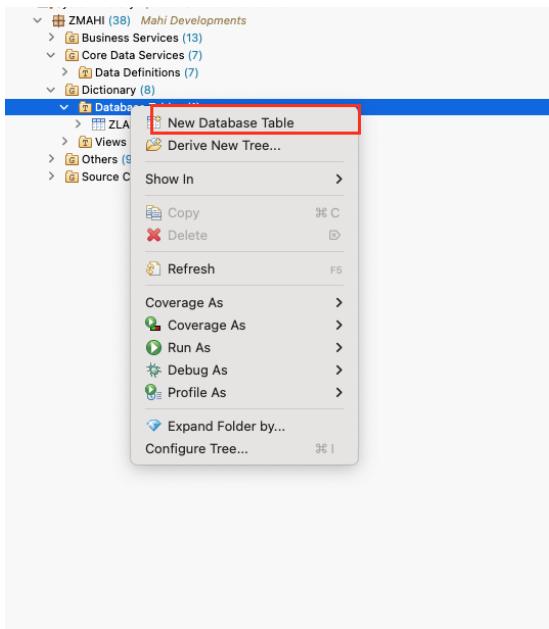


Table of Contents

Built with SAP Fiori Elements + RAP.....	2
Create DB Table.....	2
DB Table Fields and Datatypes	2
Create Class to insert sample records!.....	3
Activate and execute the class	6
Create Interview for the DB table	7
Create CDS View with required annotation.....	8
Activate and Execute CDS view	9
Display the CDS View	10
Create Consume View.....	10
Create Text Table	15
Execute CDS View	22
Analytics CDS View	25
Activate Class and execute the class.....	36
Create Item Table	39
Create Class for Item Table to insert records	40
Create CDS View for Item Table	42
Metadata Extension for Item table	42
Preview and Final Output for List View.....	44
Final Output for Object Page.....	44
Final Output for Value Help	44

Built with SAP Fiori Elements + RAP

Create DB Table



DB Table Fields and Datatypes

```
@EndUserText.label : 'Warehouse Movement Table'  
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE  
@AbapCatalog.tableCategory : #TRANSPARENT  
@AbapCatalog.deliveryClass : #A  
@AbapCatalog.dataMaintenance : #RESTRICTED  
define table zwh_movements {  
  
    key movement_id : sysuuid_x not null;  
    material_id : matnr;  
    batch_number : char10;  
    from_location : werks_d;  
    to_location : werks_d;  
    movement_type : char3;  
    movement_date : dats;  
    @Semantics.quantity.unitOfMeasure : 'zwh_movements.uom'  
    quantity : menge_d;  
    uom : msehi;  
    created_by : syuname;  
  
}
```

```

1 @EndUserText.label : 'Warehouse Movement Table'
2 @AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #RESTRICTED
6 define table zwh_movements {
7
8   key movement_id : sysuuid_x not null;
9   material_id : matnr;
10  batch_number : char10;
11  from_location : werks_d;
12  to_location : werks_d;
13  movement_type : char3;
14  movement_date : dats;
15  @Semantics.quantity.unitOfMeasure : 'zwh_movements.uom'
16  quantity : menge_d;
17  uom : msneh1;
18  created_by : syuname;
19
20 }

```

Create Class to insert sample records!

```

CLASS zcl_wh_movement_loader DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC.

  PUBLIC SECTION.
    INTERFACES if_oo_adt_classrun.
ENDCLASS.

```

CLASS zcl_wh_movement_loader IMPLEMENTATION.

METHOD if_oo_adt_classrun~main.

```

DATA: lt_movements TYPE STANDARD TABLE OF zwh_movements,
      ls_movement TYPE zwh_movements.
DATA: lt_materials TYPE STANDARD TABLE OF char10 WITH DEFAULT KEY,
      lt_types     TYPE STANDARD TABLE OF char3 WITH DEFAULT KEY,
      lt_uom       TYPE STANDARD TABLE OF char3 WITH DEFAULT KEY,
      lt_plants    TYPE STANDARD TABLE OF char10 WITH DEFAULT KEY.

```

```

lt_materials = VALUE #(( 'MAT1001') ( 'MAT1002') ( 'MAT1003') ( 'MAT1004')).
lt_types    = VALUE #(( '101') ( '261') ( '311') ( '122')).
lt_uom      = VALUE #(( 'KG') ( 'EA') ( 'L')).
lt_plants   = VALUE #(( 'PL1') ( 'PL2') ( 'PL3') ( 'PL4')).
DATA: lv_material TYPE char10,
      lv_type     TYPE char3,
      lv_uom      TYPE char3,
      lv_from     TYPE char10,
      lv_to       TYPE char10,
      lv_date     TYPE sy-datum,
      lv_batch    TYPE char10.

```

```

DELETE FROM zwh_movements.
out->write( 'Old records deleted.' && cl_abap_char_utilities=>newline ).
```

```

DO 10000 TIMES.
lv_material = lt_materials[ sy-index MOD lines( lt_materials ) + 1 ].
lv_type   = lt_types[ sy-index MOD lines( lt_types ) + 1 ].
lv_uom    = lt_uom[ sy-index MOD lines( lt_uom ) + 1 ].
lv_from   = lt_plants[ sy-index MOD lines( lt_plants ) + 1 ].
lv_to     = lt_plants[ ( sy-index + 1 ) MOD lines( lt_plants ) + 1 ].
lv_date   = sy-datum - ( sy-index MOD 30 ).
lv_batch  = |BATCH{ sy-index PAD = '0' }|.

CLEAR ls_movement.
ls_movement-movement_id = cl_system_uuid->create_uuid_x16_static( ).
ls_movement-material_id = lv_material.
ls_movement-batch_number = lv_batch.
ls_movement-from_location = lv_from.
ls_movement-to_location = lv_to.
ls_movement-movement_type = lv_type.
ls_movement-movement_date = lv_date.
ls_movement-quantity     = sy-index MOD 100 + 1.
ls_movement-uom          = lv_uom.

APPEND ls_movement TO lt_movements.

" Insert in batches of 1000
IF lines( lt_movements ) >= 1000.
  INSERT zwh_movements FROM TABLE @lt_movements.
  CLEAR lt_movements.
ENDIF.
ENDDO.

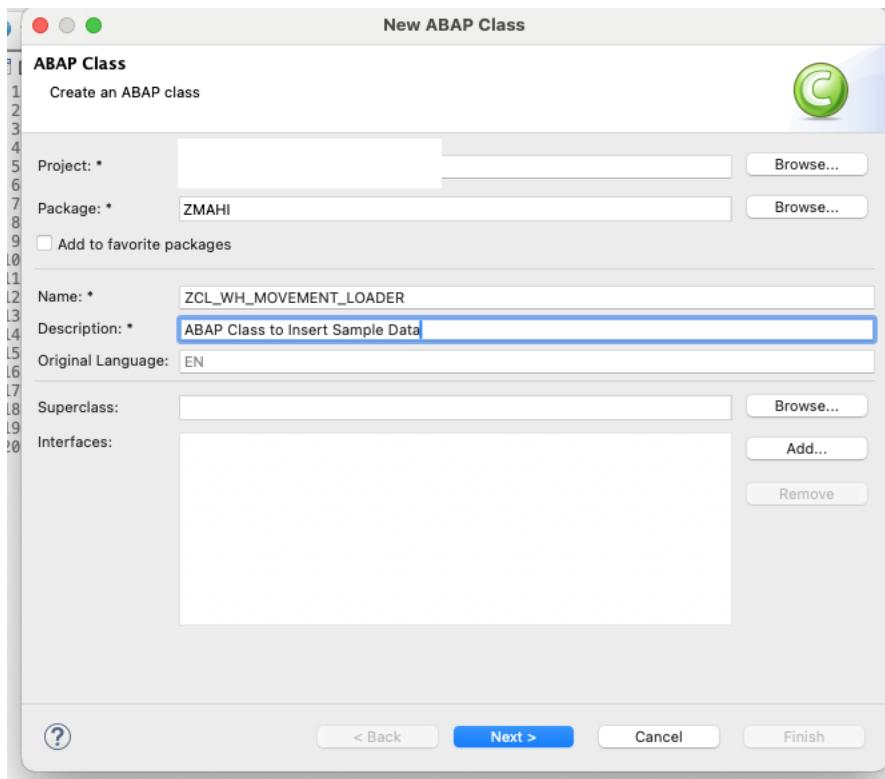
" Insert remaining
IF lt_movements IS NOT INITIAL.
  INSERT zwh_movements FROM TABLE @lt_movements.
ENDIF.

out->write( ' 10,000 test records inserted into ZWH_MOVEMENTS.' ).

ENDMETHOD.

ENDCLASS.

