

/sap/opu/odata/SAP/ZODATA_DEMOS_SRV/Product_DetailSet?\$filter=Product
Id eq 'HT-1000' or ProductId eq 'HT-
1002'&\$inlinecount=allpages&\$format=json

URI for \$skiptoken =>

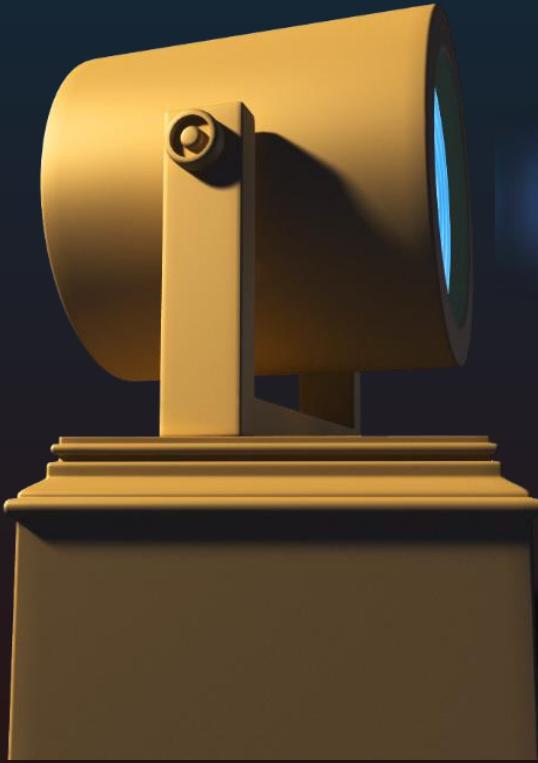
[http://103.44.1.51:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/Pro
duct_DetailSet?\\$filter=ProductId%20eq%20%27HT-
1000%27%20or%20ProductId%20eq%20%27HT-
1001%27%20or%20ProductId%20eq%20%27HT-
1002%27&\\$skiptoken=0&\\$format=json](http://103.44.1.51:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/Pro
duct_DetailSet?$filter=ProductId%20eq%20%27HT-
1000%27%20or%20ProductId%20eq%20%27HT-
1001%27%20or%20ProductId%20eq%20%27HT-
1002%27&$skiptoken=0&$format=json)

Code Snippet for \$inlinecount=allpages, \$top, \$skip and \$skiptoken along with \$filter

```
METHOD productdetailset_get_entityset.  
DATA:lt_headerdata  TYPE STANDARD TABLE OF bapi_epm_product_header,  
      lt_product_id  TYPE TABLE OF bapi_epm_product_id_range,  
      ls_product_id  TYPE bapi_epm_product_id_range,  
      lt_category    TYPE STANDARD TABLE OF bapi_epm_product_categ_range,  
      lv_max_rows    TYPE bapi_epm_max_rows,  
      lv_page_size   TYPE i VALUE 2,  
      lv_skiptoken   TYPE string,  
      lv_index_start  TYPE i,  
      lv_index_end   TYPE i,  
      lv_table_size  TYPE i.  
  
DATA(lv_top) = is_paging-top.  
DATA(lv_skip) = is_paging-skip.  
IF lv_top IS NOT INITIAL.  
  lv_max_rows-bapimaxrow = lv_top + lv_skip.  
ENDIF.  
"If $skiptoken is passed  
lv_skiptoken = io_tech_request_context->get_skiptoken().  
  
DATA(lt_filters) = io_tech_request_context->get_filter()->get_filter_select_options().  
  
READ TABLE it_filter_select_options INTO DATA(ls_filter) WITH KEY property  
y = 'ProductId'.  
IF sy-subrc EQ 0.  
  LOOP AT ls_filter-select_options INTO DATA(ls_so).  
    MOVE-CORRESPONDING ls_so TO ls_product_id.  
    APPEND ls_product_id TO lt_product_id.  
  ENDLOOP.  
ENDIF.  
  
IF line_exists( lt_filters[ property = 'CATEGORY' ] ).  
  DATA(ls_category) = lt_filters[ property = 'CATEGORY' ]-  
select_options[ 1 ].  
  lt_category = VALUE #( ( sign = ls_category-sign option = ls_category-  
option low = ls_category-low ) ).  
ENDIF.  
  
CALL FUNCTION 'BAPI_EPM_PRODUCT_GET_LIST'
```

Master SAP OData

```
EXPORTING
    max_rows          = lv_max_rows
TABLES
    headerdata        = lt_headerdata
    selparamproductid = lt_product_id
    *                 = SELPARAMSUPPLIERNAMES
    selparamcategories = lt_category.
    et_entityset      = CORRESPONDING #( lt_headerdata ).  
  
IF lv_skip IS NOT INITIAL.
    DELETE et_entityset TO lv_skip.
ENDIF.  
  
"Logic for SkipToken
IF lv_skiptoken IS NOT INITIAL.
    lv_table_size = lines( et_entityset ).
    IF lv_table_size IS INITIAL.
        CLEAR es_response_context-skiptoken.
    ELSEIF lv_table_size < lv_page_size.
        "We will pass all data
    ELSE.
        lv_index_start = lv_skiptoken.
        lv_index_end   = lv_page_size + lv_skiptoken.
        DATA(lt_entityset) = et_entityset.
        CLEAR et_entityset.
        LOOP AT lt_entityset INTO DATA(ls_entity).
            IF sy-tabix > lv_index_start AND sy-tabix <= lv_index_end.
                APPEND ls_entity TO et_entityset.
            ENDIF.
        ENDLOOP.
        IF lv_index_end >= lv_table_size.
            CLEAR es_response_context-skiptoken.
        ELSE.
            es_response_context-skiptoken = lv_index_end.
            CONDENSE es_response_context-skiptoken.
        ENDIF.
    ENDIF.
ENDIF.  
  
"Logic for $inlinecount=allpages
IF io_tech_request_context->has_inlinecount( ) = abap_true.
    es_response_context-inlinecount = lines( et_entityset ).
ENDIF.
ENDMETHOD.
```



Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

www.ZAPYard.com

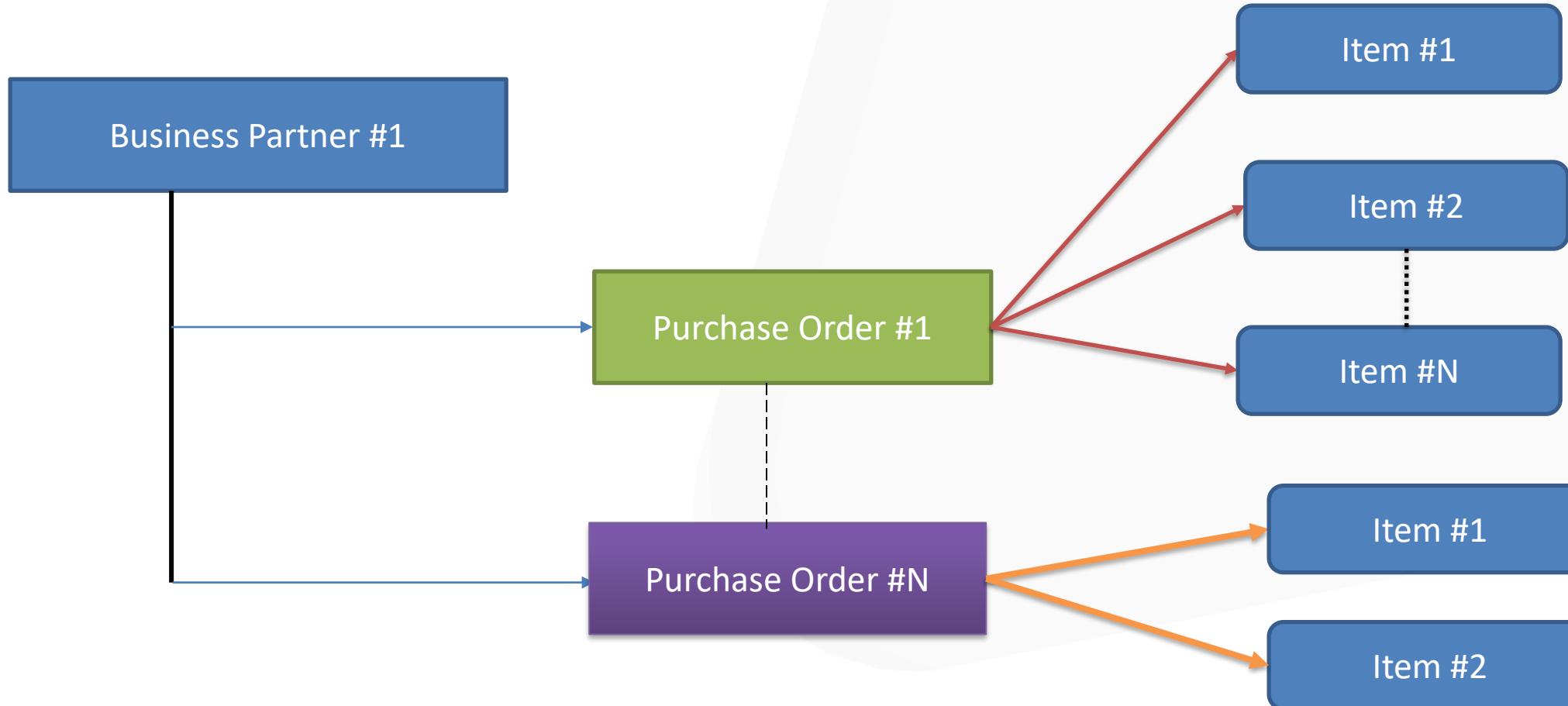
ZAPYARD

Agenda

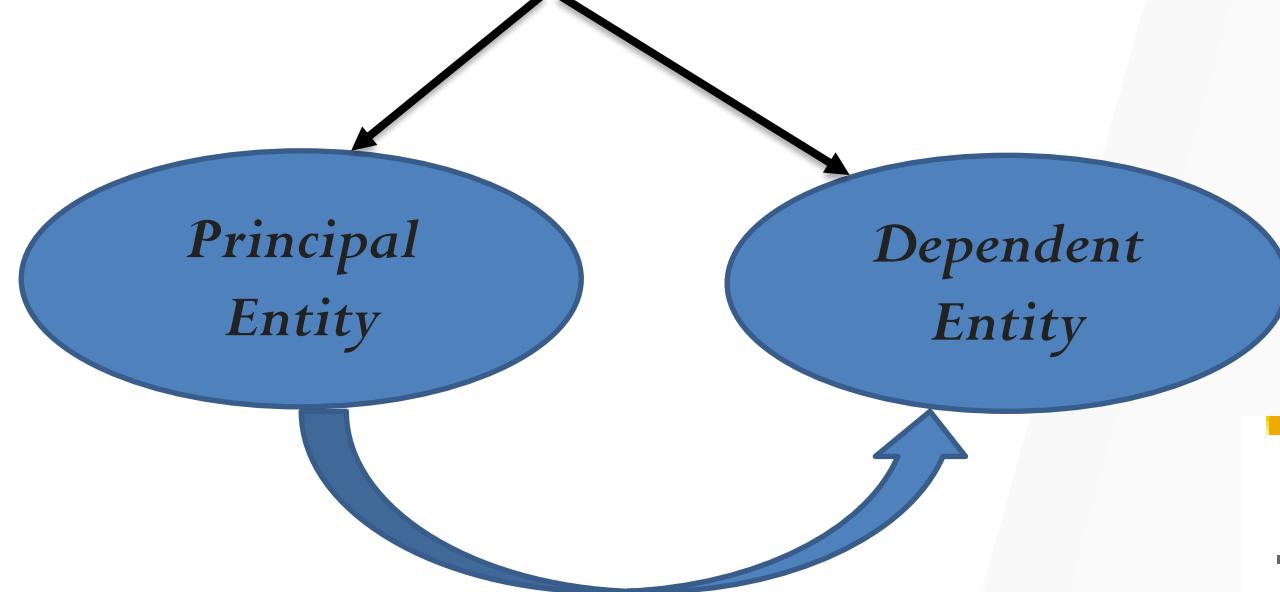
- Understanding =>
Association, Navigation and Cardinality
- Creation of Association, Navigation and usage of
Same
- Implementing Expand Operations by using
\$expand(Framework Driven)



Understanding Association, Navigation

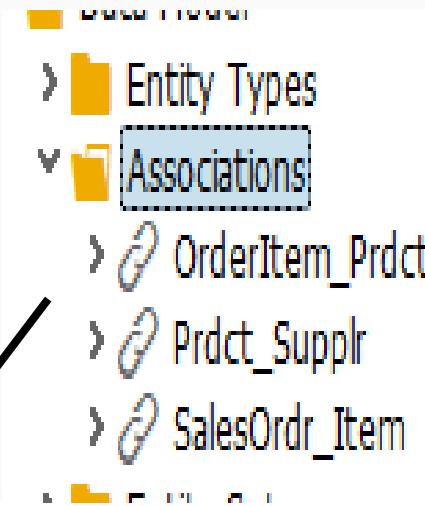


Implementing Association, Navigation and Cardinality



- Navigation Property

It helps with the occurrence of number of records



Multiple Associations may be there in a Gateway Service

Implementing Referential Constraints and Navigation Property



This basically plays a crucial role in setting up the association between principal and dependent entity.

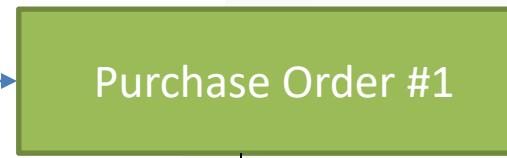
With every Association we need to setup a **Referential Constraint** so that the relationship between the Principal and Dependent entity is setup. Will the Association/Navigation fail if we later on delete this Referential Constraint?

Understanding Association, Navigation and \$Expand with an use case

Business PartnerSet-> Get_Entity



PurchaseOrderHeaderSet -> Get_EntitySet

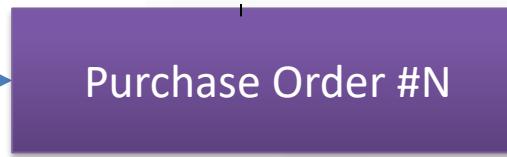


Item #1

Item #2

Item #N

PurchaseOrderItemSet -> Get_EntitySet



Item #1

Item #2

Difference between Association/Navigation and \$expand

Association/Navigation	\$expand
Provides the associated (dependent) entity/entities only by using the Navigation property	Provides the associated (dependent) entity/entities + Principal entity/entities using Navigation property

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



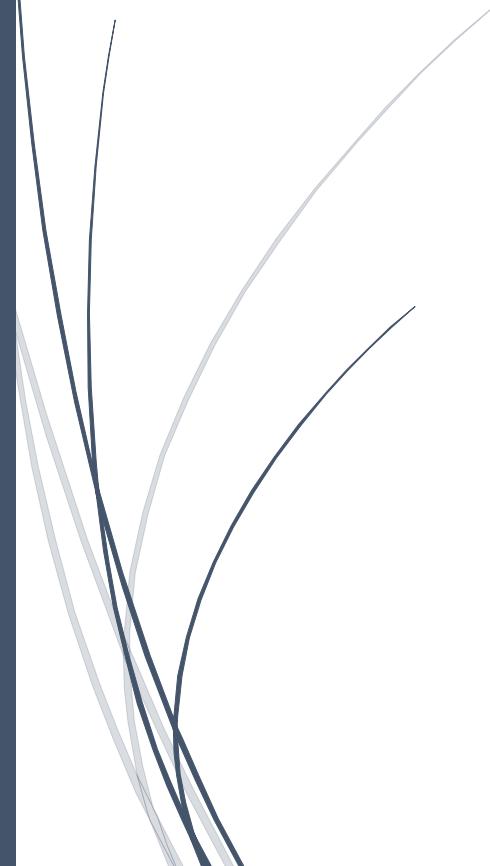
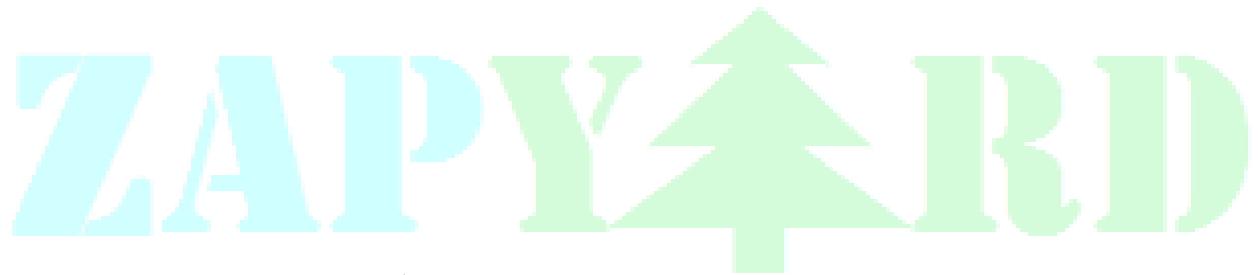
mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



Day 6

So, for basic filtering, paging and sorting we can use the below class=>method

```
CALL METHOD /iwbep/cl_mgw_data_util->paging  
  EXPORTING  
    is_paging = is_paging  
  CHANGING  
    ct_data   = et_entityset.
```

For dynamic sorting either we develop our own logic or we use the sorting method of the above class.

For the below URI there is a problem as & is used as a filter property value which needs to be replaced by %26.

