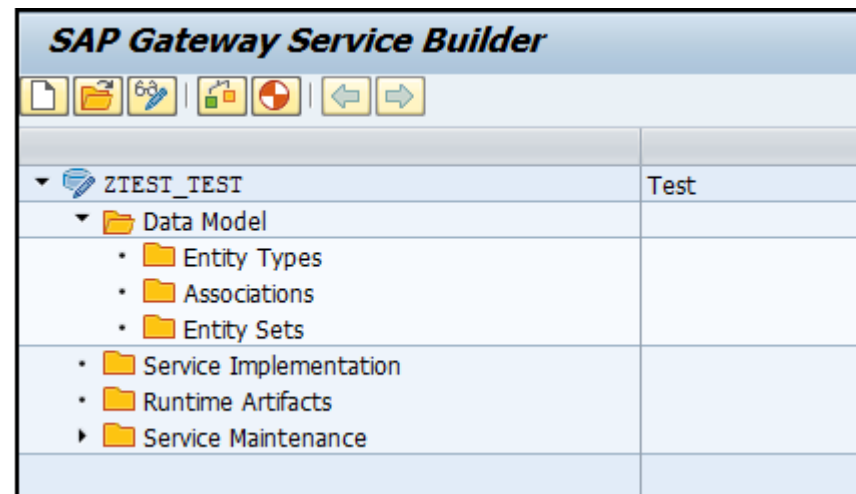**DAY 1B**

SAP Gateway

Day 1B Agenda

❖ Recap of Day 1A

❖ SEGW demo

❖ Service creation & CRUDQ operations demo

❖ Service Maintenance

❖ Testing in Gateway Client

❖ Support for Maintenance & Monitoring

Gateway Service Builder – SEGW

The Service Builder creates a new project structure, which you can use to develop or generate your new service

Steps Involved in Creating a Gateway Project

- ❖ Launch Transaction SEGW.

- ❖ Click on Create.

- ❖ In the Create Project dialog box, enter a unique name for your new project. The project name must not exceed 30 characters and must not start with a number or include any special characters (underscores are permitted). For example, ZTEST_TEST.

- ❖ Enter a short description for your new project. You must enter a short description otherwise you cannot create the new project. The description must not exceed 60 characters. The short description should provide additional context information about your project. The short description you define here is displayed in the Service Builder together with the project name.

- ❖ Under Attributes, click project type Service with SAP Annotations to enable you to use standard annotations (such as, creatable, updatable) to specify how data model artifacts are to be used.

- ❖ Under Attributes use the input help to select Standard as the generation strategy to be used.

- ❖ Enter the name of the package in which the project is to be created or choose Local Object if you want to create the project locally.

- ❖ Click Enter to create your new project together with its unique name and description in the location you specified.
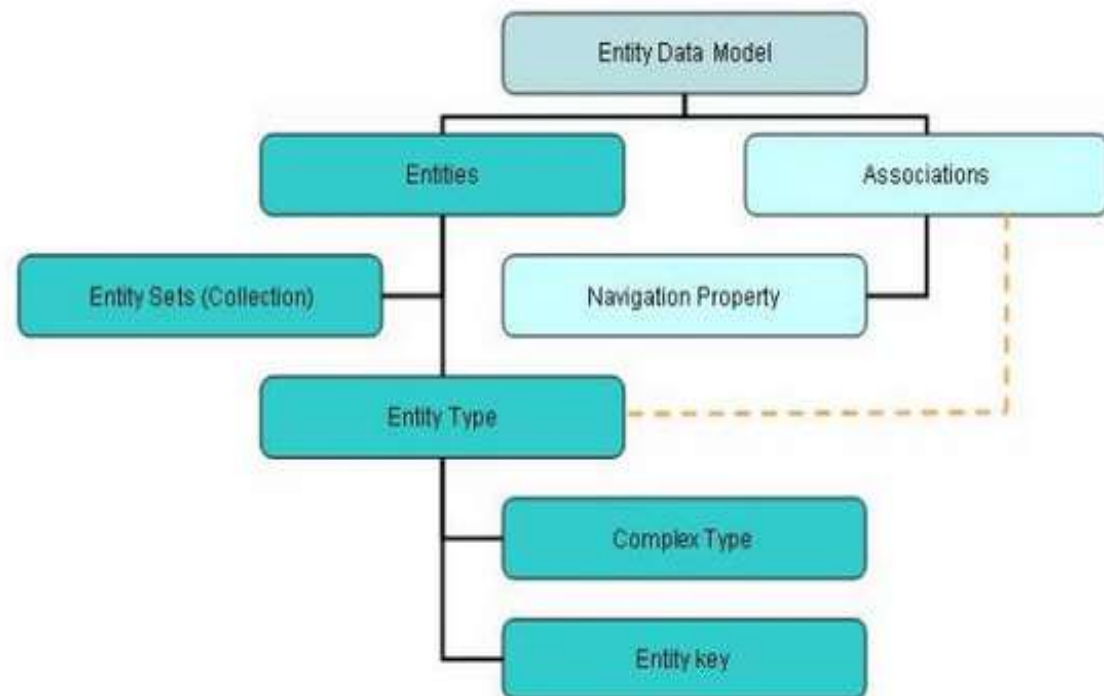
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