```



Activate and execute the class

```

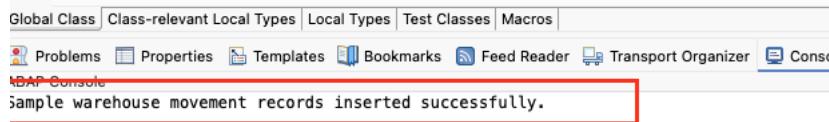
G ZCL_WH_MOVEMENT_LOADER ► ● IF_OO_ADT_CLASSRUN-MAIN
1@ CLASS zcl_wh_movement_loader DEFINITION
2 PUBLIC
3 FINAL
4 CREATE PUBLIC.
5
6 PUBLIC SECTION.
7   INTERFACES if_oo_adt_classrun.
8 ENDCLASS.
9
10@ CLASS zcl_wh_movement_loader IMPLEMENTATION.
11
12@ METHOD if_oo_adt_classrun~main.
13   DATA lt_movements TYPE STANDARD TABLE OF zwh_movements.
14   DATA ls_movement TYPE zwh_movements.
15
16   CLEAR ls_movement.
17   ls_movement-movement_id = cl_system_uuid->create_uuid_x16_static( ).
18   ls_movement-material_id = 'MAT1001'.
19   ls_movement-batch_number = 'BATCH01'.
20   ls_movement-from_location = 'PLANT1'.
21   ls_movement-to_location = 'PLANT2'.
22   ls_movement-movement_type = '101'.
23   ls_movement-movement_date = sy-datum.
24   ls_movement-quantity = '10.500'.
25   ls_movement-uom = 'KG'.
26   ls_movement-created_by = sy-uname.
27   APPEND ls_movement TO lt_movements.
28
29   CLEAR ls_movement.
30   ls_movement-movement_id = cl_system_uuid->create_uuid_x16_static( ).
31   ls_movement-material_id = 'MAT1002'.
32   ls_movement-batch_number = 'BATCH02'.
33   ls_movement-from_location = 'PLANT2'.
34   ls_movement-to_location = 'PLANT3'.
35   ls_movement-movement_type = '261'.
36   ls_movement-movement_date = sy-datum - 1.
37   ls_movement-quantity = '5.000'.
38   ls_movement-uom = 'EA'.
39   ls_movement-created_by = sy-uname.
40   APPEND ls_movement TO lt_movements.
41
42   " Insert into DB
43   INSERT zwh_movements FROM TABLE lt_movements.
44
45 System Library (press F5 to load full content)
  ▾ ZMABH (40) Main Developments
    ▾ Business Services (13)
      ▾ ZWH_MOVEMENT_LOADER (1)
        ▾ Data Definitions (7)
          ▾ Dictionary (9)
            ▾ Tables (3)
              ▾ ZLAB_DATA_SOURCE Lab KPI Data Source
              ▾ ZWH_MOVEMENTS Warehouse Movement Table
              ▾ Views (7)
            ▾ Data (1)
          ▾ Source Code Library (press F5 to load full content) (2)
            ▾ Classes (press F5 to load full contents)
              ▾ ZCL_WH_MOVEMENT_LOADER_ABAP
                ▾ New ABAP Class
                  ▾ Duplicate...
                  ▾ Open
                  ▾ Open in Project
                  ▾ Open With
                  ▾ Show In
                  ▾ Copy
                  ▾ Delete
                  ▾ Add to Favorite Objects
                  ▾ Get Where-Used List
                  ▾ Search Content...
                  ▾ Rename...
                  ▾ Export...
                  ▾ Refresh
                  ▾ Share Link...
                  ▾ API State
                  ▾ Activate
                  ▾ Smooch
                  ▾ Change Package Assignment...
                  ▾ Coverage As
                  ▾ Run As
                  ▾ Debug As
                  ▾ Profile As
                  ▾ Add Bookmark...
                  ▾ Compare With
                ▾ Insert into DB
                  ▾ NSE: zwh_movements FROM TABLE lt_movements.
                    ▾ sy-subrc.
                      ▾ out->write('Sample warehouse movement records inserted successfully.').
                      ▾ out->write('Error inserting records.').
                      ▾ NOTE.
                        ▾ METHOD.
                          ▾ ASS.
                ▾ Run Configurations...
                ▾ 1 ABAP Application
                ▾ 2 ABAP Application (Console) (selected)
                ▾ 3 ABAP Test Cockpit
                ▾ 4 ABAP Test Cockpit With...
                ▾ 5 ABAP Unit Test
                ▾ 6 ABAP Unit Test With...
                ▾ 7 Integration Forecast...
                ▾ 8 ABAP Package Check
                ▾ Run Configurations...

```

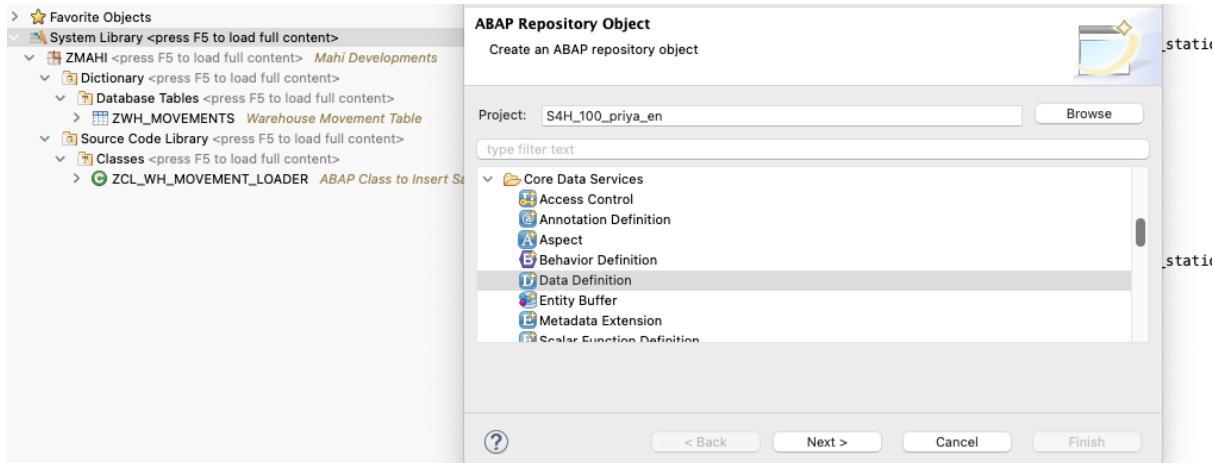
```

17 ls_movement-movement_id = cl_system_uuid->create_uuid_x16_static( ).
18 ls_movement-material_id = 'MAT1001'.
19 ls_movement-batch_number = 'BATCH01'.
20 ls_movement-from_location = 'PLANT1'.
21 ls_movement-to_location = 'PLANT2'.
22 ls_movement-movement_type = '101'.
23 ls_movement-movement_date = sy-datum.
24 ls_movement-quantity = '10.500'.
25 ls_movement-uom = 'KG'.
26 ls_movement-created_by = sy-uname.
27 APPEND ls_movement TO lt_movements.
28
29 CLEAR ls_movement.
30 ls_movement-movement_id = cl_system_uuid->create_uuid_x16_static( ).
31 ls_movement-material_id = 'MAT1002'.
32 ls_movement-batch_number = 'BATCH02'.
33 ls_movement-from_location = 'PLANT2'.
34 ls_movement-to_location = 'PLANT3'.
35 ls_movement-movement_type = '261'.
36 ls_movement-movement_date = sy-datum - 1.
37 ls_movement-quantity = '5.000'.
38 ls_movement-uom = 'EA'.
39 ls_movement-created_by = sy-uname.
40 APPEND ls_movement TO lt_movements.
41
42 " Insert into DB
43 INSERT zwh_movements FROM TABLE lt_movements.
44
45 IF sy-subrc = 0.
46   out->write( 'Sample warehouse movement records inserted successfully.' ).
47 ELSE.
48   out->write( 'Error inserting records.' ).
49 ENDIF.
50
51 ENDMETHOD.
52
53 ENDMETHOD.

```



Create Interview for the DB table



```

@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Warehouse Movement Interface View'
define root view entity ZI_WH_MOVEMENTS
  as select from zwh_movements
{
  key movement_id,
  material_id,
  batch_number,
  from_location,
  to_location,
  movement_type,
  movement_date,
  @Semantics.quantity.unitOfMeasure: 'uom'
  quantity,
  uom,
  created_by
}

```

Create CDS View with required annotation.

```

@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Warehouse Movement Interface View'
@OData.entitySet.name: 'WHMovements'
define root view entity ZI_WH_MOVEMENTS
  as select from zwh_movements

composition [0..*] of ZI_WH_MOV_ITEMS as _Items

association [0..1] to ZI_MOVTYPING as _MovementType
  on $projection.movement_type = _MovementType.movement_type

association [0..*] to ZI_WH_MOVEMENT_ANALYTICS as _Analytics
  on $projection.movement_type = _Analytics.movement_type
{
  @EndUserText.label: 'Movement ID'
  key movement_id,

  @EndUserText.label: 'Material'
  material_id,

  @EndUserText.label: 'Batch'
  batch_number,

  @EndUserText.label: 'From Location'
  from_location,

  @EndUserText.label: 'To Location'
  to_location,

  @EndUserText.label: 'Movement Type'
  @Consumption.valueHelpDefinition: [
    entity: { name: 'ZI_MOVTYPING', element: 'movement_type' }
  ]
}

```

```

@ObjectModel.text.element: ['MovementTypeText']
movement_type,
_movementType.description as MovementTypeText,
case movement_type
when '101' then 3 // Goods Receipt → Green
when '261' then 1 // Goods Issue → Red
when '311' then 2 // Transfer Posting → Yellow
else 0           // Unknown → Grey
end as movement_type_crit,
@EndUserText.label: 'Movement Date'
movement_date,
@EndUserText.label: 'Quantity'
@Semantics.quantity.unitOfMeasure: 'uom'
quantity,
@EndUserText.label: 'UoM'
uom,
_items,
_analytics
}

```

Activate and Execute CDS view

```

11     from_location,
12
13     to_location,
14
15     movement_type,
16
17     movement_date,
18
19     quantity.unitOfMeasure: 'uom'
20
21     ty,
22
23     d_by

```

The screenshot shows the SAP ABAP IDE interface. A context menu is open over a CDS View named "ZL_WH_MOVEMENTS". The menu path is "Data Definitions > ZL_WH_MOVEMENTS". The menu items include:

- New Data Definition
- New Access Control
- New Metadata Extension
- New Service Definition
- New ABAP Test Class
- New Customer Data Browser Object
- New Behavior Definition
- Duplicate...
- Generate ABAP Repository Objects...
- New Knowledge Transfer Document

Below the menu, there are standard file operations: Open, Open in Project, Open With, Show In, Copy, Delete, Add to Favorite Objects, Get Where-Used List, Refresh, Share Link..., API State, Migrate to CDS View Entity..., Activate, Unlock, Change Package Assignment..., Coverage As, Coverage As, Run As, Debug As, Profile As, Add Bookmark..., Compare With.