/sap/opu/odata/sap/ZODATA_DEMOS_O97_SRV/ProductSet?\$filter=Category eq 'PDAs & Organizers'

Proper URI:-

[http://103.44.1.51:8770/sap/opu/odata/sap/ZODATA DEMOS O97 SRV/Product DetailSet?\\$filter=Category%20eq%20%27PDAs%20%26%20Organizers%27](http://103.44.1.51:8770/sap/opu/odata/sap/ZODATA_DEMOS_O97_SRV/Product_DetailSet?$filter=Category%20eq%20%27PDAs%20%26%20Organizers%27)

Master SAP OData

For the below URI:-

```
/sap/opu/odata/sap/ZODATA_DEMOS_SRV/HeaderSet?$filter=Id eq  
'100' and CreatedOn ge datetime'2021-04-01T00:00:00' and CreatedOn  
le datetime'2021-05-10T00:00:00'
```

THE ISSUE IS IT_FILTER_SELECT_OPTIONS WILL NOT GET FILLED AND WE NEED TO HANDLE THIS SCENARIO

USAGE OF SUBSTRING: -

/sap/opu/odata/sap/ZODATA_DEMOS_SRV/HeaderSet?\$filter=substringof('J*',Name)

Call Number 1 :- Sales Order Get_Entity

Call Number 2:- Sales Order Item =>

Once you call the Navigation Property as a part of URI:-

Master SAP OData

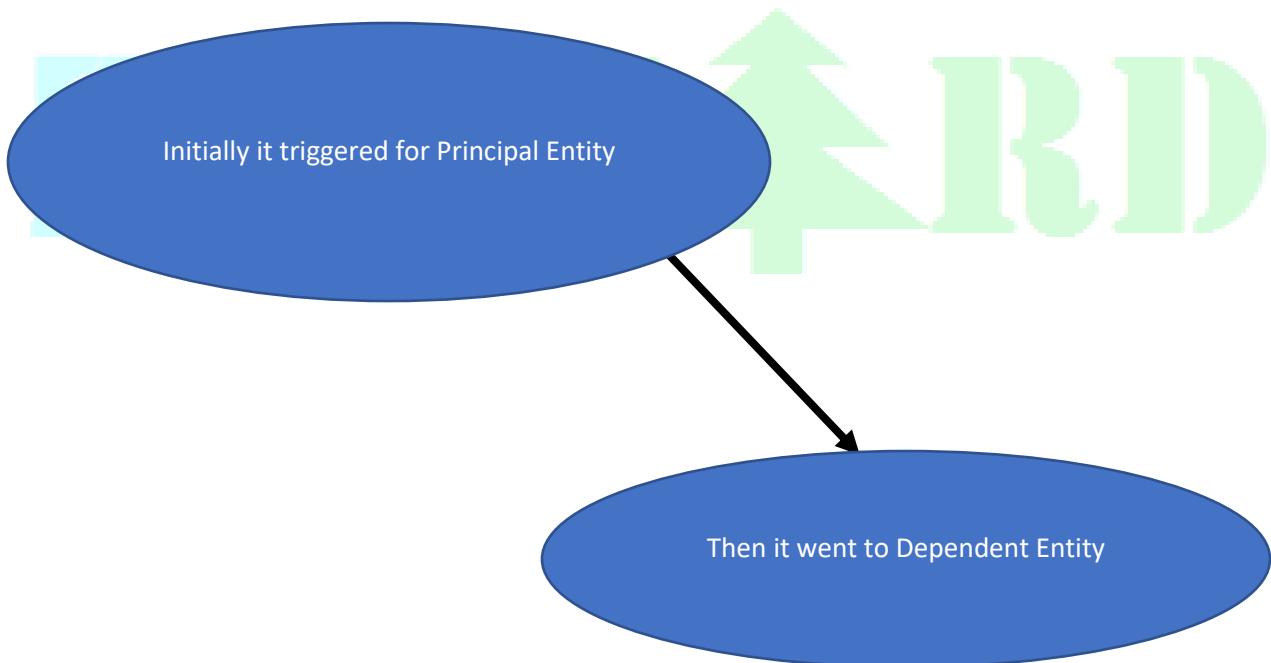
URI for Basic Association/Navigation:-

```
/sap/opu/odata/sap/ZODATA_DEMOS_SRV/SalesOrderSet('500000000')  
/Order_To_Item?$format=json
```

This will give us only the Line Items

URI for \$expand:-

```
/sap/opu/odata/sap/ZODATA_DEMOS_SRV/SalesOrderSet('500000000')  
?$expand=Order_To_Item&$format=json
```



Logic for the Same: -

```
DATA:lv_so_id      TYPE bapi_epm_so_id,
      lt_order_items TYPE TABLE OF bapi_epm_so_item.

lv_so_id = VALUE #( it_key_tab[ name = 'SoId' ]-value OPTIONAL ).

CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
  EXPORTING
    input = lv_so_id
  IMPORTING
    output = lv_so_id.

CALL FUNCTION 'BAPI_EPM_SO_GET_DETAIL'
  EXPORTING
    so_id      = lv_so_id
  TABLES
    itemdata = lt_order_items
  RETURN =
*.

IF lt_order_items IS NOT INITIAL.
  et_entityset = CORRESPONDING #( lt_order_items ).
ENDIF.
```

/sap/opu/odata/sap/ZODATA_DEMOS_SRV/SalesOrderSet('500000000')
/Order_To_Item(Sold='500000000',SoltemPos='10')?\$format=json

Navigation Keys as a part of Navigation Path

Logic for the Same: -

```
DATA:lv_so_id      TYPE bapi_epm_so_id,
      lv_soitempos TYPE snwd_so_item_pos,
      lt_ordr_itms TYPE TABLE OF bapi_epm_so_item.

      lv_so_id = VALUE #( it_key_tab[ name = 'SoId' ]-value OPTIONAL ).
      IF it_navigation_path IS NOT INITIAL.
         READ TABLE io_tech_request_context-
>get_navigation_path( ) INTO DATA(ls_nvgtn) INDEX 1.
         CASE ls_nvgtn-nav_prop.
            WHEN 'ORDER_TO_ITEM'.
               lv_soitempos = VALUE #( ls_nvgtn-key_tab[ name = 'SO_ITEM_POS' ]-value OPTIONAL ).
            WHEN OTHERS.
         ENDCASE.
      ENDIF.
      CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
      input   = lv_so_id
      IMPORTING
      output  = lv_so_id.

      CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
      input   = lv_soitempos
      IMPORTING
      output  = lv_soitempos.

      CALL FUNCTION 'BAPI_EPM_SO_GET_DETAIL'
      EXPORTING
      so_id    = lv_so_id
      TABLES
      itemdata = lt_ordr_itms
      RETURN   =
      .

      IF lt_ordr_itms IS NOT INITIAL AND lv_soitempos IS NOT INITIAL.
         er_entity = CORRESPONDING #( VALUE #( lt_ordr_itms[ so_id = lv_so_id so_item_pos = lv_soitempos ] OPTIONAL ) ).
      ENDIF.
```



Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

www.ZAPYard.com

ZAPYARD

Agenda

- Implementing Expand Operations by using \$expand using Data Provider Expand
- Usage of get_expanded_entity/entityset
- Deep Insert using create_deep_entity
- Function Imports



Data Provider Expand Comparison with Framework Expand

Framework Expand	Data Provider Expand
Requires no implementation as such	Requires Implementation
As this is handled by framework, same logic may be called multiple times or unnecessarily resulting in poor performance	In some cases, this provides better performance depending on how the code is implemented in the DPC_EXT Class

Implementing Get_Expanded_Entity/EntitySet

*If we have just redefined the Method without any logic implemented
then we will get blank response with Status as 200.*

The screenshot shows the SAP OData Test Client interface. At the top, the HTTP Method is set to GET, Request URI is /sap/opu/odata/sap/ZLEARN_ODATA_SRV/SalesOrderSet?\$expand=Order_To_Item&\$format=json, and Protocol is set to HTTP. The Test Group and Test Case tabs are visible. Below the header, there are several icons for file operations like Add File, Remove File, and Data Explorer. The main area is divided into two sections: 'HTTP Request' on the left and 'HTTP Response' on the right. The 'HTTP Response' section displays the following details:

Header Name	Value
~status_code	200
~status_reason	OK

Below the header table, the JSON response body is shown:

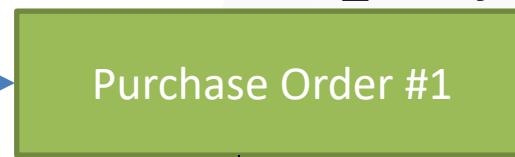
```
1: {  
2:   "d": {  
3:     "results": [  
4:       ]  
5:   }  
6: }  
7: }
```

Implementing Association, Navigation and \$Expand – Scenario 1

Business PartnerSet-> Get_Entity



PurchaseOrderHeaderSet -> Get_EntitySet

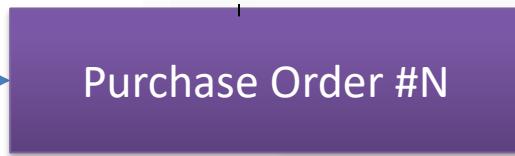


Item #1

Item #2

Item #N

PurchaseOrderItemSet -> Get_EntitySet



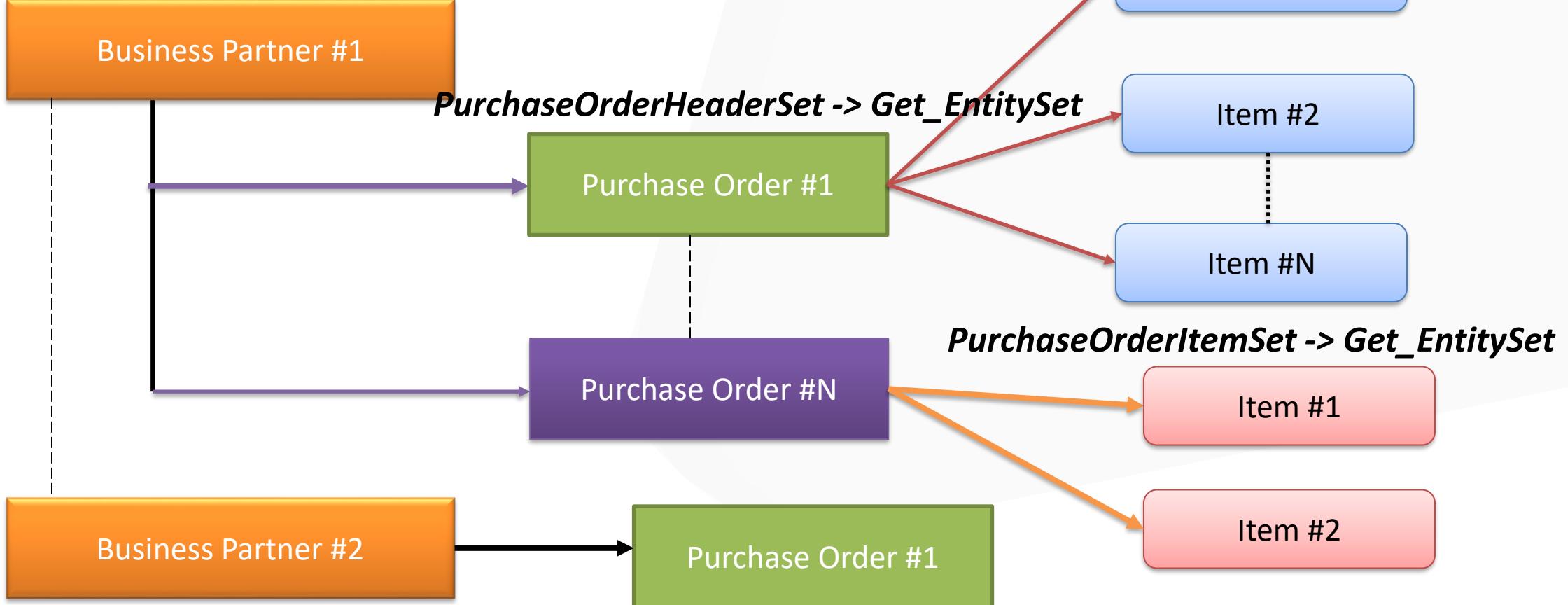
Item #1

Item #2

\$expand must be implemented in the Method /IWBEPIF_MGW_APPL_SRV_RUNTIME~GET_EXPANDED_ENTITY of the DPC_EXT Class

Implementing Association, Navigation and \$Expand – Scenario 2

Business PartnerSet-> Get_EntitySet

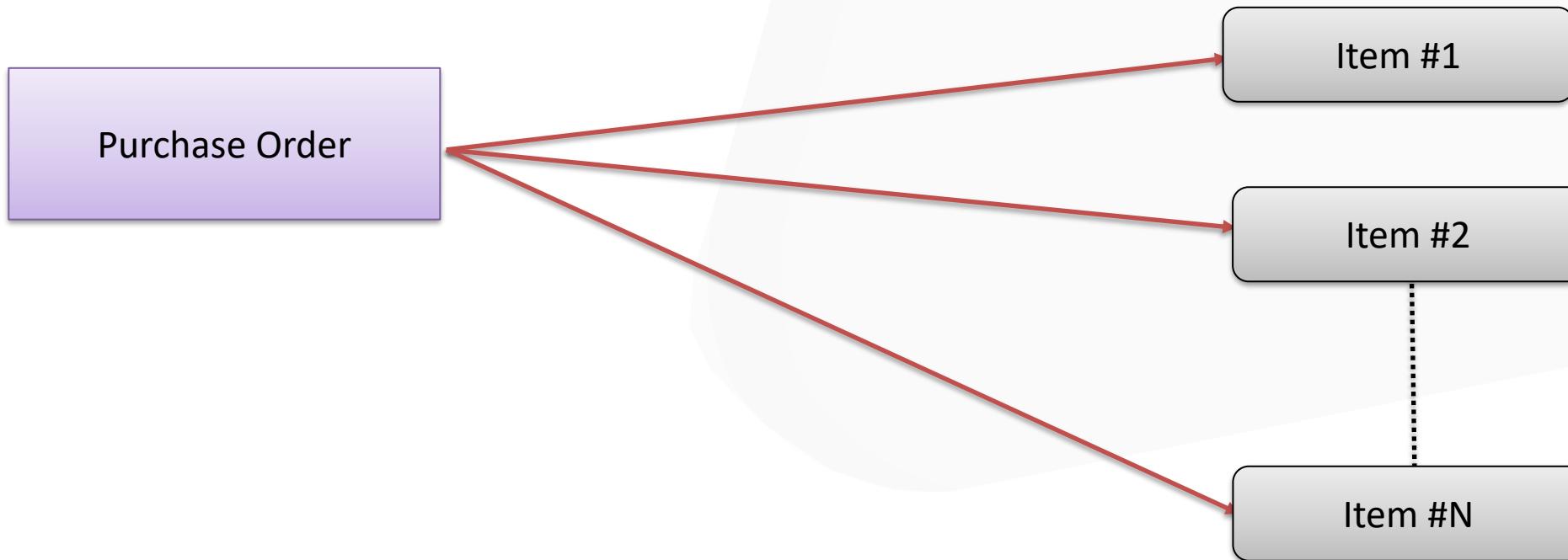


\$expand must be implemented in the Method /IWBEPIF_MGW_APPL_SRV_RUNTIME~GET_EXPANDED_ENTITYSET of the DPC_EXT Class

Implementing Deep Insert Operations

SAP Gateway Deep Insert allows to store hierarchical data like if we take an use case of Purchase Order that has both header and item data (one can not exist without another).

Deep Insert is opposite to \$expand URI option



Implementing Deep Insert Operations via Create_deep_entity

Here the request body will be based on the result of a \$expand call which can be used for HTTP POST and the Method used is

/IWBEPIF_MGW_APPL_SRV_RUNTIME~CREATE_DEEP_ENTITY of DPC_EXT Class

Request URI: sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesOrderSet

Protocol: HTTP

Test Group: Test Case

HTTP Request:

```
1 {
2   "d" : {
3     "__metadata" : {
4       "id" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesOrder('5000000020')",
5       "uri" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesOrder('5000000020')",
6       "type" : "ZGW_SALES_ORDER_SRV.SalesOrder"
7     },
8     "BuyerId" : "100000000",
9     "Order_Item" : {
10       "results" : [
11         {
12           "__metadata" : {
13             "id" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesOrder('5000000020')/Order_Item('10')",
14             "uri" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesOrder('5000000020')/Order_Item('10')",
15             "type" : "ZGW_SALES_ORDER_SRV.Order_Item"
16           },
17           "SoId" : "5000000000",
18           "SoItemPos" : "10",
19           "ProductId" : "HT-1000",
20         }
21       ]
22     }
23   }
24 }
```

HTTP Response - Processing Time = 260428 ms

Header Name	Value
~status_code	201
~status_reason	Created

Response in Browser

Error Log

HTTP Header

Use as Request

Data Explorer

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

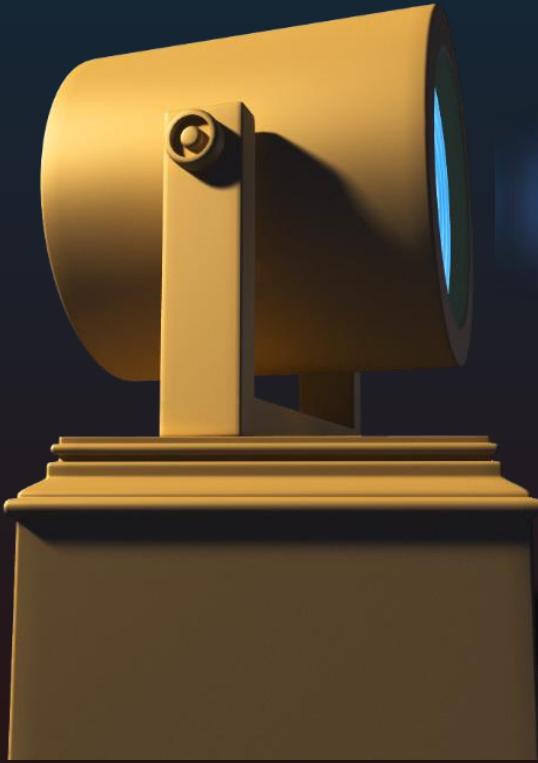
ZAPYARD



mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD



Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

www.ZAPYard.com

ZAPYARD

Agenda

- Understanding Function Imports
- Handling Function Import Parameters
- Implement Function Imports
- Understanding Media Handling in OData
- Upload and Access Media via SAP Gateway

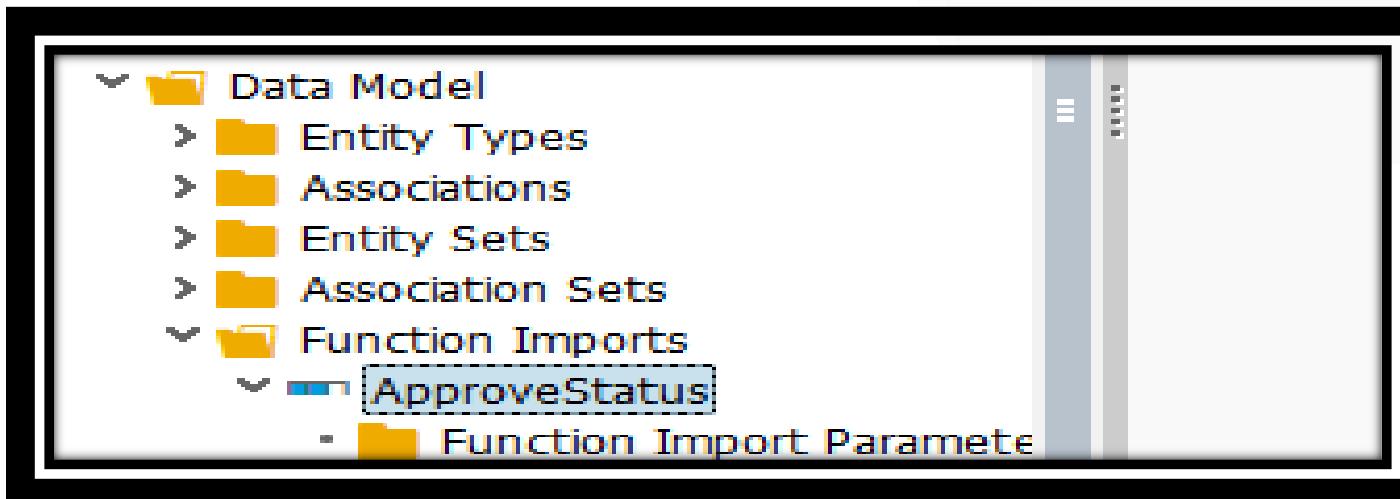


Understanding Function Imports

Function imports are used to perform an action on a business object that can't be executed by using CRUD-Q methods.

