DEEP\_ENTITY in SAP Gateway

In some scenarios only updates of hierarchical data are allowed. For example, a Sales Order and a Sales Order Item can only be created together and at the same time, not independently on their own. For this OData Channel provides **deep insert** functionality: a basic deep insert feature is offered to provide the possibility to the application to create single entities deeply. In addition, deep entities can be created in one activity.

When using deep insert the data need to be nested, that is, a deep structure is expected.

For deep insert interface /IWBEP/IF\_MGW\_APPL\_SRV\_RUNTIME provides method **CREATE\_DEEP\_ENTITY**.

The process is as follows:

1. The SAP Gateway framework extracts the expand expression out of the payload (inlined data).
2. Data is sent to the backend.
3. Method CREATE\_DEEP\_ENTITY is called.
4. The application uses /IWBEP/IF\_MGW\_ODATA\_EXPAND to validate whether the given expand can be handled.

Method **CREATE\_DEEP\_ENTITY**

This method is used for the create operation for an entity - deep insert.

This is then the operation to create an entity with deep data in an inlined format. Every deep insert request has to be handled by the implementation which has to decide whether it can fulfill the current deep insert request or not by a given expand expression.

The method signature has an additional parameter IO\_EXPAND. The SAP NetWeaver Gateway framework resolves the inlined data and translates it to an expand expression which is passed to the method via IO\_EXPAND. The ES\_DATA parameter expects a nested structure which contains the components for the inlined data.

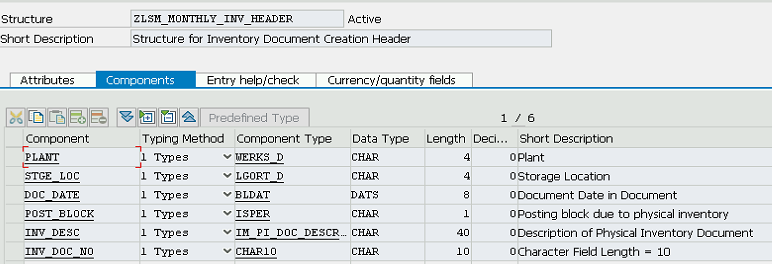
**/IWBEP/IF\_MGW\_ODATA\_EXPAND**

This interface can be used to validate whether an expand expression which can be handled by the application is applicable or not.

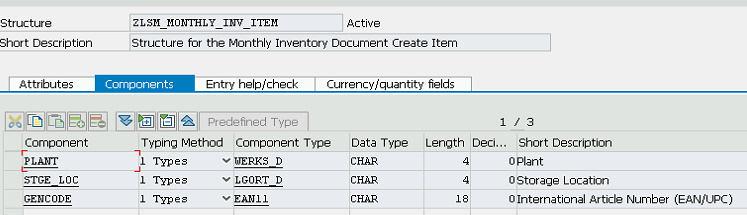
[**Step by Step development for CREATE\_DEEP\_ENTITY operation**](http://scn.sap.com/community/gateway/blog/2014/04/27/step-by-step-development-guide-for-createdeepentity-operation)

To Create a DEEP\_ENTITY we need two structures

Structure-1 – Header



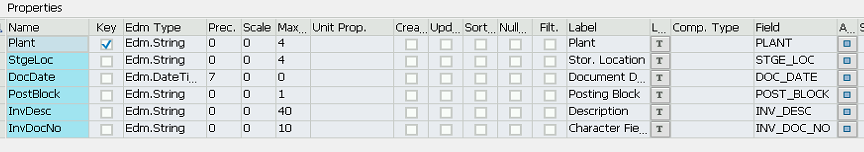
Structure-2 – Item



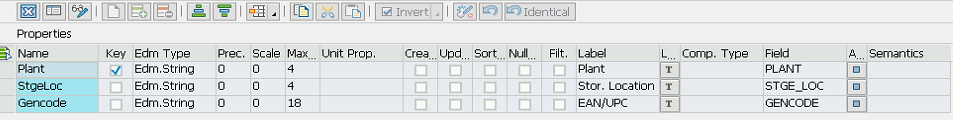
**Create Project in SEGW**

Create two entity types and Entity Sets

Entity type for Header ZINVHEADER



Entity Type for Item ZINVITEM



Create Entity Set for two Entity Type

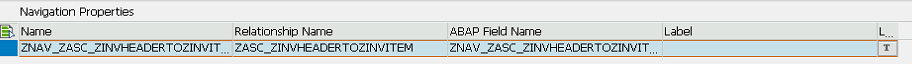
ZINVHEADERDATA and ZINVITEMDATA

Create Association Set between two Entity Type ZINVHEADER and ZINVITEM with Key Fields



Create Navigation given below

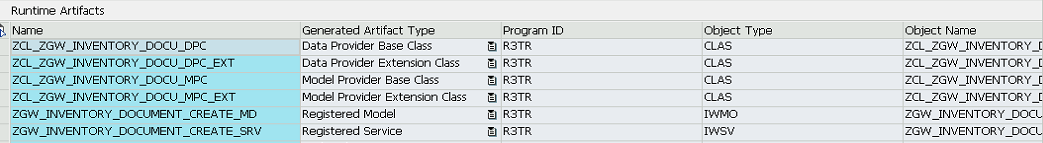
ZINVHEADER to ZINVITEM



Now let’s generate runtime artifacts. Click on generate runtime objects button. It will display

Popup. Keep the default class names as-is and click on enter button.