In the bottom right corner, a tooltip displays run configurations:

- 1 ABAP Application F8
- 2 ABAP Test Cockpit ⌘⌘ F2
- 3 ABAP Test Cockpit With... ⌘⌘ F10
- 4 ABAP Unit Test ⌘⌘ F12
- 5 ABAP Unit Test With... ⌘⌘ F12
- 6 Integration Forecast...
- 7 ABAP Package Check ⌘⌘ F5

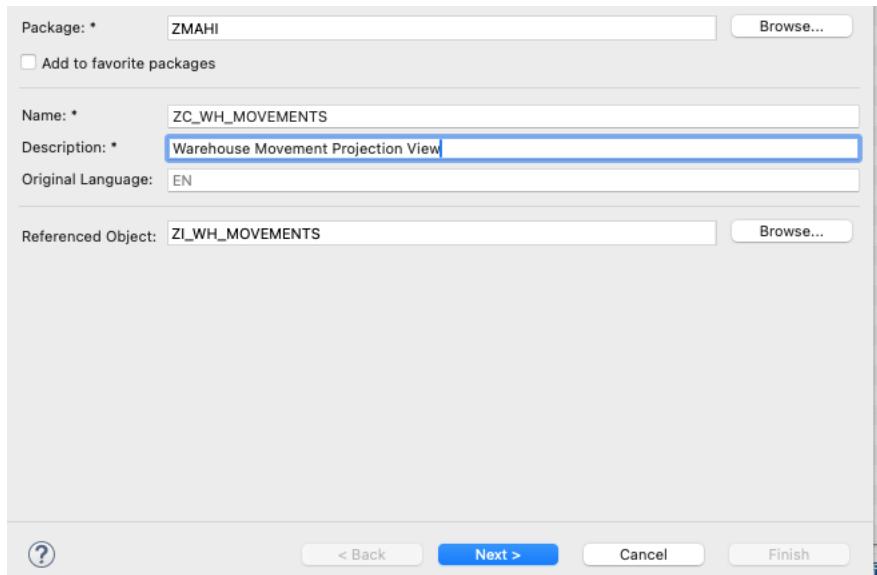
Display the CDS View

movement_id	material_id	batch_number	from_location	to_location	movement_type	movement_date	quantity	uom
2D55B4144ABE	MAT1001	BATCH01	PLAN	PLAN	101	2025-05-31	10.500	KG
2D55B4144ABE1	MAT1002	BATCH02	PLAN	PLAN	261	2025-05-30	5.000	EA

Create Consume View

The screenshot shows the SAP ABAP IDE interface. A context menu is open over a CDS View named "ZL_WH_MOV". The menu path is "Data Definitions > ZL_WH_MOV". The menu items include:

- New Data Definition
- New Access Control
- New Metadata Extension
- New Service Definition
- New ABAP Test Class
- New Customer Data Browser Object
- New Behavior Definition
- Duplicate...
- Generate ABAP Repository Objects...



```
@Search.searchable: true
@ObjectModel.semanticKey: ['movement_id']
@EndUserText.label: 'Warehouse Movement Projection View'
@Metadata.allowExtensions: true
@UI.headerInfo: {
  typeName: 'Movement',
  typeNamePlural: 'Movements',
  title: {
    type: #STANDARD,
    value: 'material_id'
  },
  description: {
    value: 'movement_id'
  }
}
define root view entity ZC_WH_MOVEMENTS
  as projection on ZI_WH_MOVEMENTS
  association [0..1] to ZI_WH_MOVEMENT_ANALYTICS as _Analytics
  on $projection.movement_type = _Analytics.movement_type

{
  key movement_id,
  @Search.defaultSearchElement: true
  material_id,
  batch_number,
  from_location,
  to_location,
  movement_type,
  MovementTypeText,
  movement_type_crit,
  movement_date,
  quantity,
  uom,
```

```

        _Items,
        _Analytics
    }

1@UI.headerInfo: {
2    typeName: 'Movement',
3    typeNamePlural: 'Movements'
4 }
5 @Search.searchable: true
6 @ObjectModel.semanticKey: ['movement_id']
7 @EndUserText.label: 'Warehouse Movement Projection View'
8 define root view entity ZC_WH_MOVEMENTS
9   as projection on ZI_WH_MOVEMENTS
!0 {
!1   key movement_id,
!2   @Search.defaultSearchElement: true
!3     material_id,
!4     batch_number,
!5     from_location,
!6     to_location,
!7     movement_type,
!8     movement_date,
!9     quantity,
!0     uom,
!1     created_by
!2 }
!3

```

Create Metadata Extension

ZMAHI <press F5 to load full content> *Mahi Developments*

Core Data Services <press F5 to load full content>

Data Definitions <press F5 to load full content>

ZC_WH_MOVEM
ZI_WH_MOVEM

Dictionary <press F5>

Database Tables <press F5>

Source Code Library

Classes <press F5>

New Data Definition

New Access Control

New Metadata Extension

New Service Definition

New ABAP Test Class

New Customer Data Browser Object

New Behavior Definition

Duplicate...

Generate ABAP Repository Objects...

New Knowledge Transfer Document

Open

Open in Project >

Open With >

Show In >

Copy ⌘ C

Delete ⌘ D

Add to Favorite Objects

Get Where-Used List ⌘ G

Refresh F5

Share Link

New Metadata Extension

Metadata Extension

Create a metadata extension

Project: * Browse...

Package: * ZMAHI Browse...

Add to favorite packages

Name: *

Description: *

Original Language: EN

Extended Entity: ZC_WH_MOVEMENTS Browse...

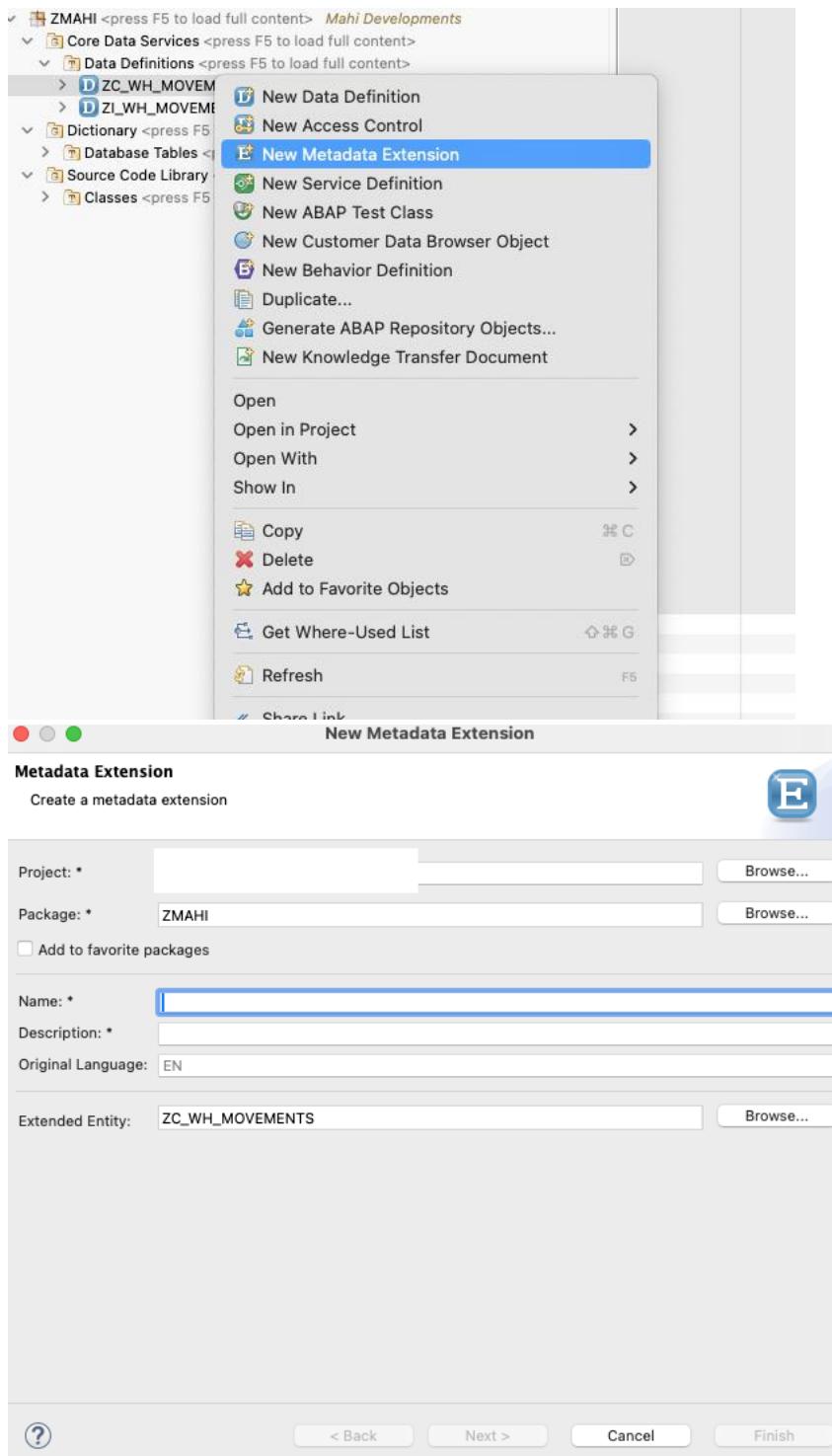
?