- We need to be familiar with below terms before we start with any gateway development using Service builder transaction

  - Entity Type
  - Entity Set
  - Association
  - Navigation
  - MPC(Model Provider Class)
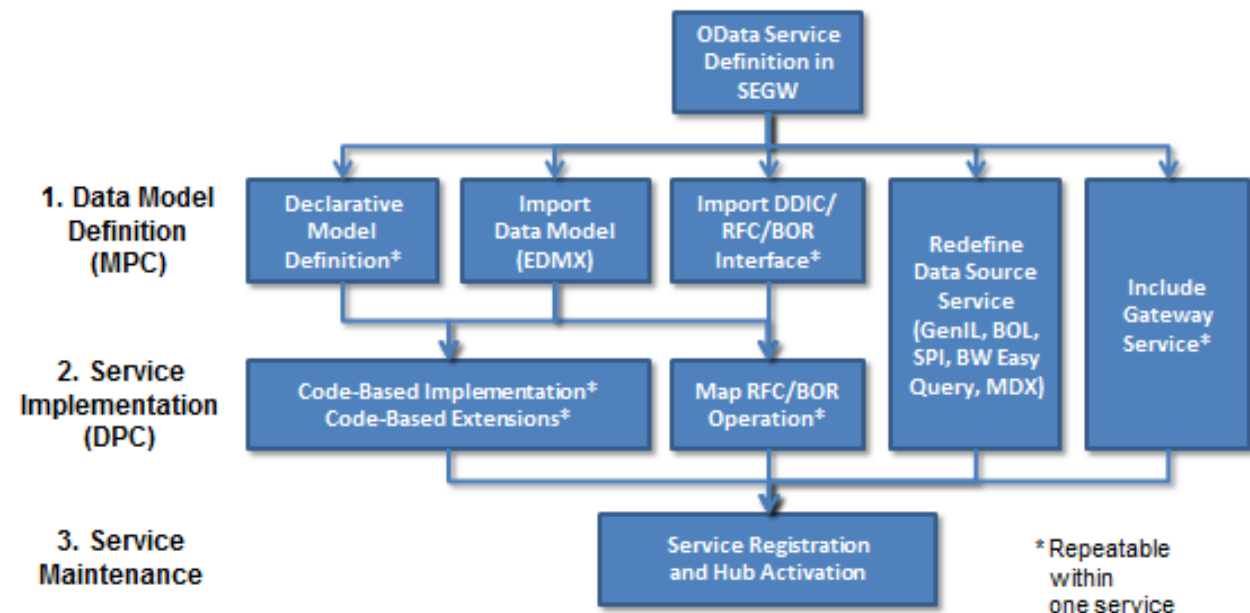 - DPC(Data Provider Class)

Data modeling in SAP Gateway

❖ OData services require a data model definition, referred as model provider class.. Based on your specific requirements, you can define a data model in the following formats:

- Define new data model
- Import
- Redefine Service
- Include Service

Gateway Model Builder

- ❖ GW Model Builder:

  - ✓ Data Model

  - ✓ Service Implementation

  - ✓ Runtime Artifacts

  - ✓ Service Maintenance

Data model elements

❖ This model offers optimum flexibility, wherein it requires manual definition of individual data model elements and their properties.