Once generation is successful, you will get 4 classes. 2 for Data provider and 2 for Model provider.



Once registration done successfully. Goto Gateway Client ( call transaction /IWFND/GW\_CLIENT to open SAP NW Gateway client)

Append $metatda to base service URL and press execute button. If everything is fine then you will HTTP

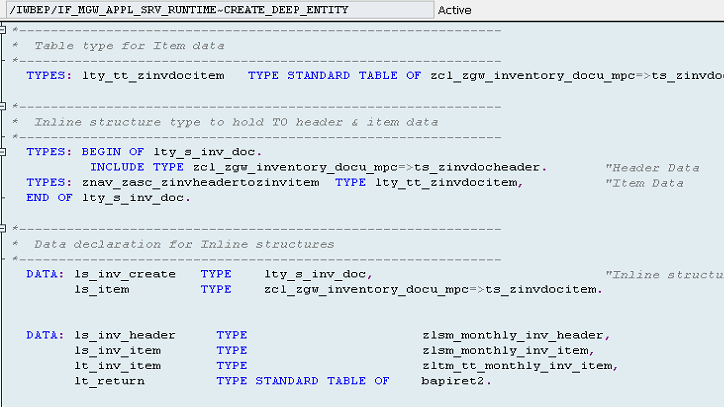
Response as below. Metadata provides information such as Entity type, key property, properties and Entity Set

name and also check service document append ?$format=xml.

Go to ABAP Workbench of Class ZCL\_ZGW\_DISPLAY\_INVENT\_DPC\_EXT class

Redefine the Inherited method /IWBEP/IF\_MGW\_APPL\_SRV\_RUNTIME~CREATE\_DEEP\_ENTITY

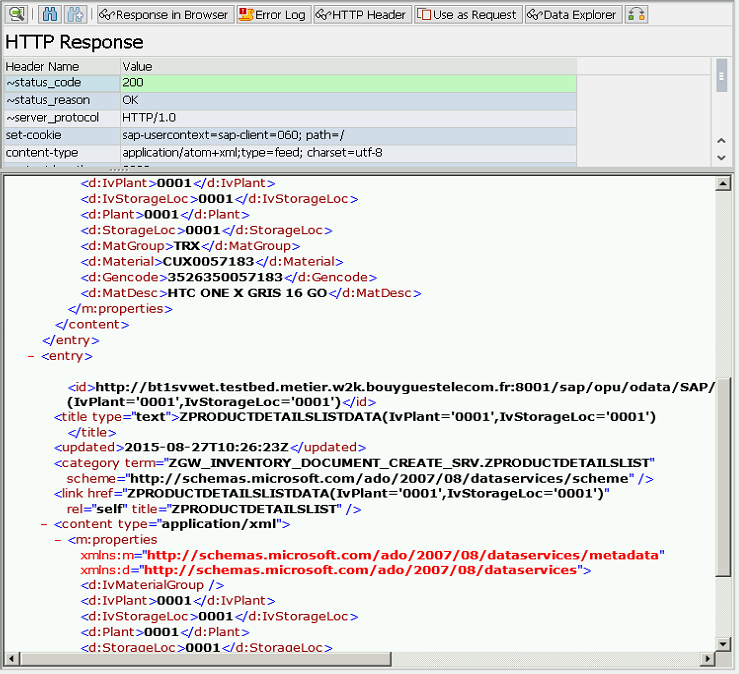
In this Method we have write logic to populate item data with respect to Header data (In this scenario).



Save and Active.

Execute the program.

Link: http://bt1svwet.testbed.metier.w2k.bouyguestelecom.fr:8001/sap/opu/odata/SAP/ZGW\_INVENTORY\_DOCUMENT\_CREATE\_SRV/ZPRODUCTDETAILSLISTDATA?$filter=IvMaterialGroup eq 'TRX,ACC-RCBT' and IvPlant eq '0001' and IvStorageLoc eq '0001'



Gateway Create Entity:

Link: /sap/opu/odata/SAP/ZGW\_INVENTORY\_DOCUMENT\_CREATE\_SRV/ZINVDOCHEADERDATA