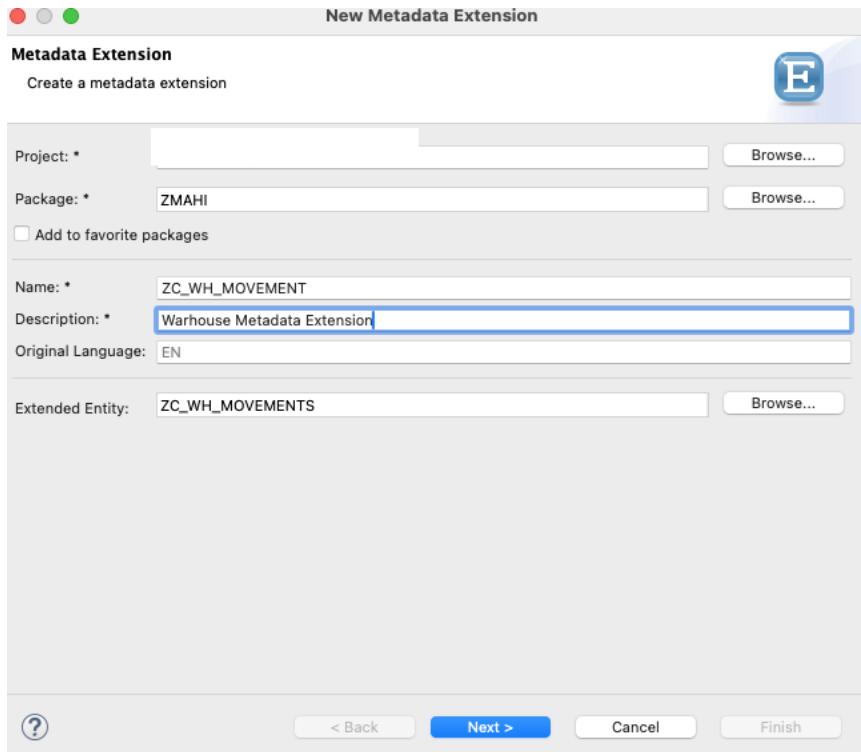
< Back

Next >

Cancel

Finish





```
1@UI.headerInfo: {
2  typeName: 'Movement',
3  typeNamePlural: 'Movements'
4 }
5 @Search.searchable: true
6 @ObjectModel.semanticKey: ['movement_id']
7 @EndUserText.label: 'Warehouse Movement Projection View'
8 @Metadata.allowExtensions: true
9 define root view entity ZC_WH_MOVEMENTS
0  as projection on ZI_WH_MOVEMENTS
1 {
2   key movement_id,
3   @Search.defaultSearchElement: true
4   material_id|
5   batch_number,
6   from_location,
7   to_location,
8   movement_type,
9   movement_date,
0   quantity,
1   uom,
2   created_by
3 }
4
```

```
@Metadata.layer: #CORE
@UI.chart: [
  qualifier: 'MovementChart',
  chartType: #COLUMN,
  dimensions: ['movement_type'],
  measures: ['quantity'],
  dimensionAttributes: [
    { dimension: 'movement_type', role: #CATEGORY }
```

```

        ],
        measureAttributes: [
            { measure: 'quantity', role: '#AXIS_1' }
        ]
    }]
}

@UI.presentationVariant: [{
    qualifier: 'ChartVariant',
    visualizations: [
        { type: '#AS_CHART', qualifier: 'MovementChart' }
    ]
}]
}

annotate view ZC_WH_MOVEMENTS with
{
    // === Facets ===
    @UI.facet: [
        // {
        //     id: 'GeneralInfo',
        //     label: 'General Information',
        //     type: '#COLLECTION',
        //     position: 10
        // },
        // {
        //     id: 'Header',
        //     type: '#IDENTIFICATION_REFERENCE',
        //     label: 'Details',
        //     parentId: 'GeneralInfo',
        //     position: 10
        // },
        {
            id: 'Chart',
            label: 'Movement Chart',
            type: '#CHART_REFERENCE',
            targetElement: 'ChartVariant',
            position: 20
        },
        {
            id: 'ItemsTable',
            label: 'Movement Items',
            type: '#LINEITEM_REFERENCE',
            targetElement: '_Items',
            position: 30
        }
    ]
}

@UI.lineItem: [{ position: 50, criticality: 'movement_type_crit' }]
@UI.selectionField: [{ position: 40 }]
@UI.textArrangement: #TEXT_ONLY
movement_type;

@UI.lineItem: [{ position: 10 }]

```

```

@UI.selectionField: [{ position: 10 }]
material_id;

@UI.lineItem: [{ position: 20 }]
batch_number;

@UI.lineItem: [{ position: 30 }]
@UI.selectionField: [{ position: 20 }]
from_location;

@UI.lineItem: [{ position: 40 }]
@UI.selectionField: [{ position: 30 }]
@UI.textArrangement: #TEXT_ONLY
to_location;

@UI.lineItem: [{ position: 60 }]
@UI.selectionField: [{ position: 50 }]
movement_date;

@UI.lineItem: [{ position: 70 }]
quantity;

@UI.lineItem: [{ position: 80 }]
 uom;
}

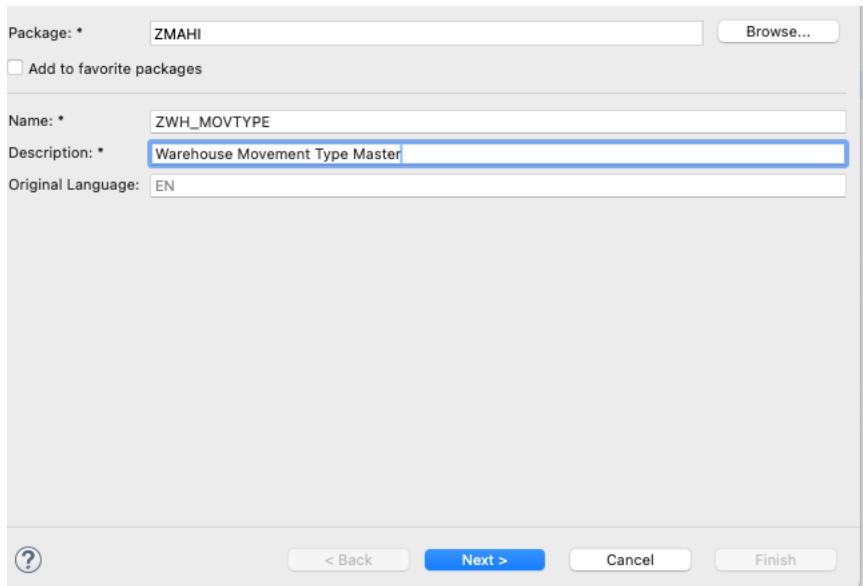
```

Create Text Table

The screenshot shows a dialog box for creating a text table. The fields are as follows:

- Package: * ZMAHI
- Name: * ZT_MOVEMENT_TYPE
- Description: * Movement Type Master Table
- Original Language: EN

At the bottom, there are navigation buttons: ? (Help), < Back, Next >, Cancel, and Finish.



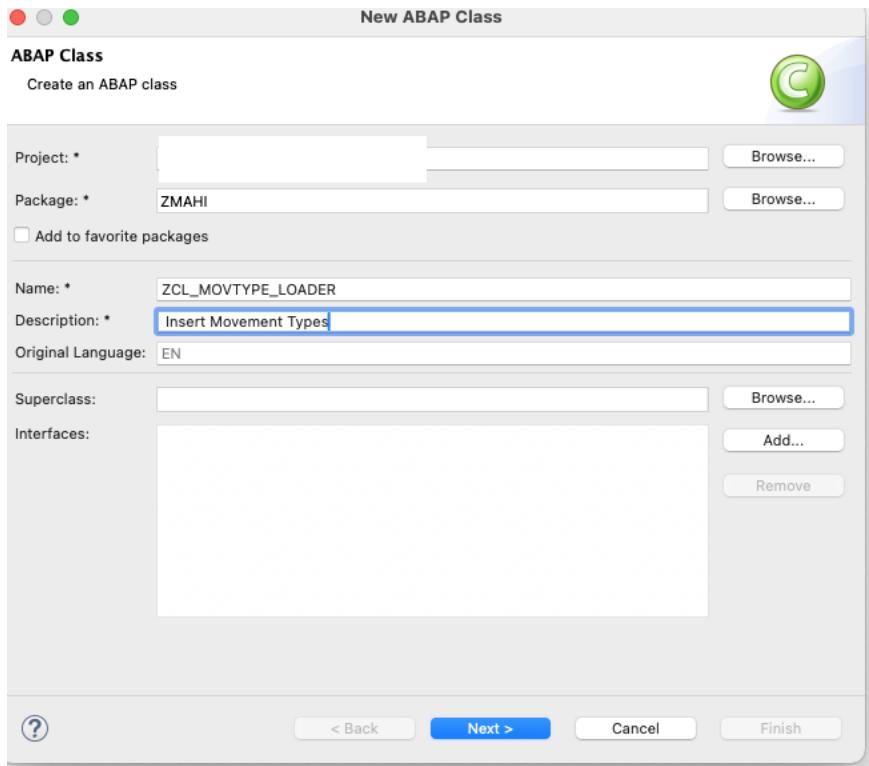
```
1 @EndUserText.label : 'Warehouse Movement Type Master'
2 @AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #RESTRICTED
6 define table zwh_movtype {
7
8   key movement_type : abap.char(3) not null;
9   description      : abap.char(40);
10
11 }
```

```
@EndUserText.label : 'Warehouse Movement Type Master'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table zwh_movtype {

  key movement_type : abap.char(3) not null;
  description      : abap.char(40);

}
```

Create Class for Text Table to Insert records



```

ZCL_MOVTYPE_LOADER ▶
@CLASS zcl_movtype_loader DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC.

  PUBLIC SECTION.
    INTERFACES if_oo_adt_classrun.

  ENDCCLASS.

@CLASS zcl_movtype_loader IMPLEMENTATION.

@ METHOD if_oo_adt_classrun~main.

  DATA: lt_data TYPE STANDARD TABLE OF zwh_movtype,
        ls_data TYPE zwh_movtype.

  " Goods Receipt
  ls_data-movement_type = '101'.
  ls_data-description    = 'Goods Receipt'.
  APPEND ls_data TO lt_data.

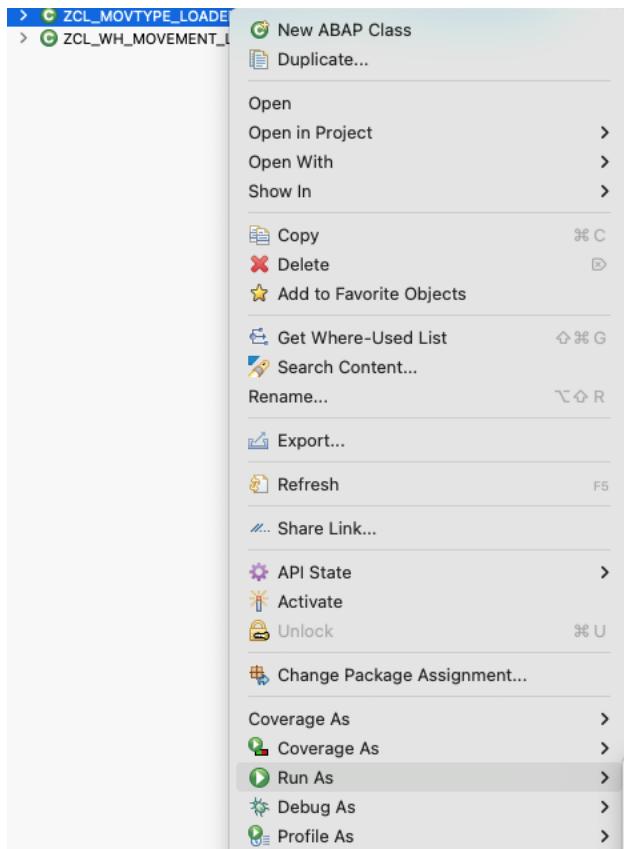
  " Goods Issue
  ls_data-movement_type = '261'.
  ls_data-description    = 'Goods Issue'.
  APPEND ls_data TO lt_data.

  " Transfer Posting
  ls_data-movement_type = '311'.
  ls_data-description    = 'Transfer Posting'.
  APPEND ls_data TO lt_data.

  " Return Delivery
  ls_data-movement_type = '122'.
  ls_data-description    = 'Return Delivery'.
  APPEND ls_data TO lt_data.

  " Insert entries into the table
  DELETE FROM zwh_movtype. " Optional: clean table before insert
  INSERT zwh_movtype FROM TABLE @lt_data.

