**Handling Complex logic in CDS views:**

**Option -1: Table function**

**DHL -011 :**

**Table Function :** /RB4H/CO\_TF\_CIV\_IHP\_COST\_SHEET

**Class name** : /rb4h/co\_civ\_cl\_cost\_sheet

@EndUserText.label: 'CIVS Table function : IHP Costing Sheet'

@AccessControl.authorizationCheck: #NOT\_REQUIRED

**define** **table** **function** /RB4H/CO\_TF\_CIV\_IHP\_COST\_SHEET

**returns**

**{**

**key** mandt **:** *abap***.***clnt***;**

**key** bukrs **:** *bukrs***;**

**key** rldnr **:** *fins\_ledger***;**

**key** posnr **:** */rb4h/co\_civ\_item1***;**

prio **:** *abap***.***char***(** 1 **);**

numbr **:** */rb4h/co\_civ\_numbc***;**

VERSN **:** *versn\_011***;**

ERGSL **:** *ergsl***;**

KHINR **:** *phinr***;**

LUMPSUM\_PERC **:** */rb4h/co\_civ\_lumpsum\_perc***;**

**}**

**implemented** **by** **method**

/RB4H/CO\_CIV\_CL\_COST\_SHEET**=>**GET\_COST\_SHEET**;**

**Class definition :**

CLASS /rb4h/co\_civ\_cl\_cost\_sheet DEFINITION

PUBLIC

FINAL

CREATE PUBLIC .

PUBLIC SECTION.

INTERFACES : if\_amdp\_marker\_hdb.

CLASS-METHODS : get\_cost\_sheet FOR TABLE FUNCTION /rb4h/co\_tf\_civ\_ihp\_cost\_sheet.

ENDCLASS.

CLASS /rb4h/co\_civ\_cl\_cost\_sheet IMPLEMENTATION.

METHOD get\_cost\_sheet BY DATABASE FUNCTION

FOR HDB LANGUAGE SQLSCRIPT

USING /rb4h/co\_civid1 /rb4h/co\_civ\_ccs.

lt\_output = SELECT

id1.mandt,

ccs.rldnr,

ccs.bukrs,

id1.posnr,

case when id1.bukrs = ccs.bukrs and id1.rldnr = ccs.rldnr then 1 else 2 end as prio,

id1.numbr,

id1.versn,

id1.ergsl,

id1.khinr,

id1.lumpsum\_perc

from "/RB4H/CO\_CIVID1" as id1

cross join "/RB4H/CO\_CIV\_CCS" as ccs ;

\*Apply Min() function to get the lowest number '1' from priority

\* here in this case - Prio = 1 will be picked by Min () function

lt\_return = select mandt,rldnr,bukrs,posnr,

min (prio)as prio,

numbr,versn,ergsl,khinr,lumpsum\_perc from :lt\_output

GROUP BY mandt,rldnr,bukrs,posnr,numbr,versn,ergsl,khinr,lumpsum\_perc;

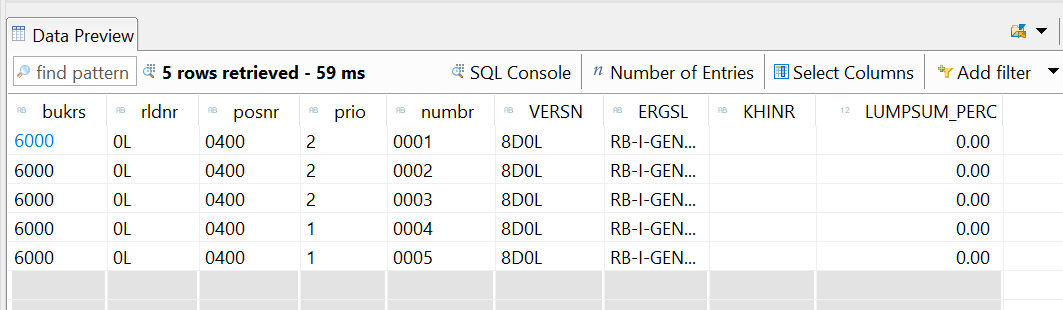
RETURN SELECT mandt,bukrs,rldnr,posnr,prio,

numbr,versn,ergsl,khinr,lumpsum\_perc from :lt\_return;

ENDMETHOD.