E.g.:- Release of a Sales Order, Confirmation of an Order based on Availability Check, To Update the Status

Function imports are defined on the service level not entity set level. As a result, we need to use the name of the function import in the service implementation .

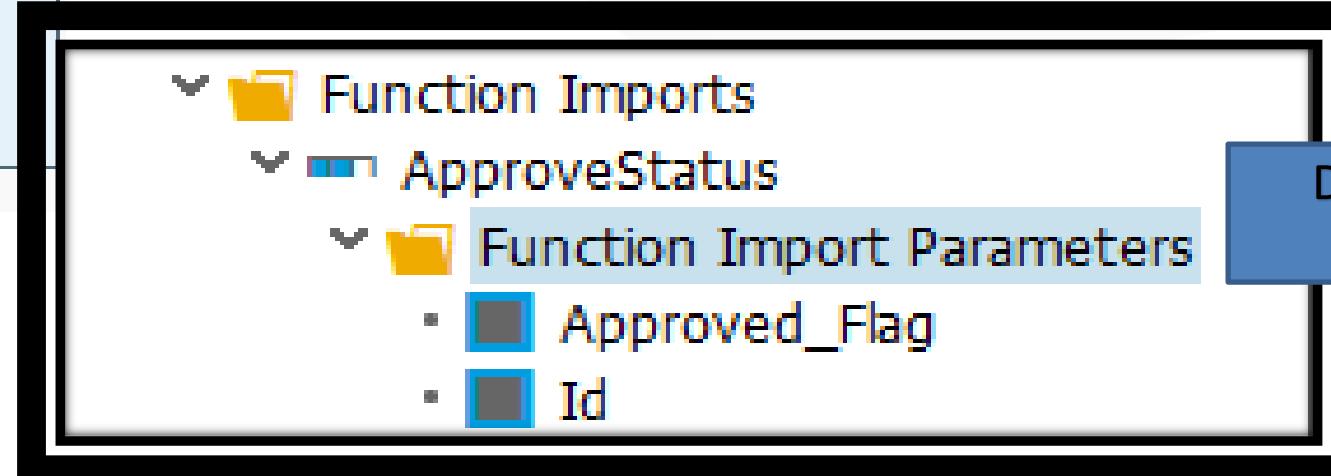


Understanding Function Imports and enabling Function Import Parameters

Function Imports							
Name	Return Type Kind	Return Type	Return Cardinality	Return Entity Set	HTTP Method Type	Action for Entity Type	Label
ApproveStatus	No return				Not specified		

Return Cardinality
1
0..1
1
0..n
1..n

Only GET or POST is allowed



Implementing Function Imports

To implement function import operation. Go to DPC_EXT class and redefine method

/IWBEPMGWAAPLSRVRUNTIME~EXECUTE_ACTION

Also, remember based on the HTTP Method Type Set as POST/GET during Function Import Creation =>

We need to accordingly set the HTTP Method Call w.r.t Request URI otherwise we will get an Error as shown below.

The screenshot shows the SAP Gateway Client interface. In the top navigation bar, the 'Service Administration' tab is selected. Below it, the 'HTTP Method' dropdown is set to 'GET'. The 'Request URI' field contains the URL: '/sap/opu/odata/sap/ZLEARN_ODATA_SRV/ApproveStatus?Id='101'&Approved_Flag='X''. The 'Protocol' dropdown is set to 'HTTP'. On the right side of the screen, the 'HTTP Response' panel displays the following information:

Header Name	Value
~status_code	405
~status_reason	Method Not Allowed

Below the response table, the raw XML error message is displayed:

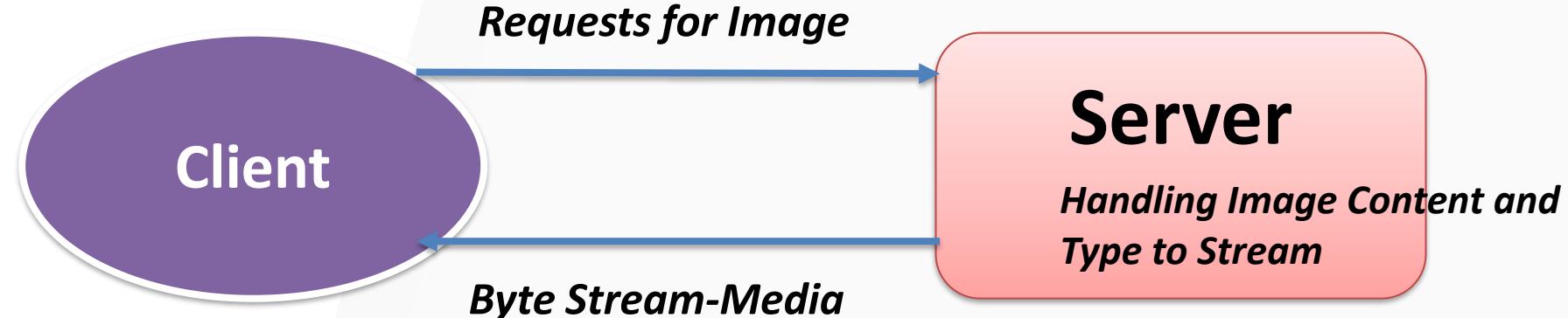
```
<?xml version="1.0" encoding="UTF-8"?>
- <error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <code>005056A509B11ED199D8826D151FC0FE</code>
  <message xml:lang="en">The specified HTTP method is not allowed for the resource identified by the Data Service Request URI</message>
  - <innererror xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
    <transactionid>7CC752319E2A0040E0060E51B66DE60E</transactionid>
    <timestamp>20210721181508.6224290</timestamp>
    - <Error_Resolution>
      <SAP_Transaction>For backend administrators: run transaction /IWFND/ERROR_LOG on SAP Gateway hub system and search for entries with the timestamp above for more details</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>
```

Implementing Function Imports – Important Points

- *Always remember that if the functionality can be achieved by CRUD-Q Methods then don't go for additional Implementation of Function Imports. Only go for the same in those scenarios where you feel that Function Imports will be the best fit case for the problem.*
- *We can have N number of function imports for a Service but the logic for all will be handled in a Single Method **/IWBEPM/IF_MGW_APPL_SRV_RUNTIME~EXECUTE_ACTION** of DPC_EXT Class.*
- *Generic Syntax of Calling Function Imports are:-*
[/sap/opu/odata/sap/<Odata_Service_Name>/<Fn_Import_Name>?<Fn_Import_Param>=Value](https://sap/opu/odata/sap/<Odata_Service_Name>/<Fn_Import_Name>?<Fn_Import_Param>=Value)

Media Handling in OData via Media Links

- Pictures and Images
- PDF Documents
- Office Documents
- Video

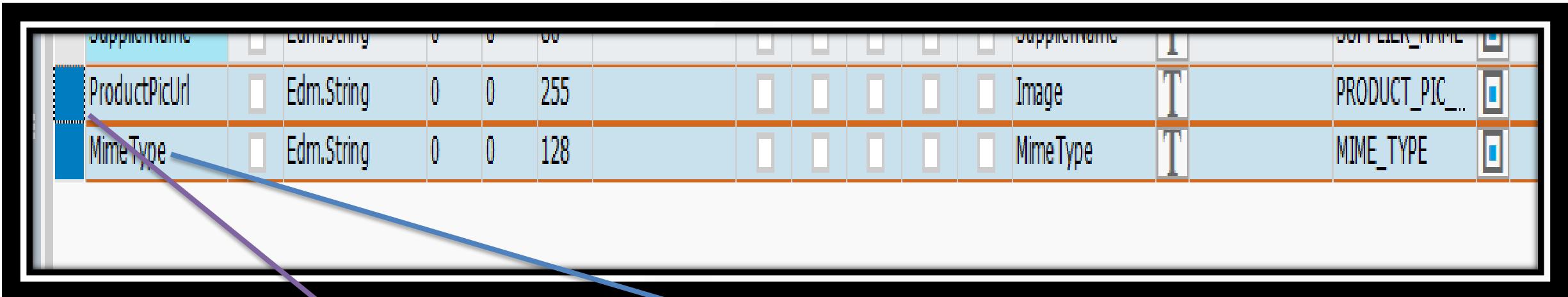


For handling Media w.r.t any Entity, the checkbox for Media must be enabled so that it can indicate the Entity as Media Supportive.

Entity Types									
	Name	ABAP Structure	Base Type	Abstr...	Label	La...	Semantics	Thing	Media
	Header	ZHEADER				T			
	Message_Ret...					T			
	Product					T			<input checked="" type="checkbox"/>
	Product_Detail					T			

Mapping with Media Content Source and Media Content Type

For our Media Entity to work, we need to make certain tweaks to our Model as in the Metadata which will be done at MPC_EXT Class-DEFINE Method.



Supplements	Comments	0	0	00	Supplements	T	JOURNAL_NAME
ProductPicUrl	Edm.String	0	0	255	Image	T	PRODUCT_PIC_
Mime Type	Edm.String	0	0	128	MimeType	T	MIME_TYPE

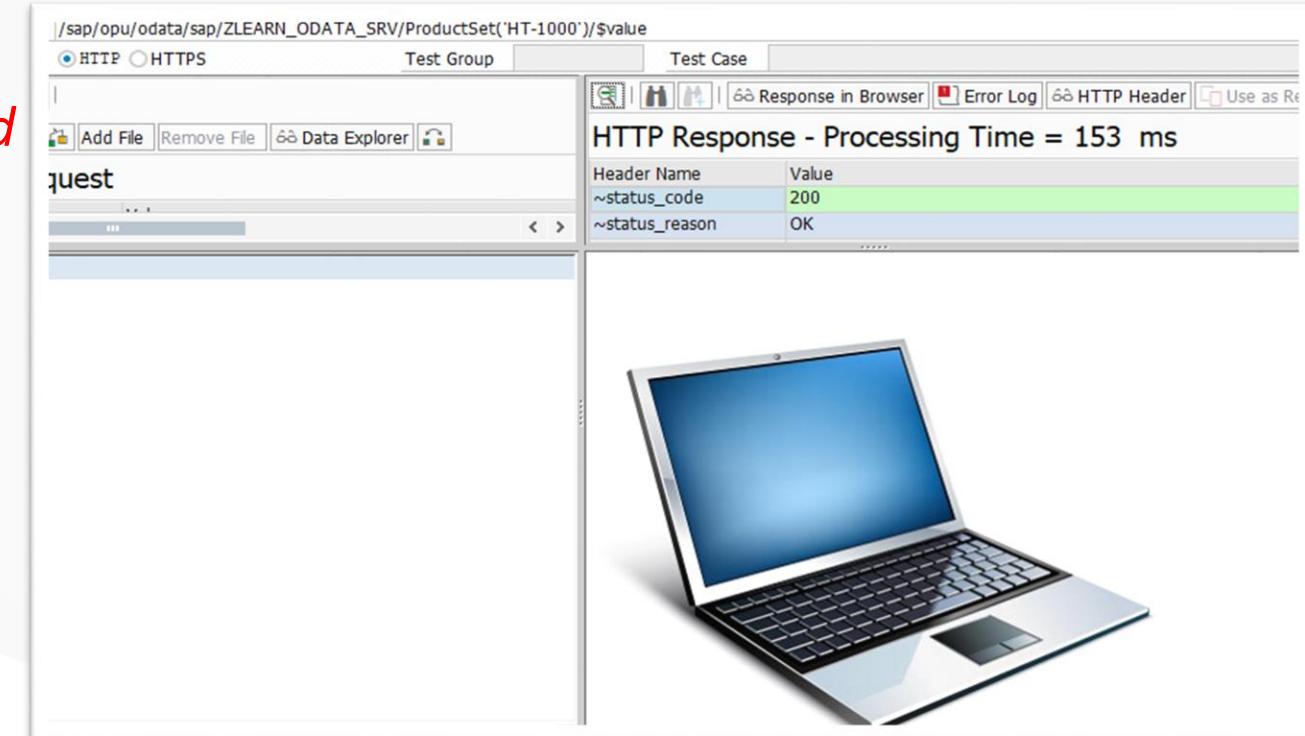
Basically we need to set the Media Content Source and Media Content Type by mapping with the Respective entity properties as in Media URL and Type.

Accessing the Media Links

For accessing the Media Links i.e. Media by Streams via Gateway we need to make use of \$value option.

The following methods need to be implemented as per the Use Case.

- ❖ ~**GET_STREAM**
- ❖ ~**CREATE_STREAM=> Key property not required**
- ❖ ~**UPDATE_STREAM**
- ❖ ~**DELETE_STREAM**



The Stream Parameter which needs to be filled will be of below type:-

/iwbep/if_mgw_core_srv_runtime=>try_s_media_resource

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



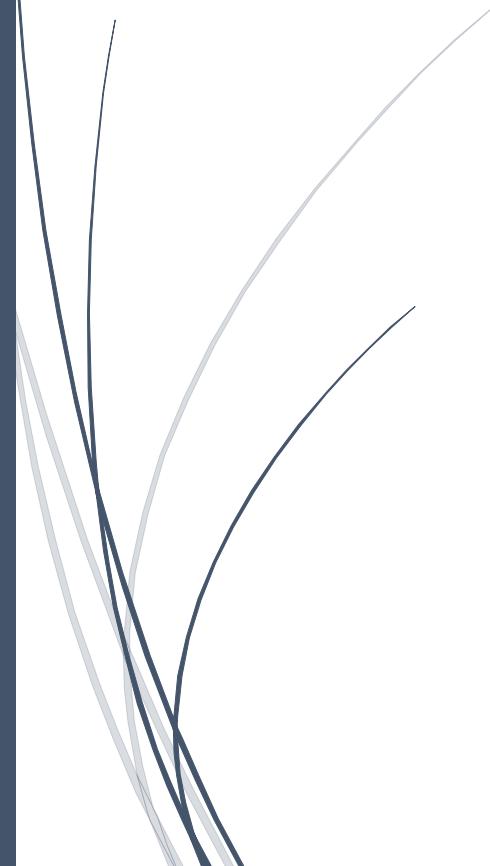
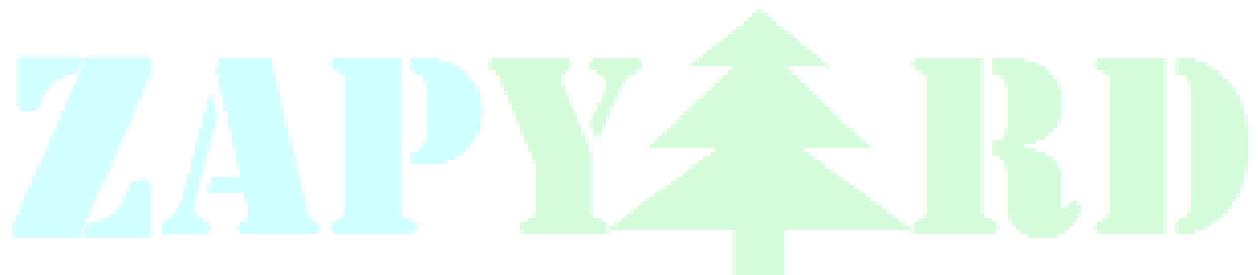
mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



Master SAP OData

Day 8

- While implementing Fn Import, if cardinality is set to 0..n or 1..n a return entity set has to mapped and selected.
- Once the Fn Import is defined, You need to define the Fn Import Parameters.

Once The Fn Import and its Parameters are defined=> We need to regenerate the runtime artifacts.

We can have multiple Fn Imports and the logic for all will be implemented in the same EXECUTE_ACTION only.