▪ Within the tree view of the Service Builder, you can use the data model folder structure to create and edit the individual data model elements. The data model consists of various sub-folders for each of the element types as listed below:

- ▪ Entity Types
- ▪ Complex Types
- ▪ Associations
- ▪ Entity Sets
- ▪ Association Sets
- ▪ Function Imports
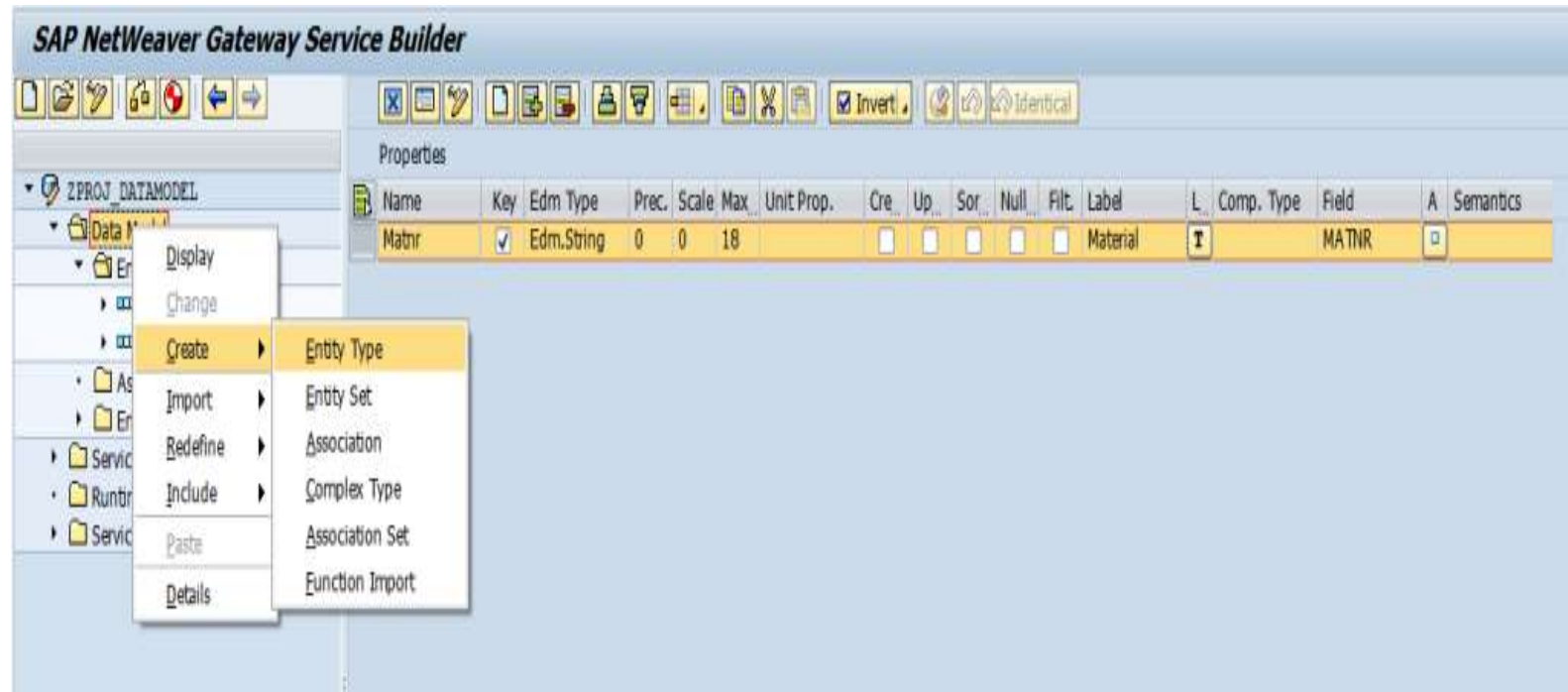
Development Phases for OData service

1. ## Model Provider Class (MPC)

   ❖ Definition of individual data model elements and their properties

2. ## Data Provider Class (DPC)

   ❖ Definition of runtime artifacts (operations & methods)