ENDCLASS.

Output: Giving Priority = 1 for rows maintained by user.



**NOTE: Table function cannot be consumed directly, we need to create CDS View on top it .**

**Option -2: Virtual elements ( Virtual element is now our Prio field) and logic will be done in ABAP class.**

@AbapCatalog.sqlViewName: 'ZIHP\_VIEW'

@AbapCatalog.compiler.compareFilter: true

@AccessControl.authorizationCheck: #NOT\_REQUIRED

@VDM.viewType: #CONSUMPTION

**@OData.publish: true**

@EndUserText.label: 'IHP Cost sheeet'

**define** **view** ZV\_CDS\_IHP **as** **select** **from** /rb4h/co\_civid1 **as** id1

**cross** **join** /rb4h/co\_civ\_ccs **as** ccs

**{**

**key** ccs**.**rldnr**,**

**key** ccs**.**bukrs**,**

**key** id1**.**posnr**,**

@ObjectModel:{ readOnly: true , virtualElement: true, virtualElementCalculatedBy: 'ABAP:ZCL\_IHP\_COST' }

**cast(** 1 **as** *abap***.***int1* **)** **as** prio**,**

//case when id1.bukrs = ccs.bukrs and id1.rldnr = ccs.rldnr then 1 else 2 end as prio,

id1**.**numbr**,**

id1**.**versn**,**

id1**.**ergsl**,**

id1**.**khinr**,**

id1**.**lumpsum\_perc

**}**

**Virtual Element logic is implemented in ABAP class and methods using SADL framework.**

**Class definition:**

class ZCL\_IHP\_COST definition

public

final

create public .

public section.

interfaces IF\_SADL\_EXIT .

interfaces IF\_SADL\_EXIT\_CALC\_ELEMENT\_READ .

protected section.

private section.

ENDCLASS.

CLASS ZCL\_IHP\_COST IMPLEMENTATION.

method IF\_SADL\_EXIT\_CALC\_ELEMENT\_READ~CALCULATE.

DATA: lt\_output  TYPE TABLE OF zv\_cds\_ihp.  
  
    IF it\_original\_data IS NOT INITIAL.  
      lt\_output = CORRESPONDING #( it\_original\_data ).  
  
  
  
      SELECT \* FROM /rb4h/co\_civid1 INTO TABLE @DATA(lt\_id1).  
      SORT lt\_id1 BY rldnr bukrs posnr numbr.  
      LOOP AT lt\_output ASSIGNING FIELD-SYMBOL(<lfs\_out>).  
        READ TABLE lt\_id1  TRANSPORTING NO FIELDS WITH KEY rldnr = <lfs\_out>-rldnr  
                                   bukrs = <lfs\_out>-bukrs  
                                   posnr = <lfs\_out>-posnr  
                                   numbr = <lfs\_out>-numbr.  
        IF sy-subrc = 0.  
          <lfs\_out>-prio = '1'.  
        ELSE.  
          <lfs\_out>-prio = '2'.  
        ENDIF.  
      ENDLOOP.  
  
      MOVE-CORRESPONDING lt\_output TO ct\_calculated\_data.  
  
  
  
    ENDIF.

endmethod.

method IF\_SADL\_EXIT\_CALC\_ELEMENT\_READ~GET\_CALCULATION\_INFO.

endmethod.