We can check the FI Impl. in Metadata Call

The screenshot shows the SAP OData Test Client interface. The top navigation bar includes 'HTTP Method' (GET selected), 'Request URI' (sap/opu/odata/sap/ZODATA_DEMOS_SRV/\$metadata), 'Protocol' (HTTP selected), and tabs for 'Test Group' and 'Test Case'. Below the navigation is a toolbar with icons for file operations like 'Add File' and 'Remove File'. The main area is divided into two panes: 'HTTP Request' on the left and 'HTTP Response' on the right. The 'HTTP Response' pane title is 'HTTP Response - Processing Time = 222 ms'. It displays three header rows: 'Header Name' (Value), 'status_code' (200), and 'status_reason' (OK). The 'HTTP Request' pane shows a single row with the value '1'. The 'HTTP Response' pane also contains XML metadata code. A yellow highlight is applied to several specific parts of the XML code, including 'FunctionImport Name="Appr_Emp"', 'EntitySet="Message_ReturnCollection"', 'for="ZODATA_DEMOS_SRV.Header"', 'm:HttpMethod="POST"', 'Return-Type="ZODATA_DEMOS_SRV.Message_Return"', 'Parameter Name="Id" MaxLength="3" Type="Edm.String" Mode="In"', and 'Parameter Name="Approved_Flag" MaxLength="1" Type="Edm.String" Mode="In"'.

```
<EntityType="ZODATA_DEMOS_SRV.SalesOrderItem" sap:deletable="false"/>
<EntitySet sap:content-version="1" Name="Message_ReturnCollection"
  EntityType="ZODATA_DEMOS_SRV.Message_Return"/>
- <AssociationSet sap:content-version="1" Name="Order_ItemSet" sap:updatable="false"
  sap:createable="false" sap:deletable="false" Association="ZODATA_DEMOS_SRV.Order_Item">
  <End Role="FromRole_Order_Item" EntitySet="SalesOrderSet"/>
  <End Role="ToRole_Order_Item" EntitySet="SalesOrderItemSet"/>
</AssociationSet>
- <FunctionImport Name="Appr_Emp" EntitySet="Message_ReturnCollection" sap:action-
  for="ZODATA_DEMOS_SRV.Header" m:HttpMethod="POST"
  Return-Type="ZODATA_DEMOS_SRV.Message_Return">
  <Parameter Name="Id" MaxLength="3" Type="Edm.String" Mode="In"/>
  <Parameter Name="Approved_Flag" MaxLength="1" Type="Edm.String" Mode="In"/>
</FunctionImport>
</EntityContainer>
<atom:link xmlns:atom="http://www.w3.org/2005/Atom"
  href="http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/
  $metadata" rel="self"/>
<atom:link xmlns:atom="http://www.w3.org/2005/Atom"
  href="http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/
  $metadata" rel="latest-version"/>
```

Logic:-

```
METHOD /iwbep/if_mgw_appl_srv_runtime~execute_action.  
DATA:lt_parameter TYPE /iwbep/t_mgw_name_value_pair,  
      ls_entity      TYPE zcl_zodata_demos_mpc_ext=>ts_message_return.  
DATA(lv_fn_imp) = io_tech_request_context->get_function_import_name( ).  
lt_parameter = io_tech_request_context->get_parameters( ).  
DATA(lv_id) = VALUE #( lt_parameter[ name = 'ID' ]-value OPTIONAL ).  
DATA(lv_flag) = VALUE #( lt_parameter[ name = 'APPROVED' ]-  
      value OPTIONAL ).  
CASE lv_fn_imp.  
  WHEN 'Appr_Emp'.  
    UPDATE zheader SET approved = lv_flag WHERE id = lv_id.  
    IF sy-subrc EQ 0.  
      ls_entity-type = 'S'.  
      ls_entity-text = 'Employee Info Successfully Approved'.  
    ELSE.  
      ls_entity-type = 'E'.  
      ls_entity-text = 'Error in Approval'.  
    ENDIF.  
    copy_data_to_ref(  
      EXPORTING  
        is_data = ls_entity  
      CHANGING  
        cr_data = er_data  
    ).  
  WHEN OTHERS.  
ENDCASE.  
ENDMETHOD.
```

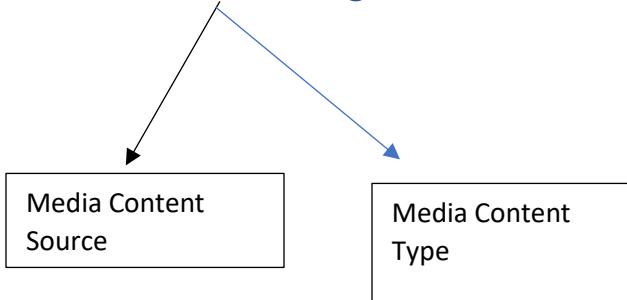
While Trying to use Fn Import => Make sure you pass all Parameters which are defined as Fn Import Parameters for the Service otherwise the URI Call will throw error.

URI:-

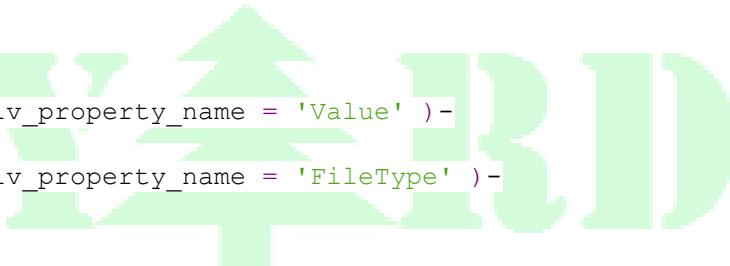
/sap/opu/odata/sap/ZODATA_DEMOS_SRV/Appr_Emp?Id='100'&Approved_Flag='X'

Master SAP OData

Media Handling =>



```
METHOD define.  
  DATA:lo_entity TYPE REF TO /iwbepl/if_mgw_odata_entity_typ.  
  super->define( ).  
  lo_entity = model->get_entity_type( iv_entity_name = 'File' ).  
  lo_entity->set_is_media(  
    iv_is_media = abap_true  
  ).  
  
  IF lo_entity IS BOUND.  
    lo_entity->get_property( iv_property_name = 'Value' )-  
    >set_as_content_source( ).  
    lo_entity->get_property( iv_property_name = 'FileType' )-  
    >set_as_content_type( ).  
  ENDIF.  
ENDMETHOD.
```



If we check the Metadata Call:-

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

- Method:** GET
- URI:** /sap/opu/odata/sap/ZODATA_DEMOS_SRV/\$metadata
- HTTP Request:** TTP Request
- HTTP Response - Processing Time = 692 ms:**
 - Header Name: Value
~status_code: 200
~status_reason: OK
 - Content:

```
<Property Name="Quantity" sap:unicode="false" Nullable="false" Type="Edm.Decimal"  
Precision="13" sap:createable="false" Scale="3"/>  
<Property Name="QuantityUnit" sap:filterable="false" sap:sortable="false" sap:updatable="false"  
sap:label="Unit of Measure" sap:unicode="false" MaxLength="3" Nullable="false"  
Type="Edm.String" sap:createable="false" sap:semantics="unit-of-measure"/>  
</EntityType>  
- <EntityType sap:content-version="1" Name="File" m:HasStream="true">  
- <Key>  
  <PropertyRef Name="Filename"/>  
<Key>  
  <Property Name="Filename" sap:filterable="false" sap:sortable="false" sap:updatable="false"  
  sap:label="Name of the File" sap:unicode="false" MaxLength="200" Nullable="false"  
  Type="Edm.String" sap:createable="false"/>  
  <Property Name="Value" sap:filterable="false" sap:sortable="false" sap:updatable="false"  
  sap:label="File Content" sap:unicode="false" Nullable="false" Type="Edm.Binary"  
  sap:createable="false"/>  
  <Property Name="FileType" sap:filterable="false" sap:sortable="false" sap:updatable="false"  
  sap:label="Type of File" sap:unicode="false" MaxLength="100" Nullable="false"/>
```

Master SAP OData

We need to implement the below Logic w.r.t Create_Stream.

```
/IWBEP/IF_MGW_APPL_SRV_RUNTIME-CREATE_STREAM active
METHOD /iwbep/if_mgw_appl_srv_runtime-create_stream.
  DATA:ls_file TYPE zfileupload.
  ls_file = VALUE #( filename = iv_slug value = is_media_resource-value file_type = is_media_resource-mime_type ).
  INSERT INTO zfileupload VALUES ls_file.
  IF sy-subrc EQ 0.
    copy_data_to_ref(
      EXPORTING
        is_data = ls_file
      CHANGING
        cr_data = er_entity
    ).
  ELSE.
  ENDIF.

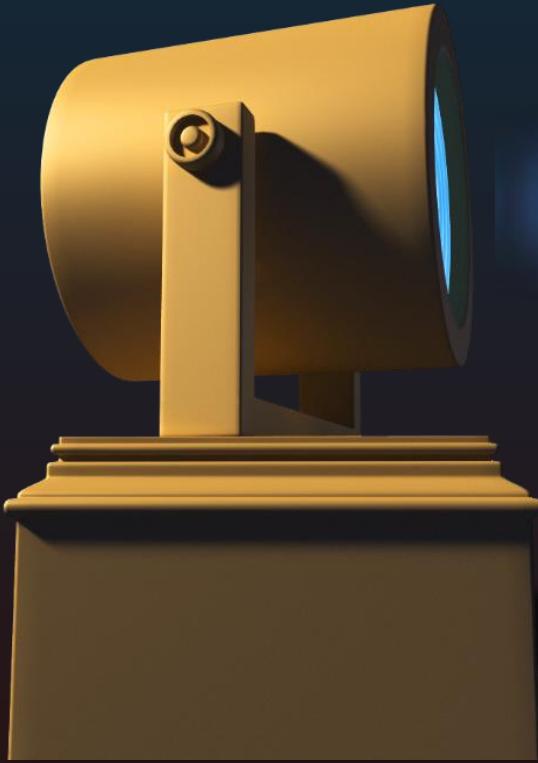
ENDMETHOD.
```

Then on uploading a file, we will get the below response.

The screenshot shows the SAP Gateway Client interface. The top navigation bar includes 'Execute', 'Select', 'Service Administration', 'Service Implementation', 'EntitySets', 'Add URI Option', and tabs for 'HTTP Method' (POST selected), 'Request URI' (/sap/opu/odata/sap/ZODATA_DEMOS_SRV/FileSet), 'Protocol' (HTTP selected), 'Test Group', and 'Test Case'. Below this is a toolbar with icons for file operations like 'Add File', 'Remove File', and 'Data Explorer'. The main area is divided into sections: 'HTTP Request' (headers: Content-Type: image/jpeg, SLUG: Test_Image1, X-CSRF-Token: v779Q41 Th6k7HMrkdHWIA==), 'HTTP Response - Processing Time = 8029 ms' (status code 201, reason Created, headers: sap-processing-info: ODatABEP=,crp=,st=,MedCacheHub=SHM,codeployed=X,softstate=, server_protocol: HTTP/1.0, content-type: application/atom+xml;type=entry; charset=utf-8), and a preview pane showing the uploaded image (Thor movie poster) and its XML representation.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
       xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
       xmlns="http://www.w3.org/2005/Atom"
       xml:base="http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/"
       d:id="http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/FileSet('Test_Image1')">
  <id>http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/FileSet('Test_Image1')</id>
  <title type="text">FileSet('Test_Image1')</title>
  <updated>2021-07-24T04:09:36Z</updated>
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"
            term="ZODATA_DEMOS_SRV.File"/>
  <link title="File" rel="self" href="FileSet('Test_Image1')"/>
  <content type="image/jpeg"
           src="FFD8FFE000104A46494600010201004800480000FFE11AF1457869600004D4D002A000000"/>
  <m:properties>
    <d:filename>Test Image1</d:filename>
  </m:properties>

```



Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

www.ZAPYard.com

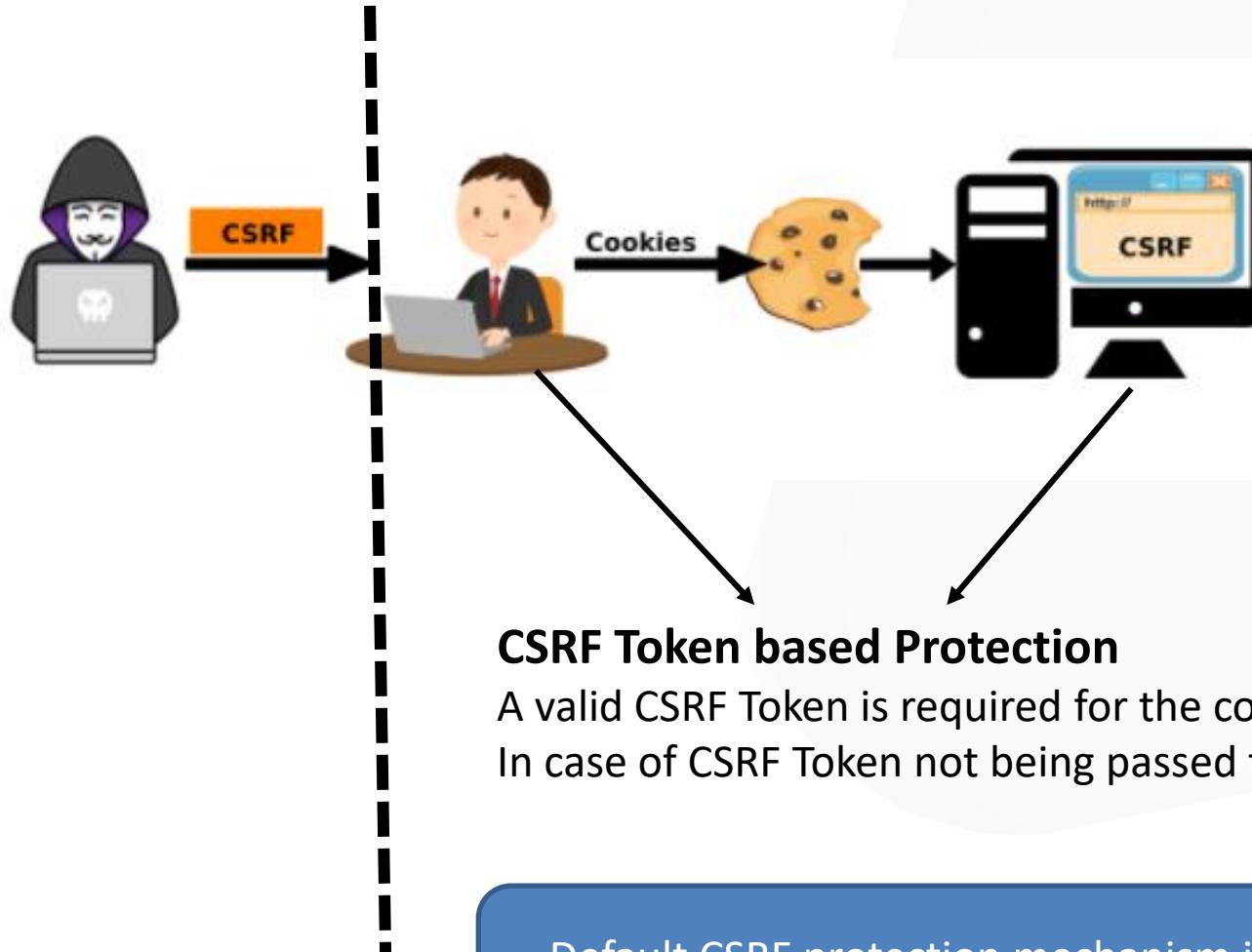
ZAPYARD

Agenda

- Handling OData/HTTP Calls via POSTMAN
- X-CSRF-Token and handling the same via Gateway Client and Postman
- How to disable the CSRF Token-based Protection Mechanism
- Creation of Simple Rest Service in S4 HANA(POST/GET Call)
- Testing of Simple Rest Service via Browser/ Postman



CSRF Token (Cross-site Request Forgery) and X-CSRF-Token



Default CSRF protection mechanism is based on the token exchange principle which is the default for the OData **Standard Mode**

Auto-Handling of CSRF Token by SAP Gateway for Modifying Requests

The screenshot illustrates the SAP Gateway Client interface for making requests to an OData service. The top navigation bar includes 'Execute', 'Select', 'Service Administration', 'Service Implementation', 'EntitySets', and 'Add URI Option'. The 'HTTP Method' dropdown is set to 'POST', which is highlighted with a yellow box and an arrow pointing from the title. The 'Request URI' is '/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/EMPLOYEE_INFOSet'. The 'Protocol' is set to 'HTTP'. The main area is divided into 'HTTP Request' and 'HTTP Response' sections.

HTTP Request:

Header Name	Value
Content-Type	application/json; charset=utf-8
X-CSRF-Token	AEWs8Zw_Ufc7KgrbE8RrAw==

HTTP Response - Processing Time = 76 ms

Header Name	Value
~status_code	201
~status_reason	Created
sap-processing-info	ODataBEP=,crp=,st=,MedCacheHub=SHM,codeployed=X,softstate=
~server_protocol	HTTP/1.0
content-type	application/atom+xml;type=entry; charset=utf-8

HTTP Request Body (JSON):

```
1: {  
2:   "Zemployeeid" : "3",  
3:   "Zfullname" : "Ajith1",  
4:   "Zemailid" : "ajith_12@gmail.com",  
5:   "Zdesignation" : "Systems Engineer",  
6:   "Zdoj" : "19.09.2019"  
7: }
```

HTTP Response Body (XML Entity Entry):

```
<?xml version="1.0" encoding="UTF-8"?>  
- <entry xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"  
  xml:base="http://Six30.mydomain.com:8770/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/"  
  xmlns="http://www.w3.org/2005/Atom"  
  xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">  
  <id>http://Six30.mydomain.com:8770/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/EMPLOYEE_INFOSet('3')</id>  
  <title type="text">EMPLOYEE_INFOSet('3')</title>  
  <updated>2021-06-19T02:09:10Z</updated>  
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"  
    term="ZU6_EMPLOYEE_INFO_SRV.EMPLOYEE_INFO"/>  
  <link title="EMPLOYEE_INFO" href="EMPLOYEE_INFOSet('3')" rel="self"/>  
  - <content type="application/xml">  
    - <m:properties>
```

Text Annotation:

The X-CSRF-Token is generated by SAP Gateway and is sent as a part of Request for successful processing and we don't need to do any manual work here.

Handling CSRF Token for Postman Scenario for Non-Modifying Requests

The screenshot shows a Postman request configuration for a GET request to `http://103.44.1.51:8770/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/EMPLOYEE_INFOSet('5')?format=json`. The **Headers** tab is selected, displaying a table with one row. A yellow highlight covers the entire row. An arrow points from the text "Way to fetch X-CSRF Token" to the "Value" column of the highlighted row.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> x-csrf-token	fetch	

Body Cookies (2) Headers (8) Test Results

		Status: 200 OK Time: 523 ms Size: 737 B Save Response
content-length	378	
x-csrf-token	S0wOOJzHraahQH6mdpEK_g==	
dataserviceversion	2.0	
sap-metadata-last-modified	Sat, 25 Apr 2020 06:10:34 GMT	
cache-control	no-store, no-cache	
sap-processing-info	ODataBEP=.cro=.st=.MedCacheHub=SHM.codeployed=X.softstate=	

Handling CSRF Token for Postman Scenario w.r.t Modifying Requests

POST http://103.44.1.51:8770/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/EMPLOYEE_INFOSet

Params Authorization Headers (12) Body Pre-request Script Tests Settings

Headers 10 hidden

KEY	VALUE	DESCRIPTION
Accept	application/json	
x-csrf-Token	S0wOOJzHraahQH6mdpEK_g==	

CSRF Token needs to be passed in Header

Body Cookies (2) Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 "d": {  
2     "__metadata": {  
3         "id": "http://103.44.1.51:8770/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/EMPLOYEE_INFOSet('3')",  
4         "uri": "http://103.44.1.51:8770/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/EMPLOYEE_INFOSet('3')",  
5         "type": "ZU6_EMPLOYEE_INFO_SRV.EMPLOYEE_INFO"  
6     },  
7     "Zemployeeid": "3",  
8     "Zfull_name": "Amitabh"  
9 }
```