```



```

ls_data TYPE zwh_movtype

" Goods Receipt
ls_data-movement_type = '101'.
ls_data-description    = 'Goods Receipt'.
APPEND ls_data TO lt_data.

" Goods Issue
ls_data-movement_type = '261'.
ls_data-description    = 'Goods Issue'.
APPEND ls_data TO lt_data.

" Transfer Posting
ls_data-movement_type = '311'.
ls_data-description    = 'Transfer Posting'.
APPEND ls_data TO lt_data.

" Return Delivery
ls_data-movement_type = '122'.
ls_data-description    = 'Return Delivery'.
APPEND ls_data TO lt_data.

1 ABAP Application F8
2 ABAP Application (Console) F9
3 ABAP Test Cockpit F2

```

the table

CLASS zcl_movtype_loader **DEFINITION**

```

PUBLIC
FINAL
CREATE PUBLIC.

```

```
PUBLIC SECTION.
```

```
INTERFACES if_oo_adt_classrun.
```

ENDCLASS!

CLASS zcl_movtype_loader **IMPLEMENTATION**.

```
METHOD if_oo_adt_classrun~main.
```

```
DATA: lt_data TYPE STANDARD TABLE OF zwh_movtype,
      ls_data TYPE zwh_movtype.
```

```
" Goods Receipt
ls_data-movement_type = '101'.
ls_data-description    = 'Goods Receipt'.
APPEND ls_data TO lt_data.
```

```
" Goods Issue
ls_data-movement_type = '261'.
ls_data-description    = 'Goods Issue'.
```

```

APPEND ls_data TO lt_data.

" Transfer Posting
ls_data-movement_type = '311'.
ls_data-description   = 'Transfer Posting'.
APPEND ls_data TO lt_data.

" Return Delivery
ls_data-movement_type = '122'.
ls_data-description   = 'Return Delivery'.
APPEND ls_data TO lt_data.

" Insert entries into the table
DELETE FROM zwh_movtype. " Optional: clean table before insert
INSERT zwh_movtype FROM TABLE @lt_data.

IF sy-subrc = 0.
  out->write( ' Records inserted into ZWH_MOVTYPE successfully.' ).
ELSE.
  out->write( ' Insert failed.' ).
ENDIF.

ENDMETHOD.

ENDCLASS.

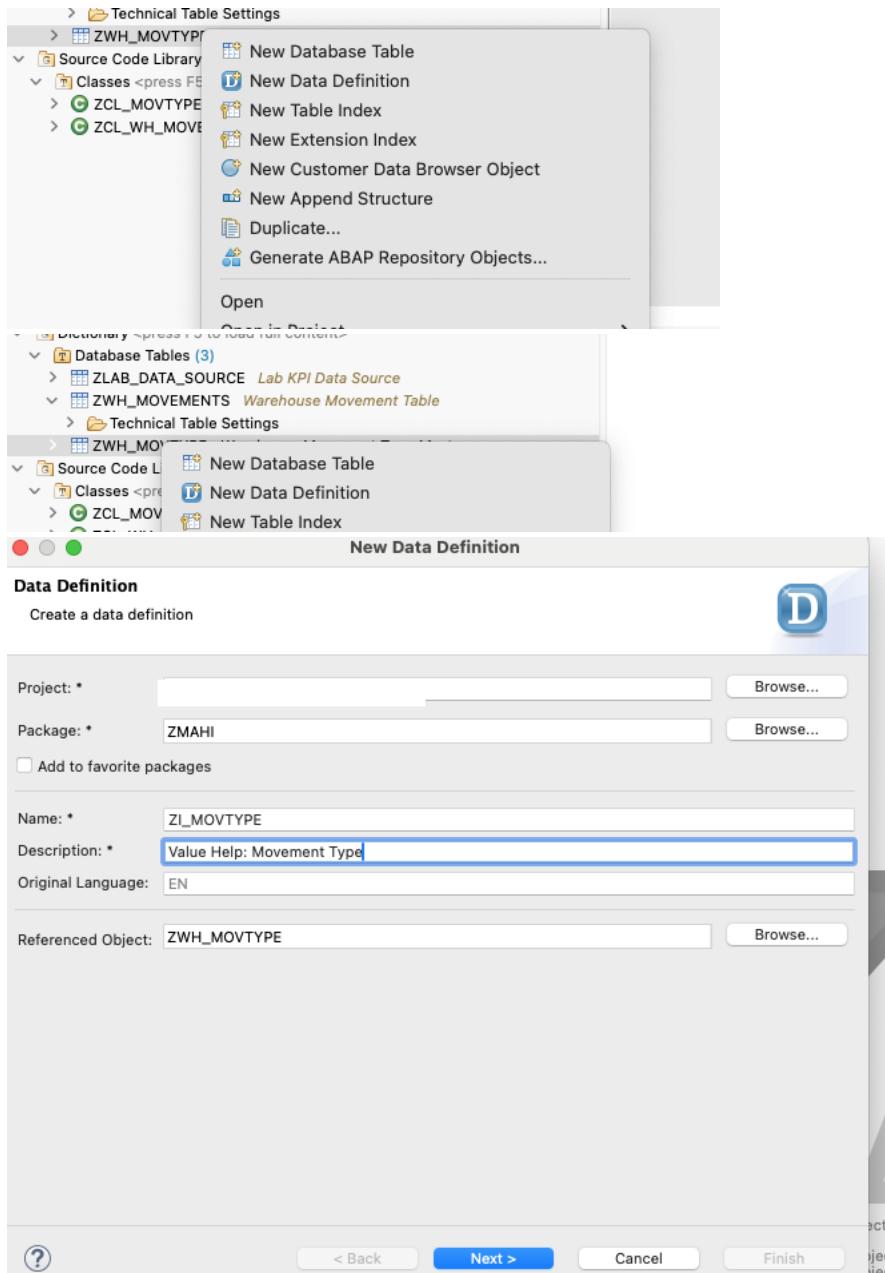
```

Records inserted into ZWH_MOVTYPE successfully.

MOVEMENT_TYPE	DESCRIPTION
101	Goods Receipt
261	Goods Issue
311	Transfer Posting
122	Return Delivery

 The row for MOVEMENT_TYPE 311 is highlighted."/>

Create CDS View for Text Table



The screenshot shows the SAP ABAP code editor with the title '[S4H] ZI_MOVTYPE X'. The code displayed is:

```
1 @EndUserText.label: 'Value Help: Movement Type'
2 @ObjectModel.resultSet.sizeCategory: #XS      // makes it a dropdown
3 define view entity ZI_MOVTYPE
4   as select from zwh_movtype
5 {
6   key movement_type,
7     description
8 }
9
```

```

@EndUserText.label: 'Value Help: Movement Type'
@ObjectModel.resultSet.sizeCategory: #XS // makes it a dropdown
define view entity ZI_MOVTYPE
  as select from zwh_movtype
{
  key movement_type,
  description
}

```

Execute CDS View



```

1@AccessControl.authorizationCheck: #NOT_REQUIRED
2 @EndUserText.label: 'Warehouse Movement Interface View'
3 define root view entity ZI_WH_MOVEMENTS
4   as select from zwh_movements
5 {
6   key movement_id,
7
8     material_id,
9
0     batch_number,
1
2     from_location,
3
4     to_location,
5
6     movement_type,
7
8     movement_date,
9
0   @Semantics.quantity.unitOfMeasure: 'uom'
1     quantity,
2
3     uom,
4
5     created_by
6 }
7

```

```

1@AccessControl.authorizationCheck: #NOT_REQUIRED
2 @EndUserText.label: 'Warehouse Movement Interface View'
3 define root view entity ZI_WH_MOVEMENTS
4   as select from zwh_movements
5 {
6   key movement_id,
7
8     material_id,
9
10    batch_number,
11
12    from_location,
13
14    to_location.
15  @Consumption.valueHelpDefinition: [{ 
16    entity: { name: 'ZI_MOVTTYPE', element: 'movement_type' }
17 }]
18   | movement_type,
19
20   movement_date,
21
22  @Semantics.quantity.unitOfMeasure: 'uom'
23    quantity,
24
25    uom,
26
27    created_by
28 }
29

30 @AccessControl.authorizationCheck: #NOT_REQUIRED
31 @EndUserText.label: 'Warehouse Movement Interface View'
32 define root view entity ZI_WH_MOVEMENTS
33   as select from zwh_movements
34   association [0..1] to ZI_MOVTTYPE as _MovementType on $projection.movement_type = _MovementType.movement_type
35 {
36   key movement_id,
37
38     material_id,
39
40     batch_number,
41
42     from_location,
43
44     to_location,
45     @Consumption.valueHelpDefinition: [{ 
46       entity: { name: 'ZI_MOVTTYPE', element: 'movement_type' }
47     }]
48     @ObjectModel.text.element: ['MovementTypeText']
49     movement_type,
50     _MovementType.description as MovementTypeText,
51     movement_date,
52
53     @Semantics.quantity.unitOfMeasure: 'uom'
54     quantity,
55
56     uom,
57
58     created_by
59 }
```

```
@UI.headerInfo: {
    typeName: 'Movement',
    typeNamePlural: 'Movements'
}
@Search.searchable: true
@ObjectModel.semanticKey: ['movement_id']
@EndUserText.label: 'Warehouse Movement Projection View'
@Metadata.allowExtensions: true
define root view entity ZC_WH_MOVEMENTS
    as projection on ZI_WH_MOVEMENTS
{
    key movement_id,
    @Search.defaultSearchElement: true
        material_id,
        batch_number,
        from_location,
        to_location,
        movement_type,
        MovementTypeText,|
        movement_date,
        quantity,
        uom,
        created_by
}

@Metadata.layer: #CORE
annotate view ZC_WH_MOVEMENTS with
{
    @UI.lineItem: [{ position: 10 }]
    @UI.selectionField: [{ position: 10 }]
    material_id;

    @UI.lineItem: [{ position: 20 }]
    batch_number;

    @UI.lineItem: [{ position: 30 }]
    @UI.selectionField: [{ position: 20 }]
    from_location;

    @UI.lineItem: [{ position: 40 }]
    @UI.selectionField: [{ position: 30 }]
    @UI.textArrangement: #TEXT_ONLY
    to_location;

    @UI.lineItem: [{ position: 50 }]
    @UI.selectionField: [{ position: 40 }]
    movement_type;

    @UI.lineItem: [{ position: 60 }]
    @UI.selectionField: [{ position: 50 }]
    movement_date;

    @UI.lineItem: [{ position: 70 }]
    quantity;

    @UI.lineItem: [{ position: 80 }]
    uom;

    @UI.lineItem: [{ position: 90 }]
    created_by;
}
```

```

1 @AccessControl.authorizationCheck: #NOT_REQUIRED
2 @EndUserText.label: 'Warehouse Movement Interface View'
3 define root view entity ZI_WH_MOVEMENTS
4   as select from zwh_movements
5   association [0..1] to ZI_MOVTTYPE as _MovementType on $projection.movement_type = _MovementType.movement_type
6
7 {
8   key movement_id,
9
0   material_id,
1
2   batch_number,
3
4   from_location,
5
6   to_location,
7   @Consumption.valueHelpDefinition: [{
8     entity: { name: 'ZI_MOVTTYPE', element: 'movement_type' }
9   }]
0   @ObjectModel.text.element: ['MovementTypeText']
1   movement_type,
2
3   _MovementType.description as MovementTypeText,
4   case movement_type
5     when '101' then 3 -- Goods Receipt → Green
6     when '261' then 1 -- Goods Issue → Red
7     when '311' then 2 -- Transfer Posting → Yellow
8     else 0
9   end
0           as movement_type_crit,
1   movement_date,
2
3   @Semantics.quantity.unitOfMeasure: 'uom'
4   quantity,
5
6   uom,
7
8   created_by
9