   ❖ Code-based implementation

   ❖ Mapping operations to a data source

3. ## Service Maintenance

   ❖ Registering & activating the service
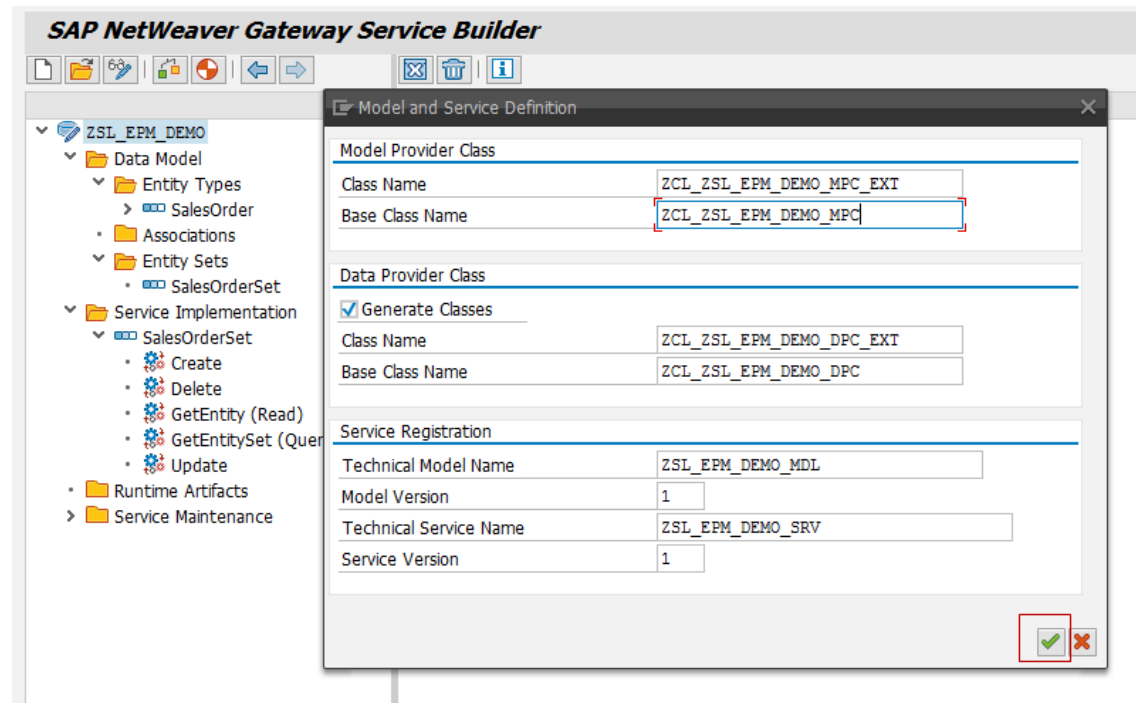
Gateway Model Builder

❖ Runtime Artifacts :

When we generate the project in GW builder, the below classes are generated under the Runtime

Artifacts.

- ✓ Model Provider Class

- ✓ Model Provider Extension Class

- ✓ Data Provider Class

- ✓ Data Provider Extension Class

Creating the ABAP Classes

- Generate Button

Registering the Service
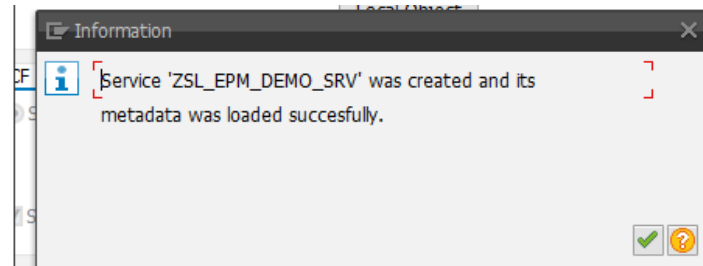
❖ To register the OData service, go to **/IWFND/MAINT_SERVICE** transaction.

❖ Click on Add Service button to register the service

❖ Search for the service name in the system where you have created your service. Click on **Get Services**

- Hub Deployment – Provide the alias as your back-end system.
- Embedded Deployment – Provide "LOCAL"

❖ Select the service and Click in Add Selected Service button

❖ Assign the package to the service and leave other fields as it is and Click on OK.

❖ You will get below popup saying that service is registered successfully.

Service Maintenance
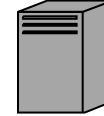
❖ We need to Register the service in the respective System Alias maintained in the Service Maintenance.

❖ Once the Service is registered then we can maintain the service to test from Gateway Client.

❖ Go back to service maintenance tcode /IWFND/MAINT_SERVICE, You will find the service and technical name of the service will be <ProjectName>_SRV.
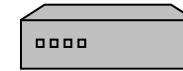
Development Process

	RFC/BOR Client	Gateway Client	Consumer

1.Pre-Requisites (AddOn / SDK) ✓ ✓ ✓

2.Define or Generate Meta data Provider Class ✓ (Model)

3.Define or Generate Data Provider Class ✓ (Code)

4.Register Service ✓ (Configuration)