Status: 201 Created Time: 154 ms Size: 705 B

HTTP Status as 403 even though valid CSRF Token is Passed –

Gateway Client Scenario for Modifying Requests

SAP Gateway Client

HTTP Method: POST

Request URI: http://Six30.mydomain.com:8770/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/EMPLOYEE_INFOSet

Protocol: HTTP

HTTP Request

Header Name	Value
X-CSRF-Token	MYENfh9YGld9q7HwgQM3uw==

HTTP Response - Processing Time = 85 ms

Header Name	Value
~status_code	403
~status_reason	Forbidden
sap-processing-info	ODataBEP=,crp=,st=,MedCacheHub=,codeployed=X,softstate=
content-type	text/plain; charset=utf-8
~server_protocol	HTTP/1.0

Request Body:

```
1 {  
2   "Zemployeeid": "2",  
3   "Zfullname": "John",  
4   "Zemailid": "john@gmail.com",  
5   "Zdesignation": "Systems Engineer",  
6   "Zdoj": "19.09.2018"  
7 }
```

Response Body:

```
1 CSRF token validation failed
```

HTTP Status as 403 even though valid CSRF Token is Passed –

POSTMAN Scenario for Modifying Requests

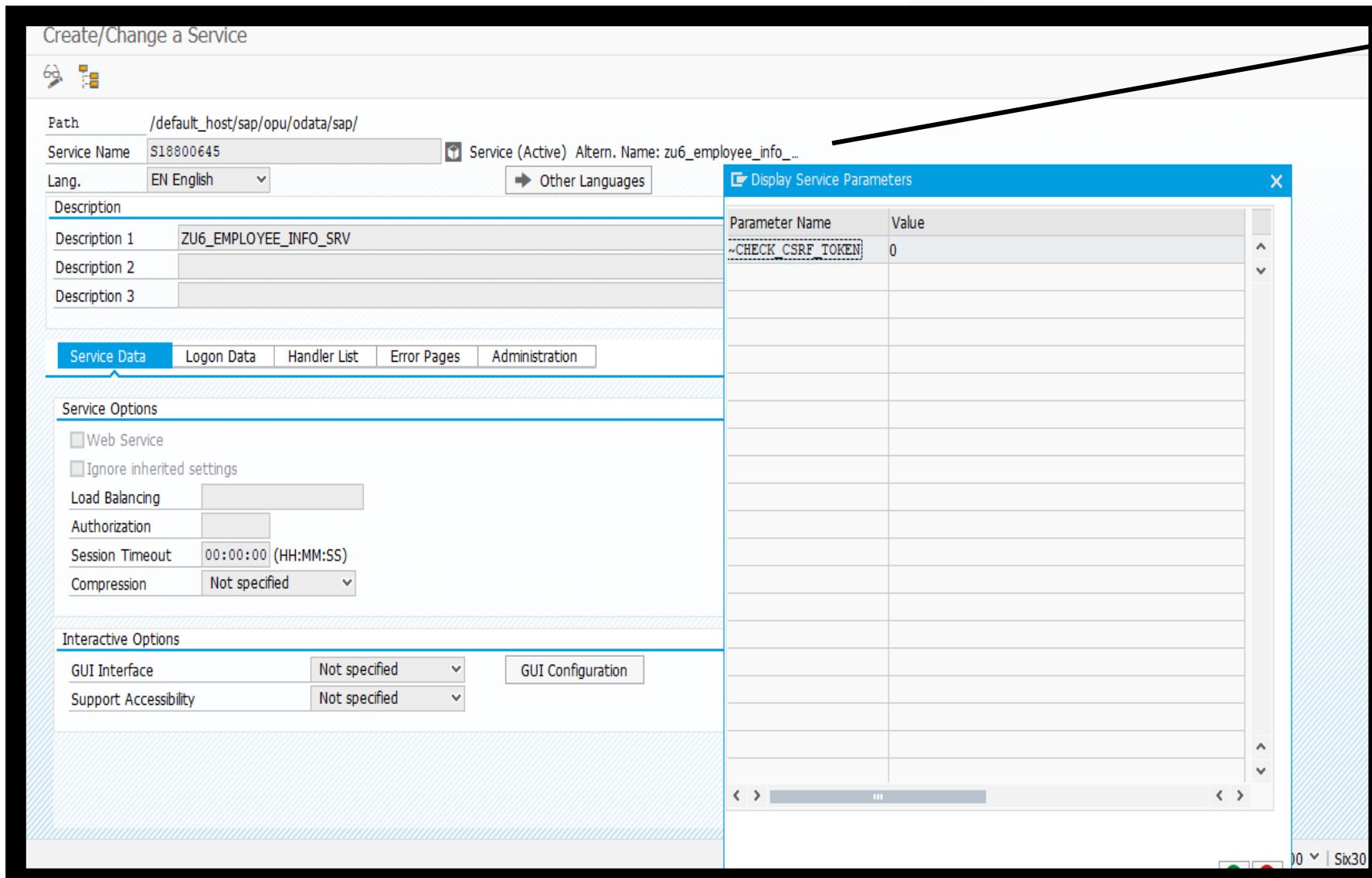
The screenshot shows a POST request to `http://103.44.1.51:8770/sap/opu/odata/SAP/ZU6_EMPLOYEE_INFO_SRV/EMPLOYEE_INFOSet`. The Headers tab is selected, showing an `x-csrf-token` header with value `R-pN1jEOrkXhKv4_ugPEBA==`. The Body tab contains JSON data for updating an employee record. The response status is `403 Forbidden`, with the message `CSRF token validation failed`.

Passed in Header

KEY	VALUE
<input checked="" type="checkbox"/> x-csrf-token	R-pN1jEOrkXhKv4_ugPEBA==

This issue comes when we disable the use of cookies for HTTP call by setting the profile parameter `login/ticket_only_by_https` to 1 under **Maintain Profile Parameters**.

Disabling the CSRF Token based Protection Mechanism



This needs to be done at GUI Config via SICF Node w.r.t Service for which we want to disable.

This parameter name needs to be passed with ~CHECK_CSRF_TOKEN and the value needs to be set as 0 for disabling the same.

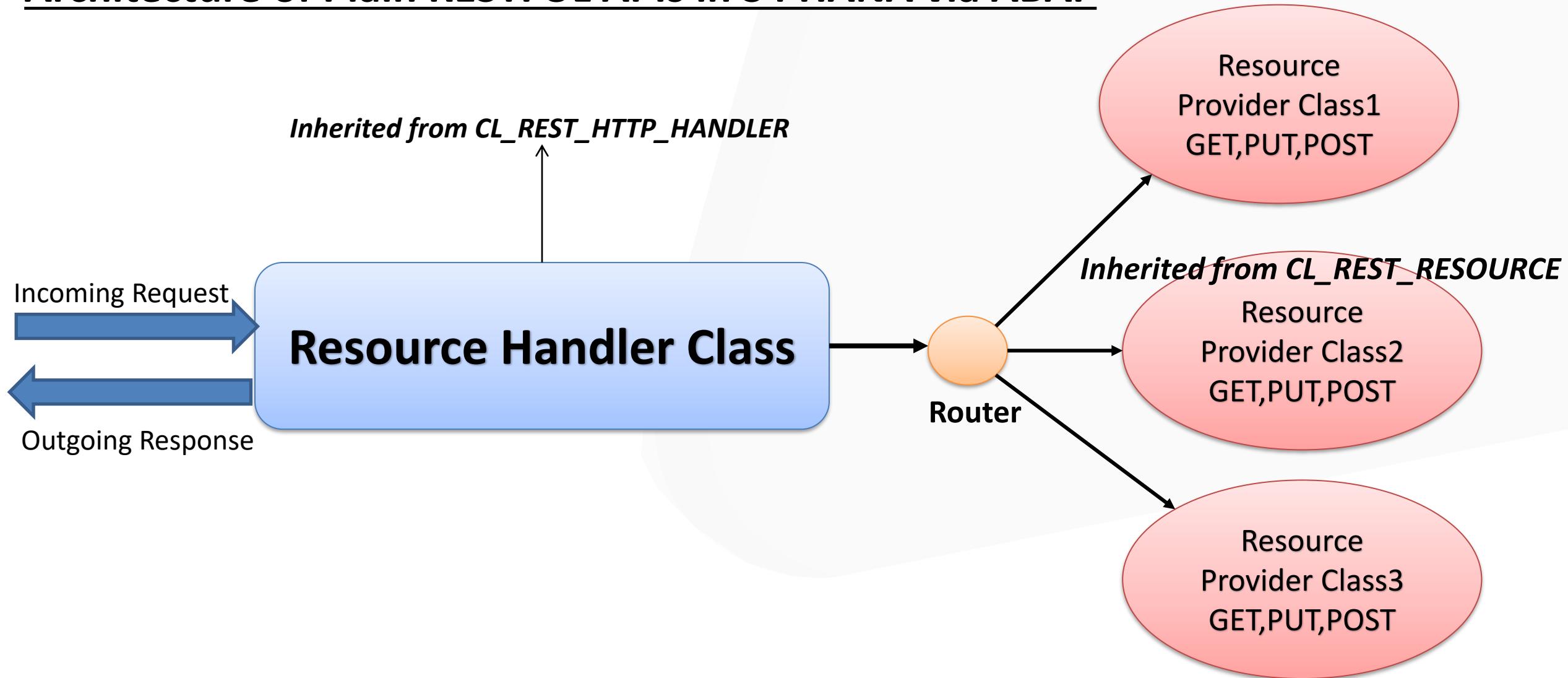
Creation of Plain RESTFUL APIs in S4 HANA via ABAP – An Alternative to OData



Expose the Data from our S4 HANA System and consume the data via Third Party Applications

Why Simple Rest Service over OData ???

Architecture of Plain RESTFUL APIs in S4 HANA via ABAP



Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



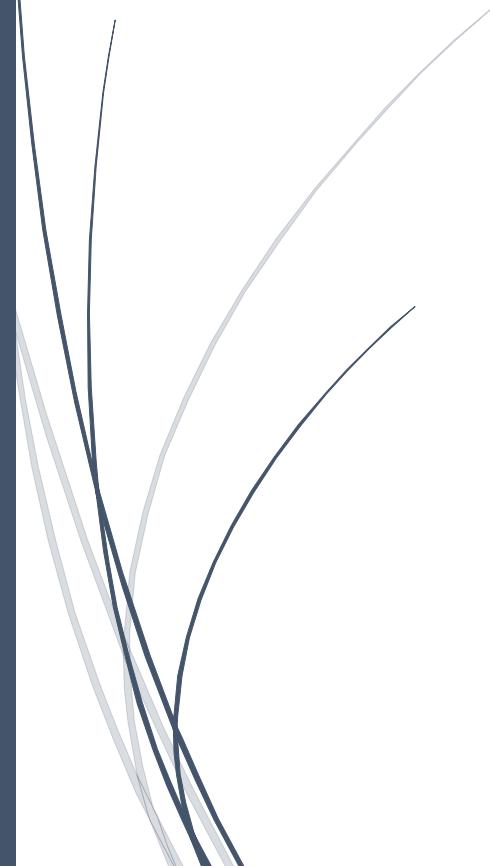
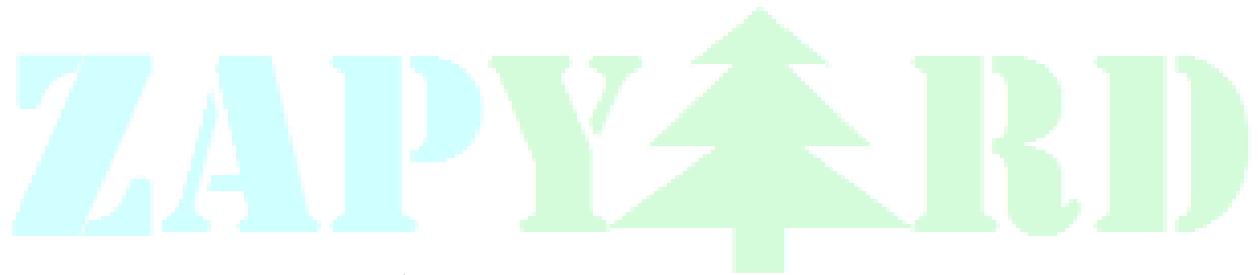
mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



Day 9

Get_Stream => We implemented the logic and we also made use of Stream Parameter. **VVI

URI:-

/sap/opu/odata/sap/ZODATA_DEMOS_SRV/FileSet('Test_Image')/\$value

Imp to access the Media Link

The screenshot shows a Postman interface with the following details:

- Method: POST
- URL: http://103.44.1.51:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/FileSet
- Headers tab selected
- Header key: SLUG, value: Test_Image_Postman
- Status: 403 Forbidden
- Message: CSRF token validation failed

Logic for Get_Stream:-

```
METHOD /iwbep/if_mgw_appl_srv_runtime~get_stream.  
DATA:ls_stream TYPE ty_s_media_resource.  
CASE iv_entity_set_name.  
WHEN 'FileSet'.  
    DATA(lv_filename) = VALUE #( it_key_tab[ name = 'Filename' ]-  
value OPTIONAL ).  
    SELECT SINGLE FROM zfileupload FIELDS *  
        WHERE filename = @lv_filename  
        INTO @DATA(ls_file).  
    IF sy-subrc EQ 0.  
        ls_stream-value = ls_file-value.  
        ls_stream-mime_type = ls_file-file_type.  
        copy_data_to_ref(  
            EXPORTING  
            is_data = ls_stream
```

Master SAP OData

```
CHANGING
    cr_data = er_stream
).
ENDIF.
WHEN OTHERS.
ENDCASE.
ENDMETHOD.
```

Modifying Operations => POST, PUT/MERGE,DELETE

x-csrf-token => fetch

Calls => 1. With x-csrf-token

2. Without x-csrf-token => X-Requested-With = x

http://103.44.1.51:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/FileSet('Test_Image')/\$value

GET http://103.44.1.51:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/FileSet('Test_Image')/\$value

Headers (9)

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> x-csrf-token	fetch	
Key	Value	Description

Body Cookies (2) Headers (11) Test Results

KEY	VALUE
set-cookie ⓘ	sap-usercontext=sap-client=800; path=/
set-cookie ⓘ	SAP_SESSIONID_S20_800=imC_yJHPvKMFFBeu92UuothmXNPs8xHrkagY2tr1Xno%3d; path=/; HttpOnly
content-type ⓘ	image/jpeg
content-length ⓘ	100307
sap-metadata-last-modified ⓘ	Sat, 24 Jul 2021 07:29:43 GMT
cache-control ⓘ	no-store, no-cache
content-type-options ⓘ	noneif

Status: 200 OK Time: 736 ms Size: 98.48 KB Save Response

Without x-csrf-token how the POST Call will work ???