```

Analytics CDS View

```

@AbapCatalog.sqlViewName:'ZVWHMOVCHART'
@EndUserText.label: 'Warehouse Movement Analytics View'
@VDM.viewType: #BASIC
@AccessControl.authorizationCheck: #NOT_REQUIRED
define view ZI_WH_MOVEMENT_ANALYTICS
  as select from zwh_movements
{
  // Key field
  key movement_type,
  // Aggregated fields
  @DefaultAggregation: #SUM
  @Semantics.quantity.unitOfMeasure: 'uom'
  sum(quantity) as TotalQty,
  uom,
  material_id,
  movement_date,
  from_location,
  to_location

```

```
}
```

group by

```
movement_type,  
uom,  
material_id,  
movement_date,  
from_location,  
to_location
```

```
D [S4H] ZI_WH_MOVEMENT_ANALYTICS X
```

```
1 @AbapCatalog.sqlViewName: 'ZVWHMOVCHART'  
2 @EndUserText.label: 'Warehouse Movement Analytics View'  
3 @VDM.viewType: #BASIC  
4 @AccessControl.authorizationCheck: #NOT_REQUIRED  
5 define view ZI_WH_MOVEMENT_ANALYTICS  
6   as select from zwh_movements  
7 {  
8   // Key field  
9   key movement_type,  
10  
11  // Aggregated fields  
12  @DefaultAggregation: #SUM  
13  @Semantics.quantity.unitOfMeasure: 'uom'  
14  sum(quantity) as TotalQty,  
15  
16  uom,  
17  
18  material_id,  
19  movement_date,  
20  from_location,  
21  to_location  
22 }  
23 group by  
24   movement_type,  
25   uom,  
26   material_id,  
27   movement_date,  
28   from_location,  
29   to_location  
30
```

ZI_WH_MOVEMENT_ANALYTICS

60 rows retrieved - 106 ms

movement_type	TotalQty	uom	material_id	movement_date	from_location	to_location
122	9960.000	L	MAT1004	2025-05-02	PL4	PL1
122	8036.000	KG	MAT1004	2025-05-04	PL4	PL1
122	9296.000	EA	MAT1004	2025-05-06	PL4	PL1
122	7368.000	L	MAT1004	2025-05-08	PL4	PL1
122	8632.000	KG	MAT1004	2025-05-10	PL4	PL1
122	10000.000	EA	MAT1004	2025-05-12	PL4	PL1
122	7968.000	L	MAT1004	2025-05-14	PL4	PL1
122	9332.000	KG	MAT1004	2025-05-16	PL4	PL1
122	7304.000	EA	MAT1004	2025-05-18	PL4	PL1
122	8664.000	L	MAT1004	2025-05-20	PL4	PL1
122	10040.000	KG	MAT1004	2025-05-22	PL4	PL1
122	7996.000	EA	MAT1004	2025-05-24	PL4	PL1
122	9372.000	L	MAT1004	2025-05-26	PL4	PL1
122	7328.000	KG	MAT1004	2025-05-28	PL4	PL1
122	8704.000	EA	MAT1004	2025-05-30	PL4	PL1
101	8203.000	EA	MAT1001	2025-05-03	PL1	PL2
101	9462.000	L	MAT1001	2025-05-05	PL1	PL2
101	7535.000	KG	MAT1001	2025-05-07	PL1	PL2
101	8798.000	EA	MAT1001	2025-05-09	PL1	PL2
101	6867.000	L	MAT1001	2025-05-11	PL1	PL2
101	8134.000	KG	MAT1001	2025-05-13	PL1	PL2
101	9499.000	EA	MAT1001	2025-05-15	PL1	PL2
101	7470.000	L	MAT1001	2025-05-17	PL1	PL2
101	8831.000	KG	MAT1001	2025-05-19	PL1	PL2
101	6807.000	EA	MAT1001	2025-05-21	PL1	PL2
101	8163.000	L	MAT1001	2025-05-23	PL1	PL2
101	9539.000	KG	MAT1001	2025-05-25	PL1	PL2
101	7495.000	EA	MAT1001	2025-05-27	PL1	PL2
101	8871.000	L	MAT1001	2025-05-29	PL1	PL2
101	6826.000	KG	MAT1001	2025-05-31	PL1	PL2
261	8370.000	L	MAT1002	2025-05-02	PL2	PL3
261	9228.000	KG	MAT1002	2025-05-04	PL2	PL3

```

@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Warehouse Movement Interface View'
@OData.entitySet.name: 'WHMovements'
define root view entity ZI_WH_MOVEMENTS
  as select from zwh_movements

composition [0..*] of ZI_WH_MOV_ITEMS as _Items

association [0..1] to ZI_MOVTYPEn as _MovementType
  on $projection.movement_type = _MovementType.movement_type

association [0..*] to ZI_WH_MOVEMENT_ANALYTICS as _Analytics
  on $projection.movement_type = _Analytics.movement_type
{
  @EndUserText.label: 'Movement ID'
  key movement_id,

  @EndUserText.label: 'Material'
  material_id,

  @EndUserText.label: 'Batch'
  batch_number,

  @EndUserText.label: 'From Location'
  from_location,

  @EndUserText.label: 'To Location'
  to_location,
}
```

```
@EndUserText.label: 'Movement Type'
@Consumption.valueHelpDefinition: [{
    entity: { name: 'ZI_MOVTYPE', element: 'movement_type' }
}]
@ObjectModel.text.element: ['MovementTypeText']
movement_type,
_movementType.description as MovementTypeText,
case movement_type
when '101' then 3 // Goods Receipt → Green
when '261' then 1 // Goods Issue → Red
when '311' then 2 // Transfer Posting → Yellow
else 0           // Unknown → Grey
end as movement_type_crit,
@EndUserText.label: 'Movement Date'
movement_date,
@EndUserText.label: 'Quantity'
@Semantics.quantity.unitOfMeasure: 'uom'
quantity,
@EndUserText.label: 'UoM'
uom,
_items,
_analytics
}
```

```

1@UI.headerInfo: {
2    typeName: 'Movement',
3    typeNamePlural: 'Movements'
4 }
5 @Search.searchable: true
6 @ObjectModel.semanticKey: ['movement_id']
7 @EndUserText.label: 'Warehouse Movement Projection View'
8 @Metadata.allowExtensions: true
9 define root view entity ZC_WH_MOVEMENTS
10   as projection on ZI_WH_MOVEMENTS
11 {
12   key movement_id,
13   @Search.defaultSearchElement: true
14     material_id,
15     batch_number,
16     from_location,
17     to_location,
18     movement_type,
19     MovementTypeText,
20     movement_type_crit, | movement_type_crit, |
21     movement_date,
22     quantity,
23     uom,
24     created_by
25 }
26

1 @Metadata.layer: #CORE
2 annotate view ZC_WH_MOVEMENTS with
3 {
4@  @UI.lineItem: [{ position: 10 }]
5@  @UI.selectionField: [{ position: 10 }]
6  material_id;
7
8  @UI.lineItem: [{ position: 20 }]
9  batch_number;
L0
L1@  @UI.lineItem: [{ position: 30 }]
L2@  @UI.selectionField: [{ position: 20 }]
L3  from_location;
L4
L5@  @UI.lineItem: [{ position: 40 }]
L6@  @UI.selectionField: [{ position: 30 }]
L7@  @UI.textArrangement: #TEXT_ONLY
L8  to_location;
L9
?0@  @UI.lineItem: [{ position: 50,criticality: 'movement_type_crit' }]
?1@  @UI.selectionField: [{ position: 40 }]
?2  movement_type;
?3
?4@  @UI.lineItem: [{ position: 60 }]
?5@  @UI.selectionField: [{ position: 50 }]
?6  movement_date;
?7
?8  @UI.lineItem: [{ position: 70 }]
?9  quantity;
?0
?1@  @UI.lineItem: [{ position: 80 }]
?2  uom;
?3
?4@  @UI.lineItem: [{ position: 90 }]
?5  created_by;
?6 }

```

@Metadata.layer: #CORE

```

annotate view ZC_WH_MOVEMENTS with
{
```

```
@UI.lineItem: [{ position: 50, criticality: 'movement_type_crit' }]
@UI.selectionField: [{ position: 40 }]
@UI.textArrangement: #TEXT_ONLY
movement_type;

@UI.lineItem: [{ position: 10 }]
@UI.selectionField: [{ position: 10 }]
material_id;

@UI.lineItem: [{ position: 20 }]
batch_number;

@UI.lineItem: [{ position: 30 }]
@UI.selectionField: [{ position: 20 }]
from_location;

@UI.lineItem: [{ position: 40 }]
@UI.selectionField: [{ position: 30 }]
@UI.textArrangement: #TEXT_ONLY
to_location;

@UI.lineItem: [{ position: 60 }]
@UI.selectionField: [{ position: 50 }]
movement_date;

@UI.lineItem: [{ position: 70 }]
quantity;

@UI.lineItem: [{ position: 80 }]
uom;
}
```

ZMAHI <press F5 to load full content> Main Developments

Core Data Services <press F5 to load full content>

- Behavior Definitions <press F5 to load full content>
- Data Definitions <press F5 to load full content>
 - ZC_WH_MOVEMENTS *Warehouse Movement Definition View...*
 - ZI_MOVTTYPE *New Data Definition*
 - ZI_WH_MOV *New Access Control*
- Metadata Extension *New Metadata Extension*
- ZC_WH_MOV *New Service Definition*

Dictionary <press F5 to load full content>

Database Table

- ZLAB_DATA_ *New ABAP Test Class*
- ZWH_MOVER *New Customer Data Browser Object*
 - Technical
 - ZWH_MOVT *New Behavior Definition*
- ZWH_MOVT *Duplicate...*

Source Code Library

Classes <press F5 to load full content>

10
11
12
13
14
15
16
17
18
19
20
21
22
23

New Service Definition

Service Definition
Create a service definition

Project: *

Package: * ZMAHI

Add to favorite packages

Name: * ZGW_WH_MOVEMENTS_SRV

Description: * Service Definition for Warehouse Movement

Original Language: EN

Source Type: Definition

Referenced Object: ZC_WH_MOVEMENTS

?