5.Test the service ✓ (Testing)
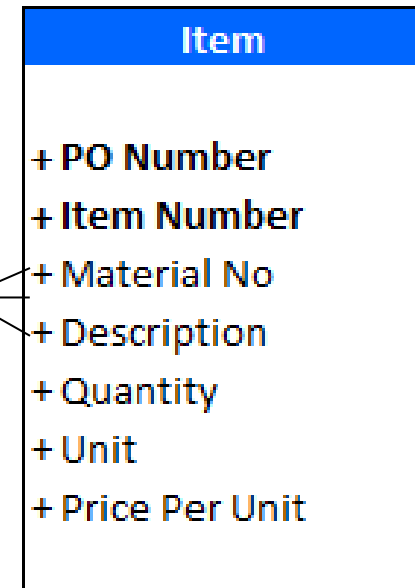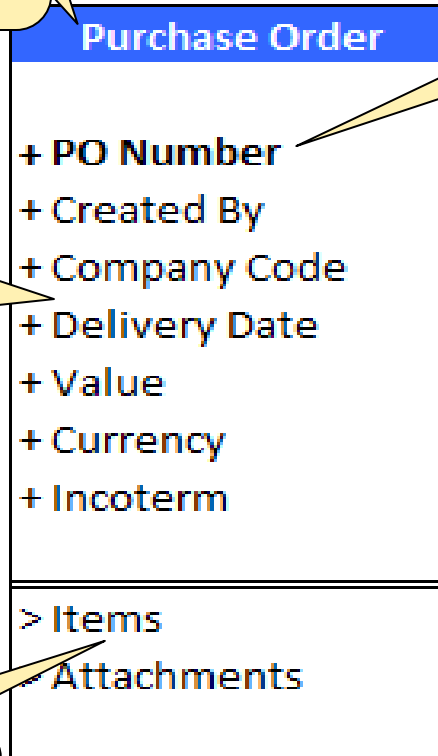
6.Consume Service ✓ (OData Consumer)

Entity Data Model

Entity Type :
- Basic Building Block
- **Represents business object**

At least one property must be nominated as key field.

Purchase Order

+ PO Number
+ Created By
+ Company Code
+ Delivery Date
+ Value
+ Currency
+ Incoterm

> Items
> Attachments

Entity Type is built up of Properties.

Item

+ PO Number
+ Item Number
+ Material No
+ Description
+ Quantity
+ Unit
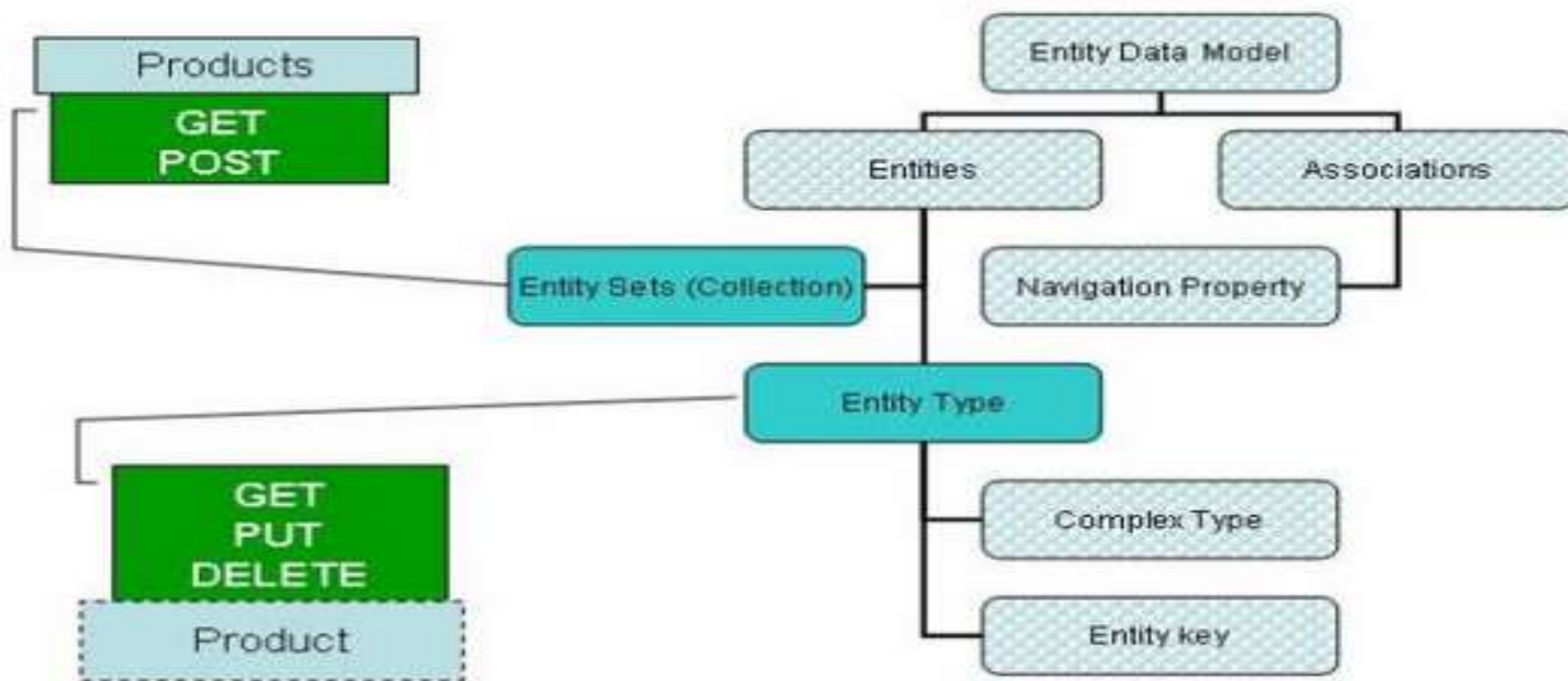+ Price Per Unit