Master SAP OData

The screenshot shows a Postman interface with the following details:

- URL: `http://103.44.1.51:8770/sap/opu/odata/sap/ZODATA_DEMOS_SRV/FileSet`
- Method: POST
- Authorization: Basic Auth
- Body: Text (empty)
- Response Status: 500 Internal Server Error
- Message: "Error when processing resource"

Rest API:-

Logic for Rest Handler Class:-

```
METHOD if_rest_application~get_root_handler.  
  DATA(lo_router) = NEW cl_rest_router( ).  
  lo_router->attach(  
    EXPORTING  
      iv_template = '/demo_api'  
    SOURCES  
      iv_handler_class = 'ZDEMO_RESOURCE_HANDLER'  
    *      it_parameter =  
  ).  
  
  ro_root_handler = lo_router.  
ENDMETHOD.
```

Logic for Resource Provider Class: -

POST Call:-

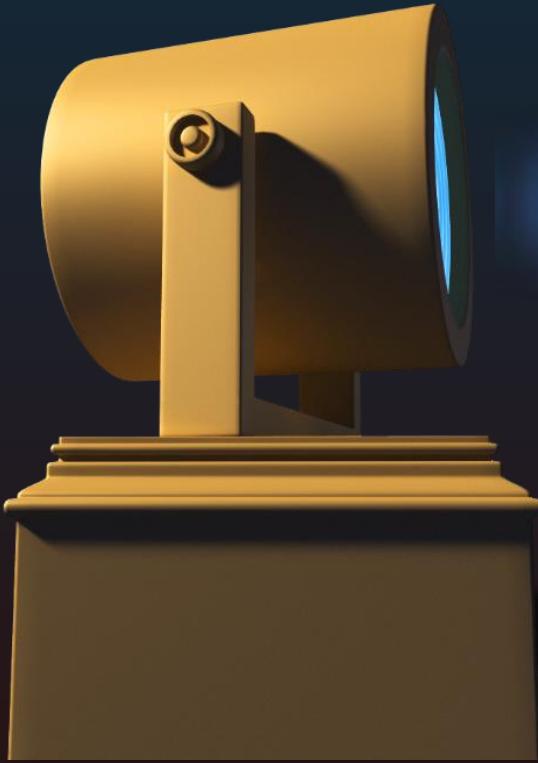
```
METHOD if_rest_resource~post.  
  DATA:ls_order TYPE zonline_order.  
  DATA(lv_data) = mo_request->get_entity( )->get_string_data( ).  
  /ui2/cl_json=>deserialize(  
    EXPORTING  
      json = lv_data  
    CHANGING  
      data = ls_order  
  ).
```

Master SAP OData

```
ls_order-online_order_id = cl_uuid_factory->create_system_uuid( ) -  
->create_uuid_c32( ).  
SELECT FROM zonline_order  
  FIELDS MAX( order_number )  
  INTO @DATA(lv_max_order_num).  
ls_order-order_number = lv_max_order_num + 1.  
ls_order-order_date = sy-datum.  
INSERT INTO zonline_order VALUES ls_order.  
/ui2/cl_json=>serialize(  
  EXPORTING  
    data          = ls_order           " Data to serialize  
*   compress      =                   " Skip empty elements  
  RECEIVING  
    r_json        = DATA(lv_data1)     " JSON string  
  ).  
  
mo_response->create_entity(  
*   iv_multipart = abap_false  
)->set_string_data( iv_data = lv_data1 ).  
mo_response->set_header_field(  
  EXPORTING  
    iv_name      = 'Content-Type'      " Header Name  
    iv_value     = 'application/json'  " Header Value  
  ).  
ENDMETHOD.
```

GET Call:-

```
METHOD if_rest_resource~get.  
  SELECT FROM zonline_order FIELDS * INTO TABLE @DATA(lt_data).  
/ui2/cl_json=>serialize(  
  EXPORTING  
    data          = lt_data           " Data to serialize  
*   compress      =                   " Skip empty elements  
  RECEIVING  
    r_json        = DATA(lv_str)      " JSON string  
  ).  
  mo_response->create_entity(  
*   iv_multipart = abap_false  
)->set_string_data( iv_data = lv_str ).  
  mo_response->set_header_field(  
    EXPORTING  
      iv_name      = 'Content-Type'      " Header Name  
      iv_value     = 'application/json'  " Header Value  
  ).  
ENDMETHOD.
```



Master SAP OData

Instructor Led Paid Online Training

mail@ZAPYard.com | whatsapp: +1-251-727-9273

www.ZAPYard.com

ZAPYARD

Agenda

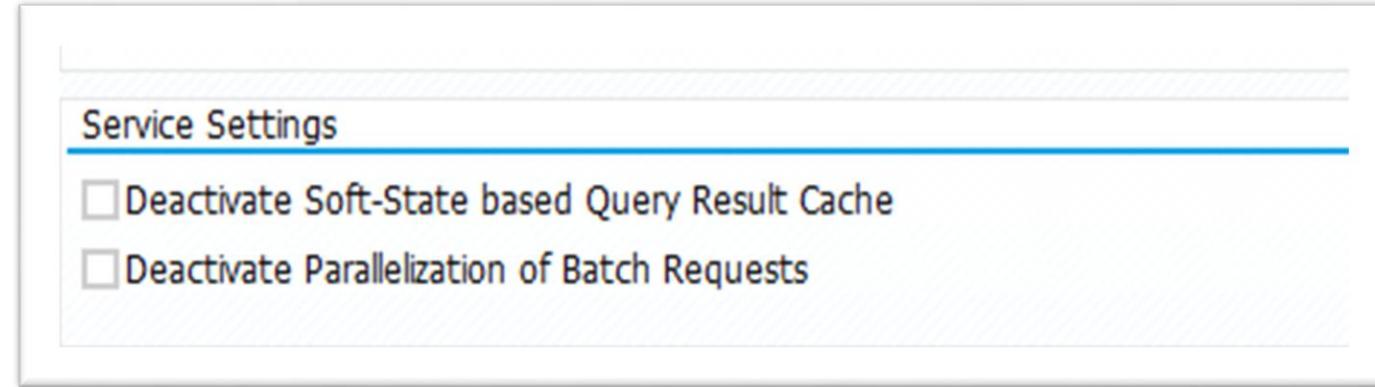
- Understanding the Concept of \$batch calls
- Exposing CDS as an OData by using an Annotation
- Enable CUD Operations with CDS based OData by using BOPF
- Exposing CDS as an OData by using CDS as a Data reference
- Handling of E-Tags and Concurrency Control for locking data while Updates



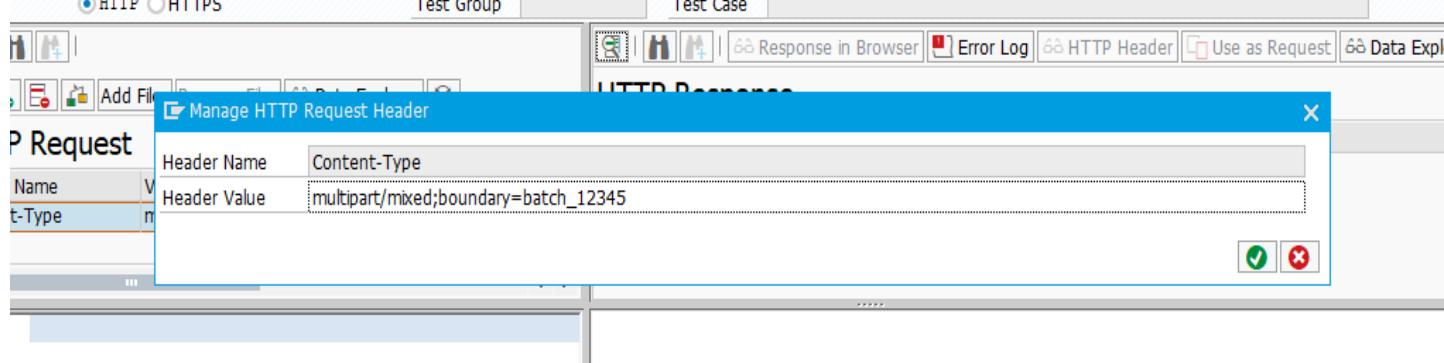
Concept of \$Batch => Used for Batch Processing

*\$batch is defined on the Service Level and the URI that will be called is:-
/sap/opu/odata/sap/<Service_name>/\$batch*

*The setting to enable/disable parallelization for a particular service can be done via T-Code
/IWBEPCONF_SERVICE*



Sending a Batch Request – Handling GET Requests



\$batch => Use Case 1 -- Sending multiple Get Requests

--batch_B01 → **Start Batch**

Content-Type: application/http

Content-Transfer-Encoding: binary

GET BusinessPartnerSet('0100000000') HTTP/1.1

Min 2 Blank Lines Have to be Used

--batch_B01

Content-Type: application/http

Content-Transfer-Encoding: binary

GET BusinessPartnerSet('0100000003') HTTP/1.1

Min 2 Blank Lines Have to be Used

--batch_B01-- → **End Batch**

Sending a Batch Request – Handling POST Request with ChangeSet

❑ \$batch => Use Case 2 -- Sending POST Requests along with GET Request

--batch_B01

Content-Type: multipart/mixed; boundary=changeset_c01

--changeset_c01

Content-Type: application/http

Content-Transfer-Encoding: binary

POST HeaderSet HTTP/1.1

sap-context-accept: header

Content-Type: application/json

Accept: application/json

{

..... JSON Body for the POST request/ We can handle in XML as well

}

--changeset_c01--

--batch_B01--

Start changeset c01

*We handle Insert, Update or Delete Operations,
Retrieve Operation is not a part of ChangeSet*

End changeset c01

Exposing CDS as an OData Service by using Annotation

Based on SADL Framework

*<CDS View>_CDS
Name of Service*

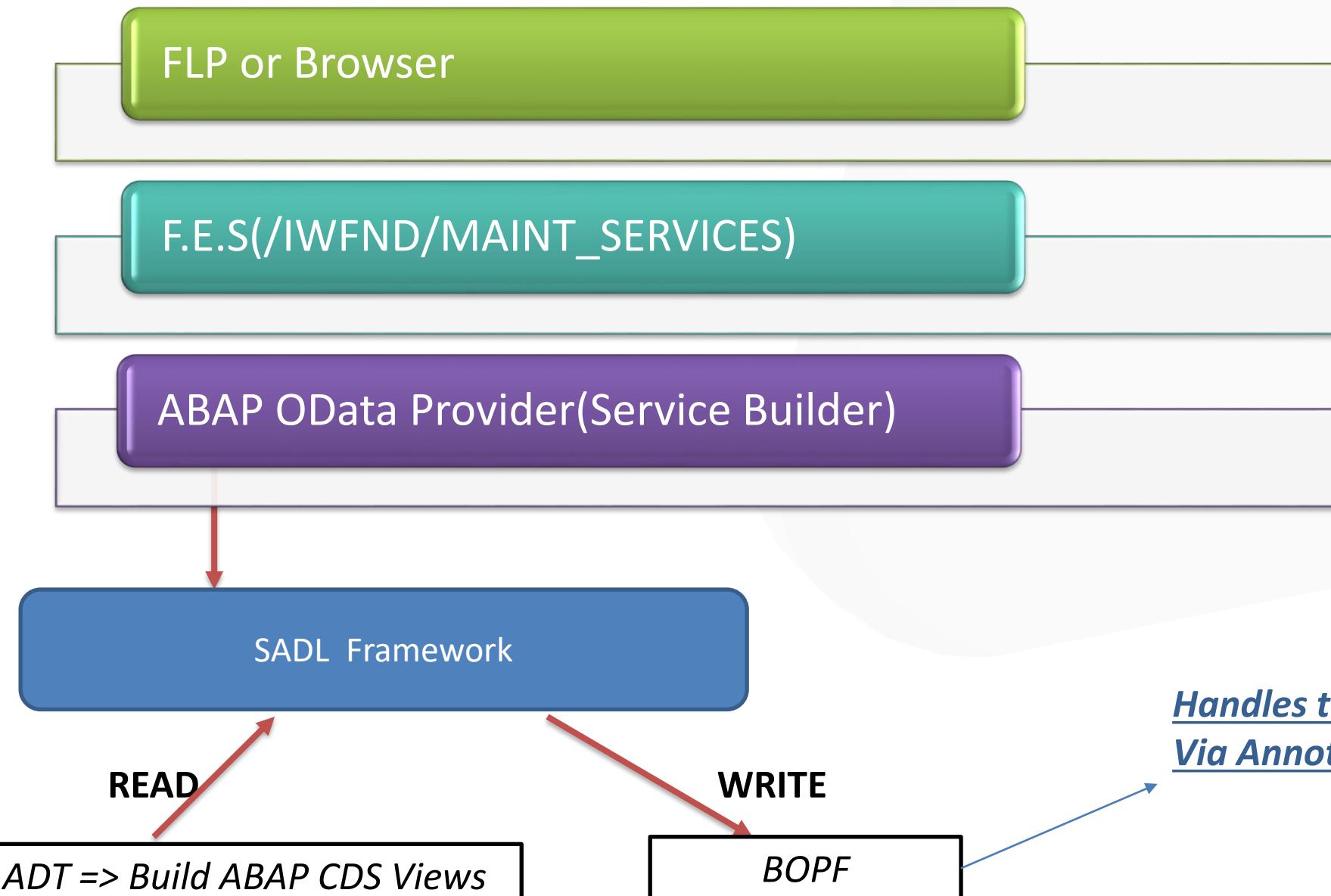


ADT => Build ABAP CDS Sources

Use Annotation to expose CDS as OData and also register and Publish the same onto SAP Gateway

The CDS view you are working on will be transformed to an Entityset with identical name of the generated OData service.

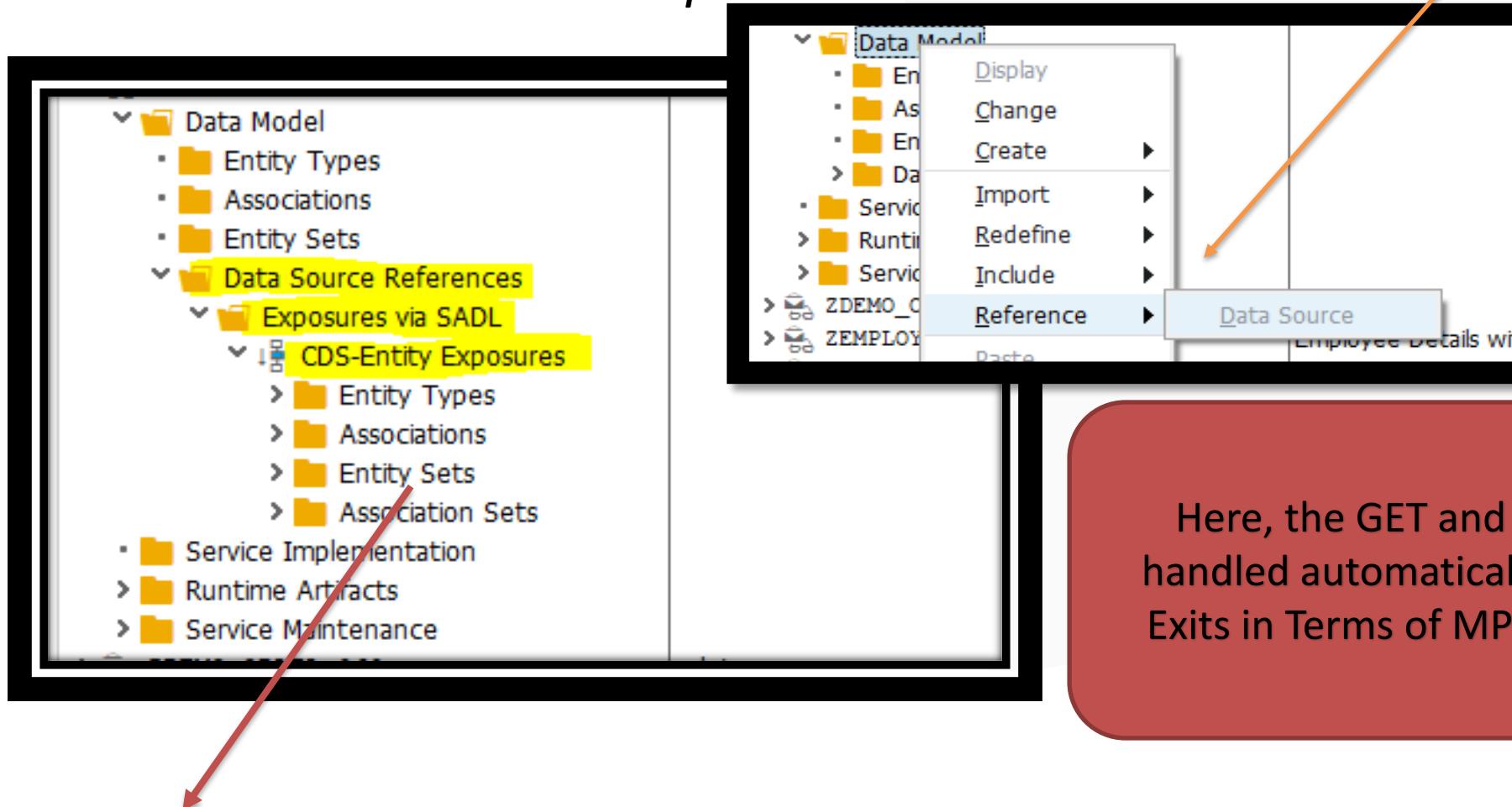
Enabling CUD Operations for CDS Based OData by usage of BOPF



*Handles the Transactional Capabilities
Via Annotation `@ObjectModel :{ }`*

Exposing CDS as an OData Service by using CDS as a Data Reference

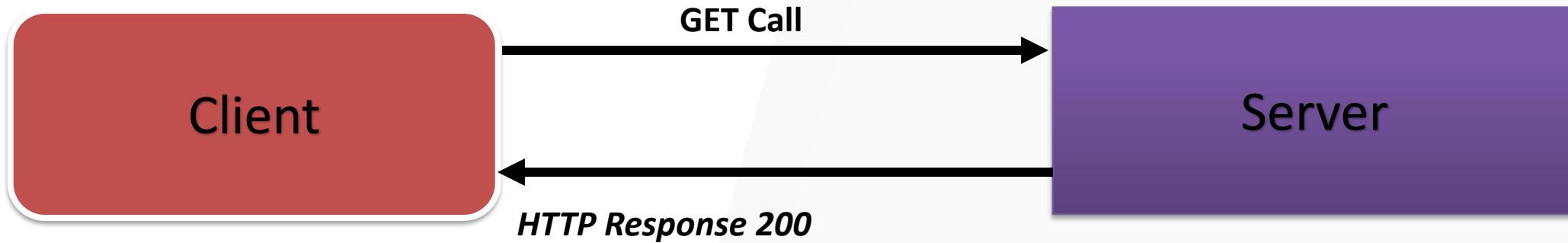
Based on SADL Framework Exposure



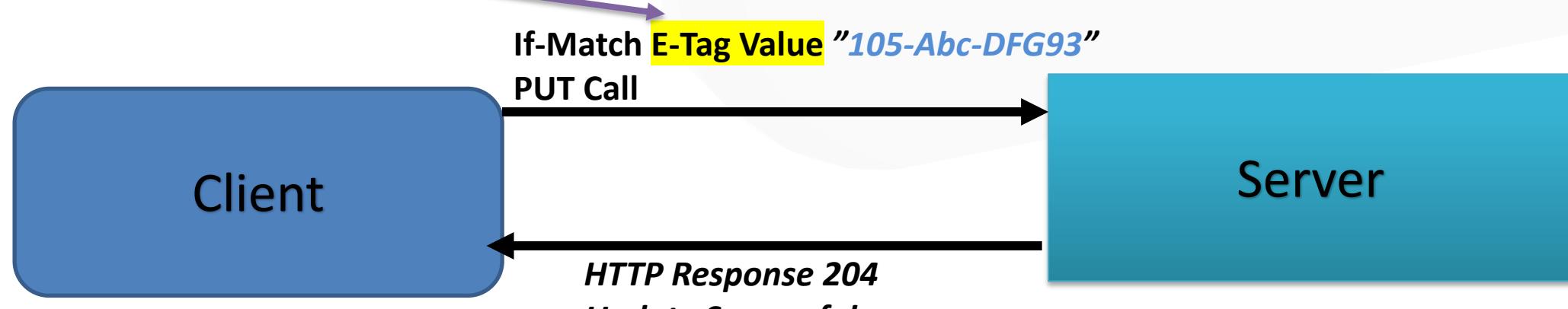
Here, the GET and Query Options are handled automatically and we have code Exits in Terms of MPC_EXT and DPC_EXT

Here the Entityset which gets Created as a part of RDS is the CDS View Entity Name

E-Tags and Concurrency Control for UPDATE Operations



Will be passed as Header Value



If the E-Tag Value sent by Client doesn't match with the one present at Server the Update will not Happen

Thank You!