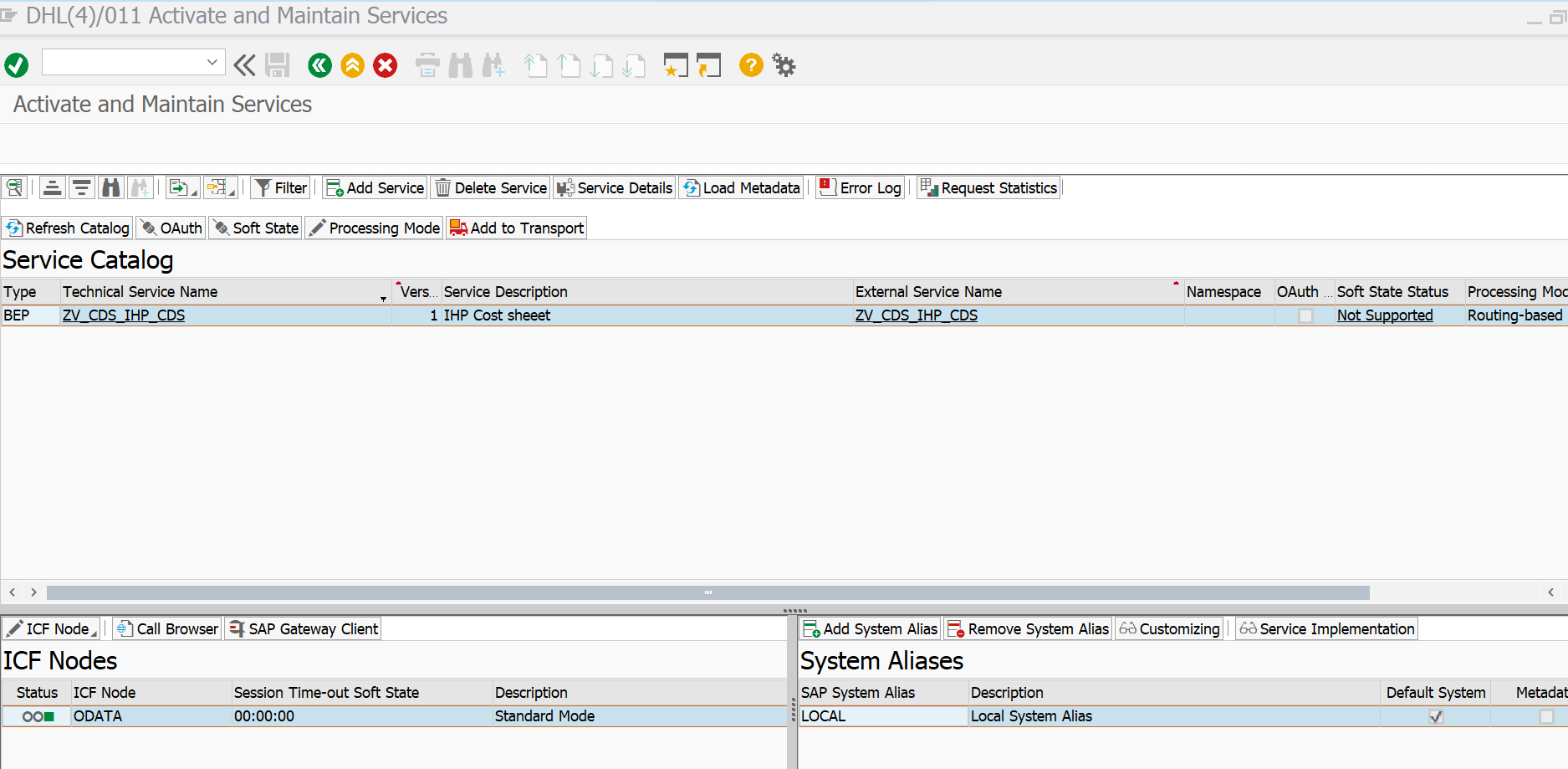
ENDCLASS.