< Back

Next >

Cancel

Finish

New Service Binding

Service Binding
Create a Service Binding.

Project: *

Package: * ZMAHI

Add to favorite packages

Name: * ZUI_C_WH_MOVEMENTS_BINDING

Description: * Service Binding- Warehouse Movements

Original Language: EN

Binding Type: * OData V4 - UI

Service Definition: * ZGW_WH_MOVEMENTS_SRV

General Information
This section describes general information about this service binding
Binding Type: OData V4 - UI

Services
Define services associated with the Service Binding
Local Service Endpoint: Unpublished

Service Name	Version	API State	Service Definition	Add Service...	Remove
ZGW_WH_MOVEMENTS_SRV	1.0.0	Not Released	ZGW_WH_MOVEMENTS_SRV	<input type="button" value="Publish"/>	<input type="button" value="Remove"/>
<small>Type filter text:</small>					
<small>General Information</small> This section describes general information about this service binding Binding Type: OData V4 - UI					
Service Version Details View information on selected service version Service Information Service URL: /sap/opu/odata4/sap/zui_c_wh_movements_binding/srv/sap/zgw_wh_movements_srv/0001/ <small>Type filter text:</small>					
Entity Set and Association ZC_WH_MOVEMENTS <input type="button" value="Preview..."/>					

Standard ▾

Material	Character Field with Length 10	Plant	Plant	3-Byte field	3-Byte field	Date	Quantity	Internal UoM	User Name
MAT1001	BATCH01	PLAN	PLAN	Goods Receipt (101)		May 31, 2025	10 500 KG	- KG	PRYVA
MAT1002	BATCH02	PLAN	PLAN	Goods Issue (261)		May 30, 2025	5 EA	- EA	PRYVA

```

@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Warehouse Movement Interface View'
define root view entity ZI_WH_MOVEMENTS
  as select from zwh_movements
  association [0..1] to ZI_MOVTTYPE as _MovementType on $projection

{
  @EndUserText.label: 'Movement ID'
  key movement_id,
  @EndUserText.label: 'Material'
  material_id,
  @EndUserText.label: 'Batch'
  batch_number,
  @EndUserText.label: 'From Location'
  from_location,
  @EndUserText.label: 'To Location'
  to_location,
  @EndUserText.label: 'Movement Type'
  @Consumption.valueHelpDefinition: [
    entity: { name: 'ZI_MOVTTYPE', element: 'movement_type' }
  ]
  @ObjectModel.text.element: ['MovementTypeText']
  movement_type,

  _MovementType.description as MovementTypeText,
  case movement_type
  when '101' then 3 -- Goods Receipt → Green
  when '261' then 1 -- Goods Issue → Red
  when '311' then 2 -- Transfer Posting → Yellow
  else 0           -- Unknown → Grey
  end               as movement_type_crit,
  @EndUserText.label: 'Movement Date'
  movement_date,
  @EndUserText.label: 'Quantity'
  @Semantics.quantity.unitOfMeasure: 'uom'
  quantity,
  @EndUserText.label: 'UoM'
  uom,
  @EndUserText.label: 'Movement Criticality'
  created_by
}

```

Create Class for Text Table

```
CLASS zcl_movtype_loader DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC.

  PUBLIC SECTION.
    INTERFACES if_oo_adt_classrun.

  ENDCCLASS.

CLASS zcl_movtype_loader IMPLEMENTATION.

  METHOD if_oo_adt_classrun~main.

    DATA: lt_data TYPE STANDARD TABLE OF zwh_movtype,
      ls_data TYPE zwh_movtype.

    " Goods Receipt
    ls_data-movement_type = '101'.
    ls_data-description   = 'Goods Receipt'.
    APPEND ls_data TO lt_data.

    " Goods Issue
    ls_data-movement_type = '261'.
    ls_data-description   = 'Goods Issue'.
    APPEND ls_data TO lt_data.

    " Transfer Posting
    ls_data-movement_type = '311'.
    ls_data-description   = 'Transfer Posting'.
    APPEND ls_data TO lt_data.

    " Return Delivery
    ls_data-movement_type = '122'.
    ls_data-description   = 'Return Delivery'.
    APPEND ls_data TO lt_data.

    " Insert entries into the table
    DELETE FROM zwh_movtype. " Optional: clean table before insert
    INSERT zwh_movtype FROM TABLE @lt_data.

    IF sy-subrc = 0.
      out->write( ' Records inserted into ZWH_MOVTYPE successfully.' ).
    ELSE.
      out->write( ' Insert failed.' ).
    ENDIF.

  ENDMETHOD.
```

ENDCLASS.

```
1 CLASS zcl_wh_movement_loader DEFINITION
2   PUBLIC
3   FINAL
4   CREATE PUBLIC.
5
6   PUBLIC SECTION.
7     INTERFACES if_oo_adt_classrun.
8 ENDCLASS.
9
0 CLASS zcl_wh_movement_loader IMPLEMENTATION.
1
2 METHOD if_oo_adt_classrun~main.
3
4   DATA: lt_movements TYPE STANDARD TABLE OF zwh_movements,
5         ls_movement  TYPE zwh_movements.
6   DATA: lt_materials TYPE STANDARD TABLE OF char10 WITH DEFAULT KEY,
7         lt_types      TYPE STANDARD TABLE OF char3  WITH DEFAULT KEY,
8         lt_uom        TYPE STANDARD TABLE OF char3  WITH DEFAULT KEY,
9         lt_plants     TYPE STANDARD TABLE OF char10 WITH DEFAULT KEY.
0
1
2   lt_materials = VALUE #( ( 'MAT1001' ) ( 'MAT1002' ) ( 'MAT1003' ) ( 'MAT1004' ) ).
3   lt_types      = VALUE #( ( '101' ) ( '261' ) ( '311' ) ( '122' ) ).  
4   lt_uom        = VALUE #( ( 'KG' ) ( 'EA' ) ( 'PC' ) ).  
5   lt_plants     = VALUE #( ( 'PLANT1' ) ( 'PLANT2' ) ( 'PLANT3' ) ( 'PLANT4' ) ).  
6   DATA: lv_material TYPE char10,  
7         lv_type      TYPE char3,  
8         lv_uom        TYPE char3,  
9         lv_from       TYPE char10,  
0         lv_to         TYPE char10,  
1         lv_date       TYPE sy-datum,  
2         lv_batch      TYPE char10.  
3
4
5   DELETE FROM zwh_movements.
6   out->write( 'Old records deleted.' && cl_abap_char_utilities=>newline ).  
7
8 DO 10000 TIMES.
9   lv_material = lt_materials[ sy-index MOD lines( lt_materials ) + 1 ].  
0   lv_type      = lt_types[ sy-index MOD lines( lt_types ) + 1 ].
```

Activate Class and execute the class

1 ZI_MOVTYPE (S4H_100_priya_en)
 2 ZWH_MOVTYPE (S4H_100_priya_en)
 3 ZC_WH_MOVEMENTS (S4H_100_priya_en)
 4 ZI_WH_MOVEMENTS (S4H_100_priya_en)
 5 ZI_OVP_DASHBOARD (S4H_100_priya_en)
 6 ACDOCA (S4H_100_priya_en)
 7 ZI_ACDOCA_DONUT1 (S4H_100_priya_en)
 8 ZGR_CDS_FINANCE1 (S4H_100_priya_en)
 9 ZC_LABOR_ANALYTIC_PROJ1 (S4H_100_priya_en)
 ZC_LABOR_ANALYTIC_PROJ (S4H_100_priya_en)

Run As >

Run Configurations...

Organize Favorites...

```

CLASS zcl_wh_movement_loader IMPLEMENTATION
  METHOD if_oo_adt_classrun~main.
    DATA: lt_movements TYPE STANDARD TABLE OF zwh_movements,
          ls_movement  TYPE zwh_movements.
    Sample warehouse movement records inserted successfully.

    ✓ Records inserted into ZWH_MOVTYPE successfully.

    Old records deleted.

    10,000 test records inserted into ZWH_MOVEMENTS.
  
```

1 ABAP Application F8
 2 ABAP Application (Console) F9
 3 ABAP Test Cockpit ⌘ F2
 4 ABAP Test Cockpit With... ⌘ F3
 5 ABAP Unit Test ⌘ F10
 6 ABAP Unit Test With... ⌘ F12
 7 Integration Forecast... ⌘ F4
 8 ABAP Package Check ⌘ F5