LIKE



SUBSCRIBE



SUPPORT



FOLLOW

ZAPYARD



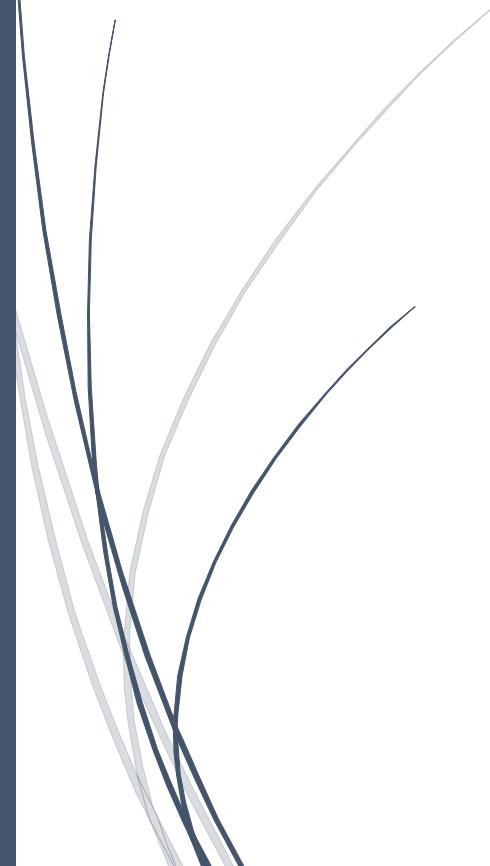
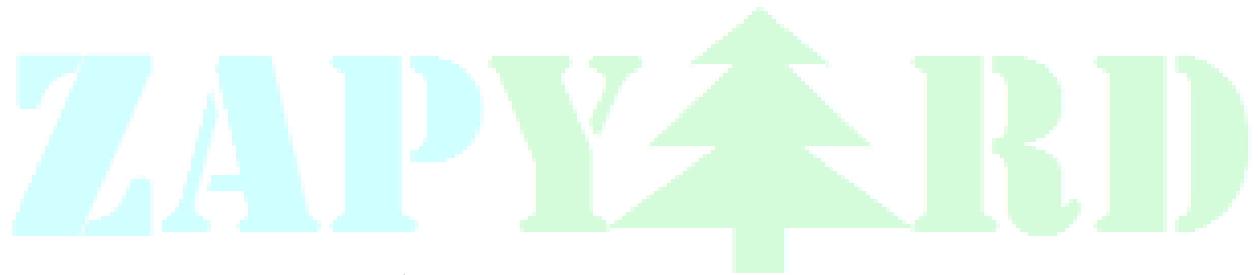
mailZAPYard@gmail.com



+1-251-727-9273
ZAPYARD

Trainer's Live Notes

Master SAP OData



Day 10

\$batch => Used to perform several service Calls in one single HTTP request.

Batch Request for multiple GET Calls:-

```
--batch_12345  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
  
GET HeaderSet('100') HTTP/1.1
```

```
--batch_12345  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
  
GET HeaderSet('101') HTTP/1.1
```

```
--batch_12345  
Content-Type: application/http  
Content-Transfer-Encoding: binary  
  
GET HeaderSet('103') HTTP/1.1
```

```
--batch_12345--
```

Insert/update/delete Operations

A \$batch call => Retrieve Operations AND/OR Change Sets

Master SAP OData

Batch Request for POST Call along with GET Call

```
--batch_12345
Content-Type: multipart/mixed;boundary=changeset_c01

--changeset_c01
Content-Type: application/http
Content-Transfer-Encoding: binary

POST HeaderSet HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

{
  "Id": "108",
  "Name": "Shyam",
  "Department": "Dept_50",
  "Approved": ""
}

--changeset_c01--
--batch_12345
Content-Type: application/http
Content-Transfer-Encoding: binary
GET HeaderSet('100') HTTP/1.1

--batch_12345
Content-Type: application/http
Content-Transfer-Encoding: binary

GET HeaderSet('101') HTTP/1.1

--batch_12345
Content-Type: application/http
Content-Transfer-Encoding: binary

GET HeaderSet('103') HTTP/1.1

--batch_12345--
```

SEGW Based(ABAP Code BASED Implementation)

- CDS Based=> 1. Mapped Data Source
- 2. RDS(Referenced Data Source)
- 3. Annotation based

SADL(Service Adaptation Description Language) => This allows us to create OData Services based on a model-driven approach using ABAP CDS Views.



```
1. Annotation one:-  
2. @AbapCatalog.sqlViewName: 'ZPODATA_V'  
3. @AbapCatalog.compiler.compareFilter: true  
4. @AbapCatalog.preserveKey: true  
5. @AccessControl.authorizationCheck: #CHECK  
6. @EndUserText.label: 'PO CDS Data'  
7. @OData.publish: true  
8. @OData.entitySet.name: 'PO_Set'  
9. define view ZPO_DATA_EXPOSE  
10. as select from ekko  
11. {  
12.   key ebeln,  
13.     bukrs,  
14.     bstyp,  
15.     bsart,  
16.     statu,  
17.     aedat,  
18.     ernam  
19. }
```

Implementing Transactional Capabilities

```
@AbapCatalog.sqlViewName: 'ZCDS_BOPF_V'  
@AbapCatalog.compiler.compareFilter: true  
@AbapCatalog.preserveKey: true  
@AccessControl.authorizationCheck: #CHECK  
@EndUserText.label: 'Bopf CDS HANDLING'  
@OData.publish: true  
@OData.entitySet.name: 'PO_Data'  
  
@ObjectModel: {  
    modelCategory: #BUSINESS_OBJECT,  
    compositionRoot: true,  
    transactionalProcessingEnabled: true,  
    createEnabled: true,  
    updateEnabled: true,  
    deleteEnabled: true,  
    writeActivePersistence: 'yeo_t01'  
}  
define view ZSRV_BOPF  
    as select from yeo_t01  
{  
    key d_id,  
    key ebeln,  
    bukrs,  
    bstyp,  
    bsart,  
    bsakz,  
    loekz,  
    statu,  
    aedat,  
    ernam,  
    custom_ip1,  
    custom_ip2,  
    on_td_flag  
}
```



CDS as an OData by enabling RDS

```
@AbapCatalog.sqlViewName: 'ZRDS_V'  
@AbapCatalog.compiler.compareFilter: true  
@AbapCatalog.preserveKey: true  
@AccessControl.authorizationCheck: #CHECK  
@EndUserText.label: 'CDS OData via RDS'  
@VDM.viewType: #BASIC  
define view ZCDS_RDS_Odata  
  as select from SEPM_I_SalesOrder_E  
{  
@ObjectModel.text.association: '_Text'  
  key SalesOrder,  
    CreatedByUser,  
    LastChangedByUser,  
    LastChangedDateTime,  
    Customer,  
    GrossAmountInTransacCurrency,  
    TransactionCurrency,  
    SalesOrderBillingStatus,  
    SalesOrderDeliveryStatus,  
    SalesOrderLifeCycleStatus,  
    _Customer,  
    _Item,  
    _Text  
}  
ZAPYARD
```

MPC_EXT logic: -

```
method DEFINE.  
  super->define( ).  
  model->get_entity_type( iv_entity_name = 'ZCDS_RDS_OdataType' )-  
>get_property( iv_property_name = 'SalesOrder_Text' )->set_updatable(  
  iv_updatable = abap_true  
  ).  
  model->get_entity_type( iv_entity_name = 'ZCDS_RDS_OdataType' )-  
>get_property( iv_property_name = 'LastChangedDateTime' )->set_as_etag( ).  
endmethod.
```

DPC EXT Logic:-

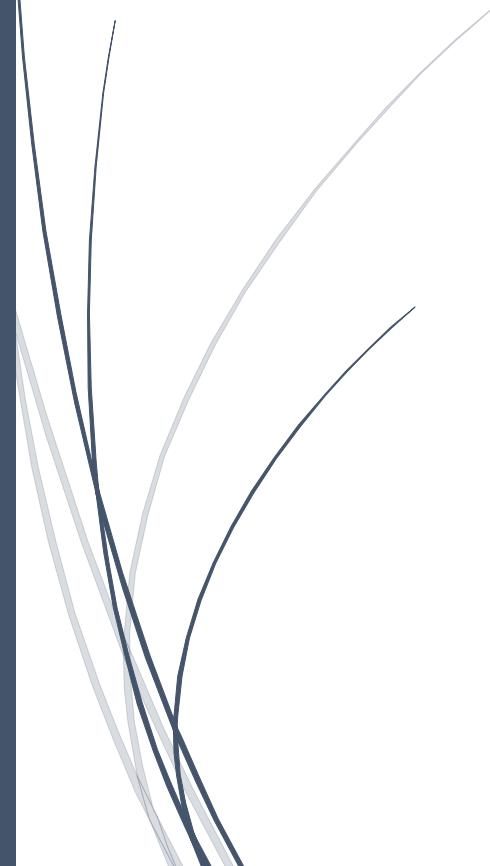
```
METHOD zcds_rds_odata_update_entity.  
DATA:ls_key_values TYPE zcl_zodata_rds_etag_mpc=>ts_zcds_rds_odatatype,  
      ls_data      TYPE zcl_zodata_rds_etag_mpc=>ts_zcds_rds_odatatype,  
      lv_so_id     TYPE bapi_epm_so_id,  
      ls_header    TYPE bapi_epm_so_header,  
      ls_headerx   TYPE bapi_epm_so_headerx,  
      lt_return    TYPE STANDARD TABLE OF bapiret2.  
io_tech_request_context->getConvertedKeys()  
  IMPORTING  
    es_key_values = ls_key_values          " Entity Key Values -  
converted  
  ).  
  
lv_so_id = ls_key_values-salesorder.  
  
io_data_provider->readEntryData(  
  IMPORTING  
    es_data = ls_data  
  ).  
  
  ls_header-so_id = lv_so_id.  
  ls_header-note = ls_data-t_salesorder.  
  ls_headerx-so_id = lv_so_id.  
  ls_headerx-note = 'X'.  
  
CALL FUNCTION 'BAPI_EPM_SO_CHANGE'  
  EXPORTING  
    so_id        = lv_so_id  
    soheaderdata = ls_header  
    soheaderdatax = ls_headerx  
  TABLES  
    return       = lt_return.  
  
  IF lt_return IS NOT INITIAL.  
    "EXCEPTIONS  
  ENDIF.  
  
ENDMETHOD.
```

E-Tag Value:-

W/"datetimeoffset'2021-07-30T19%3A53%3A57.2196850Z'"

Assignment 1

Master SAP OData



1. Provide your observations on doing the below activity with the help of T-code /iwbepl/reg_service

- Delete the Service and then try to execute the URI w.r.t. Service Document
- Unassign the Model
- Create a New Model based on the same MPC Class of our Gateway Project and then assign this new Model

Please Note: - Do not touch the trainer's Project and Service. Please use our own Gateway Project for Completing assignments.

Trainer has already discussed the tables to be used for our exercises in the class. If you missed it, please check the recordings. Also, in case you have any doubt, the trainer will discuss the solutions in the next class.

2. While fetching data from Entity set in case if we are not able to fetch any response back, we want to show a custom error message in the response as shown below: -

HTTP Response - Processing Time = 14262 ms	
Header Name	Value
~status_code	400
~status_reason	Bad Request
.....	
<pre><message xml:lang="en">No Data Returned by the Service API</message> - <innererror> - <application> <component_id/> <service_namespace>/SAP/</service_namespace></pre>	

Note: - Please implement the above logic w.r.t fetching multiple data.

3. We are able to successfully fetch data by querying an Entity set, and in the HTTP Response Header we also want to send a Custom Message/as shown below.

HTTP Response - Processing Time = 8870 ms

Header Name	Value
cache-control	no-store, no-cache
sap-message-id	<notification xmlns:sap="http://www.sap.com/Protocols/SAPData"><code>ZMSG_00000000000000000000000000000000</code>3 rows fetched successfully</message>
sap-perf-fesrec	8861603.000000

Logic:- For the Custom Message we need to display the Message as "`<Exact_Number_of_Rows_in_EntitySet> rows fetched successfully`".

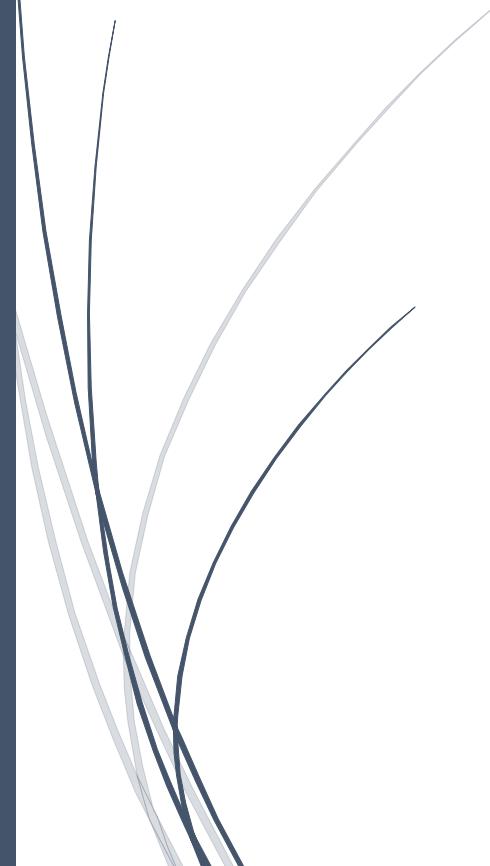
4. Create a Data Model having more than 1 key property and then implement the logic in the respective class and then do a query on the same to read a single entity on the basis of Key fields and then show the response along with the URI from Gateway Client.

Note: - You need to have at least 2 key properties.

5. If we add/remove any property from a EDM and without regenerating runtime objects if we try to call the Service Metadata Document what will be the observation/output?

Assignment 2

Master SAP OData



Master SAP OData

1. Create a Z Table which will have basic employee details like Employee ID, Employee Full Name, Employee Email ID and Designation as shown below.

Field	Key	Ini...
ZEMPLOYEEID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ZFULLNAME	<input type="checkbox"/>	<input type="checkbox"/>
ZEMAILID	<input type="checkbox"/>	<input type="checkbox"/>
ZDESIGNATION	<input type="checkbox"/>	<input type="checkbox"/>

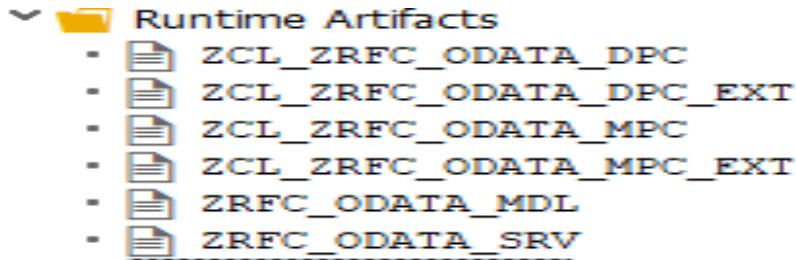
2. Create Z-RFC Enabled FMs for Create, Read, Update and Delete Operation.

Note: - 4 separate FMs to be created along with logic to handle CRUD Operation separately and you can provide names like ZFM EMP CREATE, ZFM EMP GET (This FM should have the provision of returning a single entry based on Employee ID or all entries as well), ZFM EMP UPDATE, ZFM EMP UPDATE

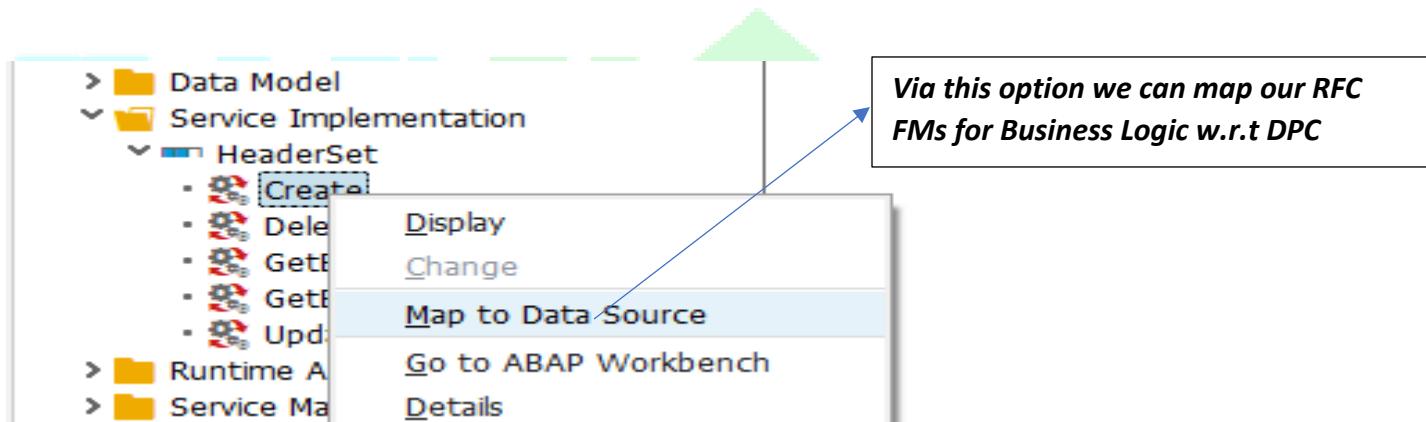
3. Now create a Gateway Project and proceed with the Creation of Data Model by Importing from DDIC Structure and then generate the Runtime Objects as shown below.