**How To consume Virtual elements ?:**

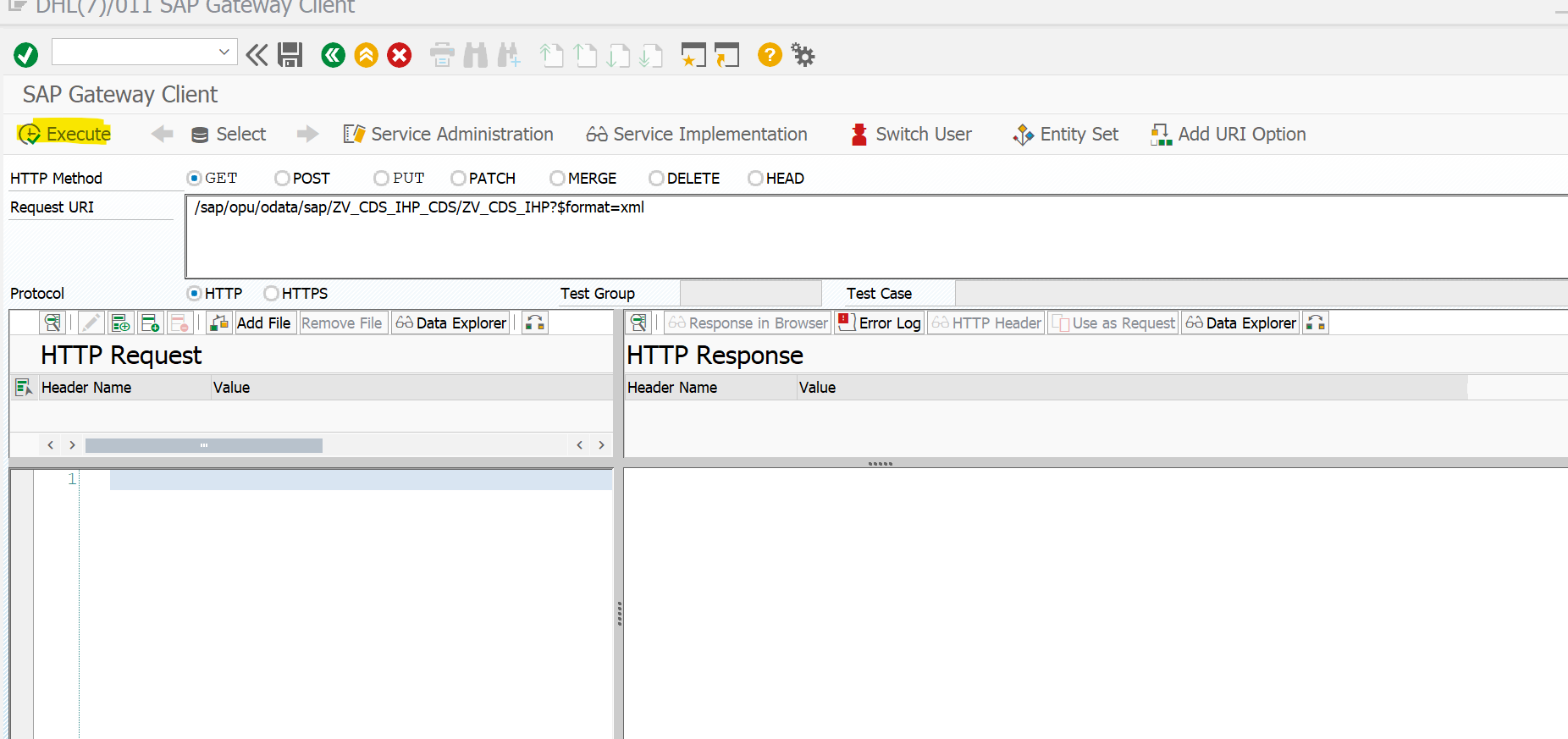
**Call odata call for this CDS entity -** ZV\_CDS\_IHP ( because of SADL interface , so the ABAP logic will be triggered only when Odata call)