Navigation :
Runtime implementation of an association.

Association : Declares that a relationship exists between two entity types

Possibility operations

- We can do a GET and POST request on an Entity Set while GET, PUT and DELETE can be done on an Entity.

CRUDQ Operations

❖ **Create**
- ▪ Used to create a one or more record in the backend system
- ▪ Http method POST will be used to perform create operation

❖ **Get Entity (Read)**
- ▪ Used to populate only one record in the output response
- ▪ Http method GET will be used to perform read operation

❖ **Update**
- ▪ Used to update a record in the backend system
- ▪ Http method PUT will be used to perform update operation

❖ **Delete**
- ▪ Used to delete a record in the backend system
- ▪ Http method DELETE will be used to perform delete operation

❖ **Get Entity Set (Query)**
- ▪ Used to populate one or more record in the output response
- ▪ Http method GET will be used to perform query operation

Service Operations

Service Operations:

- **G**etEntity

  we can refer to as work area in ABAP

- **G**etEntitySet

  we can refer to as internal table in ABAP

- **C**reate

  We can create only single record at once.

- **U**pdate

  We can single only record at once.

- **D**elete

  We can delete only single record at once.

Demo 1 : OData service with CRUDQ operations

❖ Scenario:

▪ Table : ZEMP_GW

▪ FMs in the backend system :

- ZCREATE_EMP_GW (Insert employee details)
- ZDELETE_EMP_GW (Delete employee details)
- ZQUERY_EMP_GW (Get employee list)
- ZUPDATE_EMP_GW (Update employee details)
- ZREAD_EMP_GW (Read employee details)

OData Query Options

Command operations used:

- ❖ $filter
- ❖ $top
- ❖ $skip
- ❖ $select
- ❖ $count
- ❖ $expand
- ❖ $inlinecount
- ❖ $orderby

OData Query Options (Continued...)

❖Paging >> $top & $skip

- <u>Service URI:</u> /sap/opu/odata/sap/ZDT_MAPPING_SRV_01/EmployeeSet?$top=5
  - ✓The first 5 employee records are returned

- <u>Service URI:</u> /sap/opu/odata/sap/ZDT_MAPPING_SRV_01/EmployeeSet?$skip=5
  - ✓The first 5 records are skipped and records from number 6 onwards are provided in the response

- <u>Service URI:</u>
  /sap/opu/odata/sap/ZDT_MAPPING_SRV_01/EmployeeSet?$skip=5&$top=5
  - ✓The first 5 records are skipped and the next 5 records are displayed i.e. from records 6 to 10

OData Query Options (Continued…)

❖ Projecting >> $select

- Service URI:
  /sap/opu/odata/sap/ZDT_MAPPING_SRV_01/PoHeaderSet?$select=PoNumber,TargetVal,Currency

  ✓ Only the PO Number, Target Value and Currency are returned in the output response

❖ Filtering >> $filter

- Service URI:
  /sap/opu/odata/sap/ZDT_MAPPING_SRV_01/PoHeaderSet?$filter=CoCode eq ' 1000'

  ✓ All the records with Company code equal to '1000' are returned in the output response

❖ Count >> $count

- Service URI: /sap/opu/odata/sap/ZDT_MAPPING_SRV_01/PoHeaderSet/$count

  ✓ Returns the number of records in the output response

ANY QUESTIONS