Properties															
Name	Is Key	Edm Core Type	Prec.	Scale	Max ...	Unit Prop.	Creat...	Upda...	Sorta...	Nulla...	Filt.	Label	La...	Comp. Type	ABAP Field Name
Zemployeeid	<input checked="" type="checkbox"/>	Edm.String	0	0	50		<input type="checkbox"/>	c	T		ZEMPLOYEEID				
Zfullname	<input type="checkbox"/>	Edm.String	0	0	50		<input type="checkbox"/>	c	T		ZFULLNAME				
Zemailid	<input type="checkbox"/>	Edm.String	0	0	50		<input type="checkbox"/>	c	T		ZEMAILID				
Zdesignation	<input type="checkbox"/>	Edm.String	0	0	50		<input type="checkbox"/>	c	T		ZDESIGNATION				

Master SAP OData



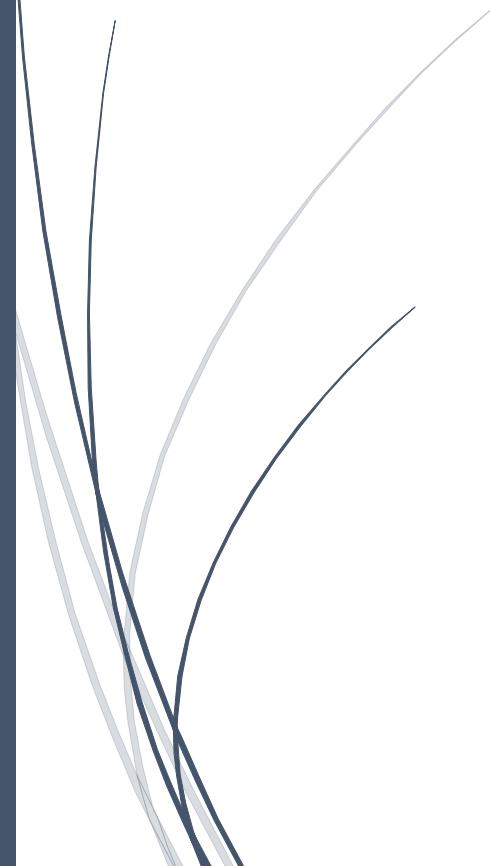
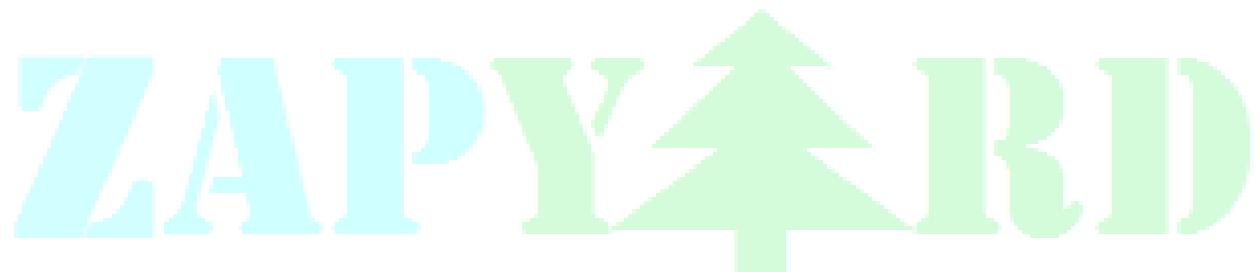
4. Now proceed with the Service Implementation and here you need to generate the Business Logic using the RFC FMs which we created earlier as a part of *Step 2* for the CRUD Operations. So basically, we need to map the individual RFC FMs to the Data Source for every Operation related to Entity Set.



5. Once the mapping is completed, generate run time objects again and then if we go to DPC Class we will find that our Business Logic has been generated w.r.t FMs mapped.
6. Once all the above activities are completed proceed with registering your OData Service so that they can be consumed and once the registration is done now, we can test the Service w.r.t Service Document and Metadata Document and also w.r.t CRUD Operations on Entity Set Created.

Assignment 3

Master SAP OData



Master SAP OData

1. Create a Gateway Service via **SEGW** and it should have a Corresponding Entity and Entity Set having the following properties along with the ProductId as a key: -

Name	Is Key	Edm Core Type	Prec.	Scale	Max ...	Unit Prop.	Creat...	Upda...	Sorta...	Nulla...	Filt.	Label	La...	Comp. Type	ABAP Field Name
ProductId	<input checked="" type="checkbox"/>	Edm.String	0	0	10							Product ID	T		PRODUCT_ID
Category	<input type="checkbox"/>	Edm.String	0	0	40							Category	T		CATEGORY
Name	<input checked="" type="checkbox"/>	Edm.String	0	0	255							Description	T		NAME
Description	<input type="checkbox"/>	Edm.String	0	0	255							Description	T		DESCRIPTION
SupplierId	<input type="checkbox"/>	Edm.String	0	0	10							Business Partner...	T		SUPPLIER_ID
SupplierName	<input type="checkbox"/>	Edm.String	0	0	80							Company	T		SUPPLIER_NAME
Price	<input type="checkbox"/>	Edm.Decimal	23	4	0							Price	T		PRICE
CurrencyCode	<input type="checkbox"/>	Edm.String	0	0	5							Currency Code	T		CURRENCY_CODE

2. The service needs to fetch all the records as a part of GET Call by making use of the FM: - **BAPI_EPM_PRODUCT_GET_LIST** as this FM has the provision to filter the records on the basis of ProductId, SupplierName and Category.

3. Now w.r.t Point 2, while fetching the records the service should have the provision of filtering the records on the basis of=>

ProductId, SupplierName, Category and Price

E.g.: - If the Consumer passes a Filter Query as shown below along with the Service URI for the Entity Set:-

\$filter=ProductId eq 'HT-1000' and Category eq 'Notebooks' and SupplierName eq 'SAP' and Price lt 1500

The Service should be capable of handling the Filter Parameters

4. Analyze the below Filter Query and let us know do you observe any issue with this query? If yes what should be the respective correction for the same?

`$filter=(ProductId eq 'HT-1000' or ProductId eq 'HT-1020')
and (Category eq 'Notebooks' or Category eq 'PDAs &
Organizers')`

5. Analyze the below data and w.r.t the data provided below can we implement a Filter Query where in if we pass the *Filter Query as \$filter=Name eq 'J*' and (Id eq '101' or Id eq '105')*.

Id	Name	Department	Created on
101	JOHN	DEPT2	07.05.2021
102	JIMMY	Dept_3	07.06.2021
105	Joe	Math	04.07.2021
106	Javed	Dept_8	06.07.2021

*And we get the desired output with the below entries only.
Output of the Service Response should contain the below data only:-*

Id	Name	Department	Created on
101	JOHN	DEPT2	07.05.2021
105	Joe	Math	04.07.2021

6. W.r.t data provided below we need the OData service to filter from this set of data and provide us an output based on the filter criteria: -

Id eq 100 & Created on date should be between 01.04.2021 to 10.05.2021

Employee ID	Id	Name	Department	Created on
	100	SAM	DEPT1	03.04.2021
	101	JOHN	DEPT2	07.05.2021
	102	JIMMY	Dept_3	07.06.2021
	104	RAVI	Dept_7	03.07.2021
	105	Joe	Math	04.07.2021
	106	Javed	Dept_8	06.07.2021

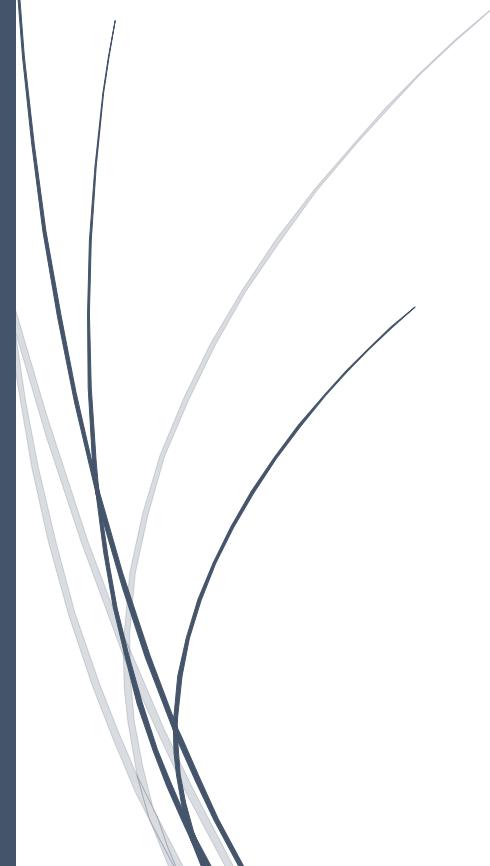
Output of the Response: -

```
<d:Id>100</d:Id>
<d:Name>SAM</d:Name>
<d:Dept>DEPT1</d:Dept>
<d:CreatedOn>2021-04-03T00:00:00</d:CreatedOn>
```

Note: - Here you need to focus on how to filter based on Date Range considering the limitations w.r.t Operators as well as we are passing one more property as well.

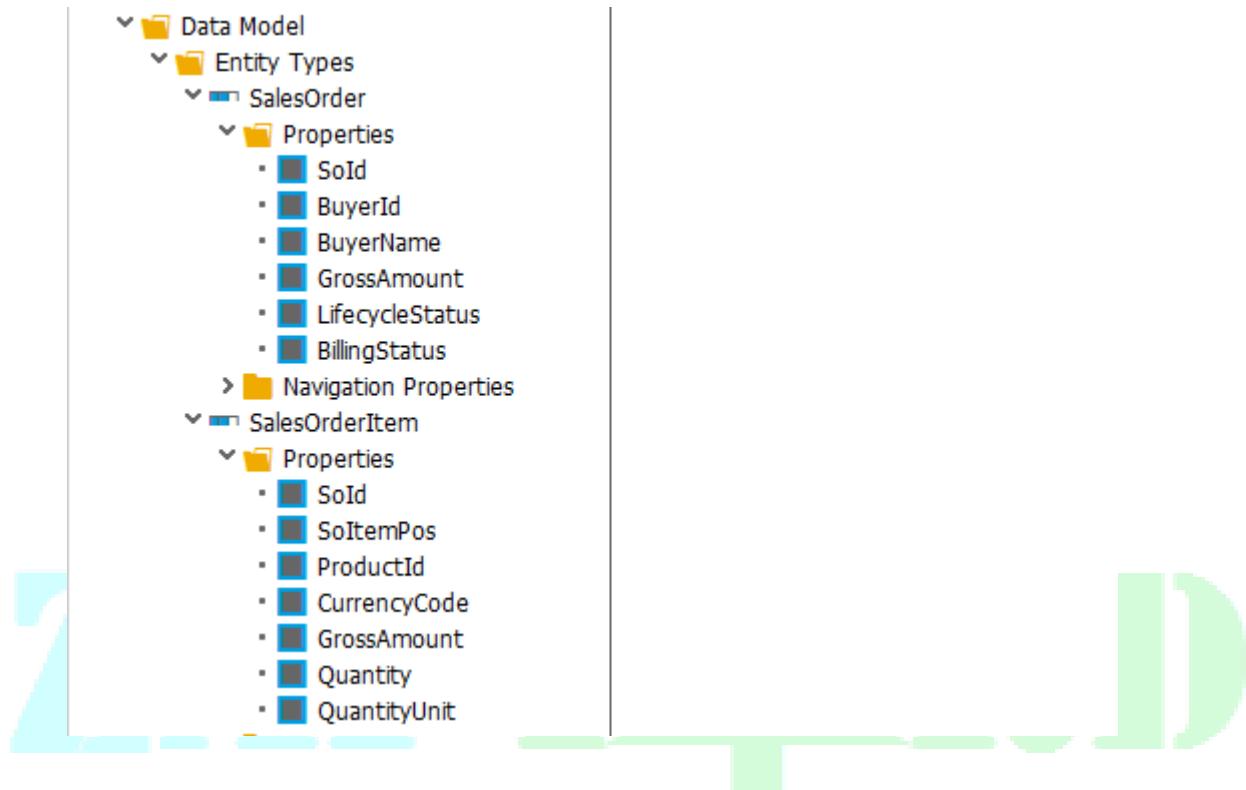
Assignment 4

Master SAP OData



Master SAP OData

1. Create a Gateway Service via SEGW and it should have a Corresponding Entity and Entity Set having the following properties along with the Sold (Sales Order ID) as a key: -



2. Create the Association and the Navigation Property between Sales Order and Sales Order Item and then generate the Runtime Artifacts and Register the Service.

Associations						
Name	Ex... Principal Entity	Principal Entity Cardinality	Dependent Entity	Dependent Entity Cardinality	Label	
Order_Item	SalesOrder	1	SalesOrderItem	N		

Navigation Properties				
Name	Relationship Name	ABAP Field Name	Label	
Order_Item	Order_Item	ORDER_ITEM		

3. Now using the Concept of Deep Insert Create a Sales Order which will have 10 Line Items and we should be able to create a new Sales Order as shown below.

The screenshot shows the SAP Data Explorer interface. On the left, the 'HTTP Request' pane displays a JSON POST body for creating a Sales Order:

```
1 {  
2   "d" : {  
3     "__metadata" : {  
4       "id" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesOrder'  
5     },  
6     "BuyerId" : "100000001",  
7     "Order_Item" : {  
8       "results" : [  
9         {  
10           "__metadata" : {  
11             "id" : "http://Six30.mydomain.com:sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesOrderSet('500000021')/Order_Item('500000001')"  
12           },  
13           "SoId" : "500000000",  
14           "SoItemPos" : "10",  
15           "ProductId" : "HT-1000",  
16         }  
17       ]  
18     }  
19   }  
20 }
```

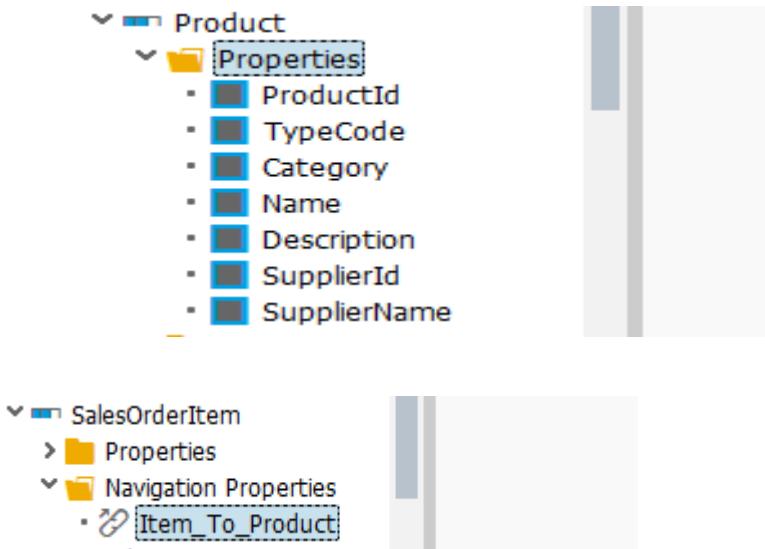
On the right, the 'HTTP Response - Processing Time = 818 ms' pane shows the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>  
- <entry xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"  
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"  
  xmlns="http://www.w3.org/2005/Atom"  
  xml:base="http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/">>  
  <id>http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesOrderSet('500000021')</id>  
  <title type="text">SalesOrderSet('500000021')</title>  
  <updated>2021-07-19T17:24:09Z</updated>  
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"  
    term="ZGW_SALES_ORDER_SRV.SalesOrder"/>  
  <link title="SalesOrder" rel="self" href="SalesOrderSet('500000021')/"/>  
  - <link title="Order_Item" type="application/atom+xml;type=feed"  
    rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Order_Item"  
    href="SalesOrderSet('500000021')/Order_Item">  
    - <m:inline>  
      - <feed  
        xml:base="http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/">  
        <id>http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZGW_SALES_ORDER_SRV/SalesO
```

The Service should be capable of handling the \$expand Data Provider Framework so that we can do a GET Call to use the same Response as a request for POST Call.

4. Create one more Entity having product Details as shown below and also create an Association and Navigation between Sales Order Item and Product so that we can get Product Details based on Sales Order ID and its Corresponding Single Line Item.

Master SAP OData



5. Once point 4 is done, implement multi-Level expansion between Sales Order, Sales Order Item and Product by using Association and Navigation so that if we call the URI as shown below, we should get Product details as below.

The screenshot shows the SAP OData service studio results page. At the top, it displays the request URI: /sap/opu/odata/sap/ZLEARN_ODATA_SRV/SalesOrderSet('500000000')/Order_To_Item(Sold='500000000',SolItemPos='10')/Item_To_Product?\$format=json. Below this, the 'HTTP Response - Processing Time = 5092 ms' is shown. The response header table includes rows for Header Name (Value) and Status Code (200). The response body is a JSON object representing a product detail:

```
1  {
2    "d" : {
3      "__metadata" : {
4        "id" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ODATA_SRV/ProductSet('HT-1000')",
5        "uri" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ODATA_SRV/ProductSet('HT-1000')",
6        "type" : "ZLEARN_ODATA_SRV.Product"
7      },
8      "ProductId" : "HT-1000",
9      "TypeCode" : "PR",
10     "Category" : "Notebooks",
11     "Name" : "Notebook Basic 15",
12     "Description" : "Notebook Basic 15 with 2,80 GHz quad core, 15\" LCD, 4 GB DDR3 RAM, 500 GB H",
13     "SupplierId" : "0100000046",
14     "SupplierName" : "SAP",
15     "Prdtc_To_Supplier" : {
16       "__deferred" : {
17         "uri" : "http://Six30.mydomain.com:8770/sap/opu/odata/sap/ZLEARN_ODATA_SRV/ProductSet('HT-1000')/Supplier"
18       }
19     }
20   }
```

URI for reference: -

**SalesOrderSet('500000000')/Order_To_Item(Sold='500000000',
0',SolItemPos='10')/Item_To_Product?\$format=json**

6. Also, by making use of the Data Provider \$expand implement the logic for calling the below URI.

`/sap/opu/odata/sap/<Service_Name>/SalesOrderSet('500000000')?$expand=Order_Item/Item_To_Product&$format=json`

Note: - You need to make use of `Get_Expanded_EntitySet`