Odata service name: ZV\_CDS\_IHP\_CDS

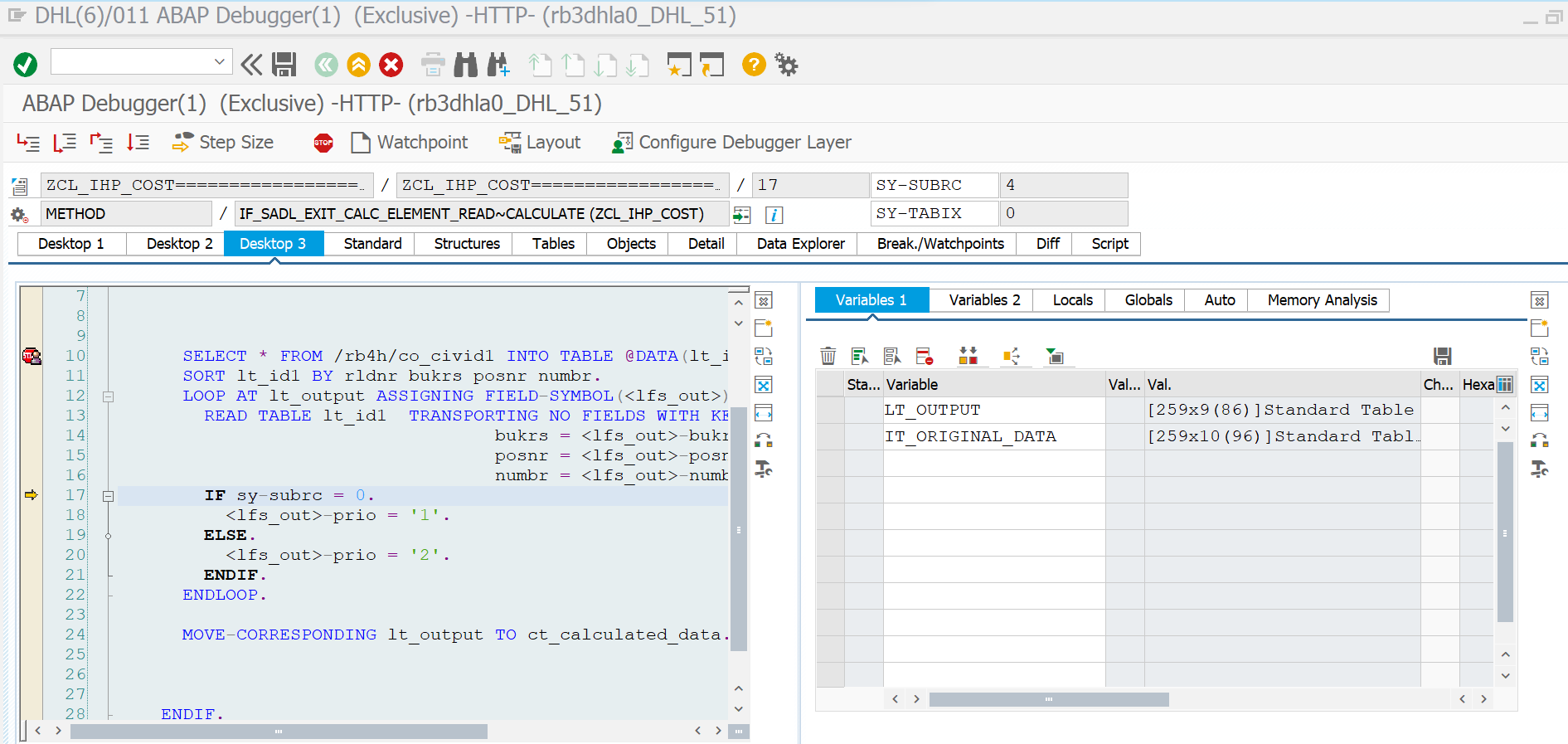
( as we have [annotation@Odata.Publish](mailto:annotation@Odata.Publish) : true)



**Gateway client :**testing Odata service



**Get Call -> will trigger backend ABAP method to apply logic for the virtual element field (Prio )before giving output response.**



Output response : 200 ok