MOVEMENT_ID	MATERIAL_ID	BATCH_NUMBER	FROM_LOCATION	TO_LOCATION	MOVEMENT_TYPE	MOVEMENT_DATE	QUANTITY	UOM
D55B4144ABE1F	MAT1002	BATCH1	PLAN	PLAN	261	2025-05-30	2.000	EA
D55B4144ABE1F	MAT1003	BATCH2	PLAN	PLAN	311	2025-05-29	3.000	PC
D55B4144ABE1F	MAT1004	BATCH3	PLAN	PLAN	122	2025-05-28	4.000	KG
D55B4144ABE1F	MAT1001	BATCH4	PLAN	PLAN	101	2025-05-27	5.000	EA
D55B4144ABE1F	MAT1002	BATCH5	PLAN	PLAN	261	2025-05-26	6.000	PC
D55B4144ABE1F	MAT1003	BATCH6	PLAN	PLAN	311	2025-05-25	7.000	KG
D55B4144ABE1F	MAT1004	BATCH7	PLAN	PLAN	122	2025-05-24	8.000	EA
D55B4144ABE1F	MAT1001	BATCH8	PLAN	PLAN	101	2025-05-23	9.000	PC
D55B4144ABE1F	MAT1002	BATCH9	PLAN	PLAN	261	2025-05-22	10.000	KG
D55B4144ABE1F	MAT1003	BATCH10	PLAN	PLAN	311	2025-05-21	11.000	EA
D55B4144ABE1F	MAT1004	BATCH11	PLAN	PLAN	122	2025-05-20	12.000	PC
D55B4144ABE1F	MAT1001	BATCH12	PLAN	PLAN	101	2025-05-19	13.000	KG
D55B4144ABE1F	MAT1002	BATCH13	PLAN	PLAN	261	2025-05-18	14.000	EA
D55B4144ABE1F	MAT1003	BATCH14	PLAN	PLAN	311	2025-05-17	15.000	PC
D55B4144ABE1F	MAT1004	BATCH15	PLAN	PLAN	122	2025-05-16	16.000	KG
D55B4144ABE1F	MAT1001	BATCH16	PLAN	PLAN	101	2025-05-15	17.000	EA
D55B4144ABE1F	MAT1002	BATCH17	PLAN	PLAN	261	2025-05-14	18.000	PC
D55B4144ABE1F	MAT1003	BATCH18	PLAN	PLAN	311	2025-05-13	19.000	KG
D55B4144ABE1F	MAT1004	BATCH19	PLAN	PLAN	122	2025-05-12	20.000	EA
D55B4144ABE1F	MAT1001	BATCH20	PLAN	PLAN	101	2025-05-11	21.000	PC
D55B4144ABE1F	MAT1002	BATCH21	PLAN	PLAN	261	2025-05-10	22.000	KG
D55B4144ABE1F	MAT1003	BATCH22	PLAN	PLAN	311	2025-05-09	23.000	EA
D55B4144ABE1F	MAT1004	BATCH23	PLAN	PLAN	122	2025-05-08	24.000	PC
D55B4144ABE1F	MAT1001	BATCH24	PLAN	PLAN	101	2025-05-07	25.000	KG
D55B4144ABE1F	MAT1002	BATCH25	PLAN	PLAN	261	2025-05-06	26.000	EA
D55B4144ABE1F	MAT1003	BATCH26	PLAN	PLAN	311	2025-05-05	27.000	PC
D55B4144ABE1F	MAT1004	BATCH27	PLAN	PLAN	122	2025-05-04	28.000	KG
D55B4144ABE1F	MAT1001	BATCH28	PLAN	PLAN	101	2025-05-03	29.000	EA
D55B4144ABE1F	MAT1002	BATCH29	PLAN	PLAN	261	2025-05-02	30.000	PC
D55B4144ABE1F	MAT1003	BATCH30	PLAN	PLAN	311	2025-05-31	31.000	KG
D55B4144ABE1F	MAT1004	BATCH31	PLAN	PLAN	122	2025-05-30	32.000	EA
D55B4144ABE1F	MAT1001	BATCH32	PLAN	PLAN	101	2025-05-29	33.000	PC
D55B4144ABE1F	MAT1002	BATCH33	PLAN	PLAN	261	2025-05-28	34.000	KG
D55B4144ABE1F	MAT1003	BATCH34	PLAN	PLAN	311	2025-05-27	35.000	EA
D55B4144ABE1F	MAT1004	BATCH35	PLAN	PLAN	122	2025-05-26	36.000	PC
D55B4144ABE1F	MAT1001	BATCH36	PLAN	PLAN	101	2025-05-25	37.000	KG
D55B4144ABE1F	MAT1002	BATCH37	PLAN	PLAN	261	2025-05-24	38.000	EA
D55B4144ABE1F	MAT1003	BATCH38	PLAN	PLAN	311	2025-05-23	39.000	PC
D55B4144ABE1F	MAT1004	BATCH39	PLAN	PLAN	122	2025-05-22	40.000	KG
D55B4144ABE1F	MAT1001	BATCH40	PLAN	PLAN	101	2025-05-21	41.000	EA
D55B4144ABE1F	MAT1002	BATCH41	PLAN	PLAN	261	2025-05-20	42.000	PC
D55B4144ABE1F	MAT1003	BATCH42	PLAN	PLAN	311	2025-05-19	43.000	KG
D55B4144ABE1F	MAT1004	BATCH43	PLAN	PLAN	122	2025-05-18	44.000	EA
D55B4144ABE1F	MAT1001	BATCH44	PLAN	PLAN	101	2025-05-17	45.000	PC
D55B4144ABE1F	MAT1002	BATCH45	PLAN	PLAN	261	2025-05-16	46.000	KG
D55B4144ABE1F	MAT1003	BATCH46	PLAN	PLAN	311	2025-05-15	47.000	EA
D55B4144ABE1F	MAT1004	BATCH47	PLAN	PLAN	122	2025-05-14	48.000	PC
D55B4144ABE1F	MAT1001	BATCH48	PLAN	PLAN	101	2025-05-13	49.000	KG
D55B4144ABE1F	MAT1002	BATCH49	PLAN	PLAN	261	2025-05-12	50.000	EA
D55B4144ABE1F	MAT1003	BATCH50	PLAN	PLAN	311	2025-05-11	51.000	PC
D55B4144ABE1F	MAT1004	BATCH51	PLAN	PLAN	122	2025-05-10	52.000	KG
D55B4144ABE1F	MAT1001	BATCH52	PLAN	PLAN	101	2025-05-09	53.000	EA
D55B4144ABE1F	MAT1002	BATCH53	PLAN	PLAN	261	2025-05-08	54.000	PC
D55B4144ABE1F	MAT1003	BATCH54	PLAN	PLAN	311	2025-05-07	55.000	KG
D55B4144ABE1F	MAT1004	BATCH55	PLAN	PLAN	122	2025-05-06	56.000	EA
D55B4144ABE1F	MAT1001	BATCH56	PLAN	PLAN	101	2025-05-05	57.000	PC
D55B4144ABE1F	MAT1002	BATCH57	PLAN	PLAN	261	2025-05-04	58.000	KG
D55B4144ABE1F	MAT1003	BATCH58	PLAN	PLAN	311	2025-05-03	59.000	EA
D55B4144ABE1F	MAT1004	BATCH59	PLAN	PLAN	122	2025-05-02	60.000	PC
D55B4144ABE1F	MAT1001	BATCH60	PLAN	PLAN	101	2025-05-31	61.000	KG
D55B4144ABE1F	MAT1002	BATCH61	PLAN	PLAN	261	2025-05-30	62.000	EA
D55B4144ABE1F	MAT1003	BATCH62	PLAN	PLAN	311	2025-05-29	63.000	PC
D55B4144ABE1F	MAT1004	BATCH63	PLAN	PLAN	122	2025-05-28	64.000	KG
D55B4144ABE1F	MAT1001	BATCH64	PLAN	PLAN	101	2025-05-27	65.000	EA
D55B4144ABE1F	MAT1002	BATCH65	PLAN	PLAN	261	2025-05-26	66.000	PC
D55B4144ABE1F	MAT1003	BATCH66	PLAN	PLAN	311	2025-05-25	67.000	KG
D55B4144ABE1F	MAT1004	BATCH67	PLAN	PLAN	122	2025-05-24	68.000	EA
D55B4144ABE1F	MAT1001	BATCH68	PLAN	PLAN	101	2025-05-23	69.000	PC
D55B4144ABE1F	MAT1002	BATCH69	PLAN	PLAN	261	2025-05-22	70.000	KG

Create Item Table

```
@EndUserText.label : 'Warehouse Movement Item Table'  
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE  
@AbapCatalog.tableCategory : #TRANSPARENT  
@AbapCatalog.deliveryClass : #A  
@AbapCatalog.dataMaintenance : #RESTRICTED  
define table zwh_mov_items {  
  
    key movement_id : sysuuid_x not null;  
    key item_id : int4 not null;  
    storage_bin : lgpla;  
    special_stock : char1;  
    handling_unit : char10;  
    @Semantics.quantity.unitOfMeasure : 'zwh_mov_items.item_uom'  
    item_quantity : menge_d;  
    item_uom : meins;  
  
}
```

```
@EndUserText.label : 'Warehouse Movement Item Table'  
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE  
@AbapCatalog.tableCategory : #TRANSPARENT  
@AbapCatalog.deliveryClass : #A  
@AbapCatalog.dataMaintenance : #RESTRICTED  
define table zwh_mov_items {  
  
    key movement_id : sysuuid_x not null;  
    key item_id : int4 not null;  
    storage_bin : lgpla;  
    special_stock : char1;  
    handling_unit : char10;  
    @Semantics.quantity.unitOfMeasure : 'zwh_mov_items.item_uom'  
    item_quantity : menge_d;  
    item_uom : meins;  
  
}
```

Create Class for Item Table to insert records

```
401 ZCL_WH_MOV_ITEM... 402 ZCL_WH_MOV_ITEM... 403 ZCL_WH_MOV_ITEM... 404 ZCL_WH_MOV_ITEM... 405 ZCL_WH_MOV_ITEM... ▾ /  
ZCL_WH_MOV_ITEM_LOADER ▶  
⊕ CLASS zcl_wh_mov_item_loader DEFINITION  
  PUBLIC  
  FINAL  
  CREATE PUBLIC.  
  
  PUBLIC SECTION.  
    INTERFACES if_oo_adt_classrun.  
  
ENDCLASS.  
  
⊕ CLASS zcl_wh_mov_item_loader IMPLEMENTATION.  
  
⊕ METHOD if_oo_adt_classrun~main.  
  
  DATA: lt_items      TYPE STANDARD TABLE OF zwh_mov_items,  
        ls_item       TYPE zwh_mov_items,  
        lv_item_id    TYPE int4.  
  
  DATA: lt_bins       TYPE STANDARD TABLE OF lgpla WITH DEFAULT KEY,  
        lt_stocks     TYPE STANDARD TABLE OF char1 WITH DEFAULT KEY,  
        lt_units      TYPE STANDARD TABLE OF char10 WITH DEFAULT KEY
```

```
CLASS zcl_wh_mov_item_loader DEFINITION
```

```
  PUBLIC  
  FINAL  
  CREATE PUBLIC.
```

```
  PUBLIC SECTION.  
    INTERFACES if_oo_adt_classrun.
```

```
ENDCLASS;
```

```
CLASS zcl_wh_mov_item_loader IMPLEMENTATION.
```

```
  METHOD if_oo_adt_classrun~main.
```

```
    DATA: lt_items      TYPE STANDARD TABLE OF zwh_mov_items,  
          ls_item       TYPE zwh_mov_items,  
          lv_item_id    TYPE int4.
```

```
    DATA: lt_bins       TYPE STANDARD TABLE OF lgpla WITH DEFAULT KEY,  
          lt_stocks     TYPE STANDARD TABLE OF char1 WITH DEFAULT KEY,  
          lt_units      TYPE STANDARD TABLE OF char10 WITH DEFAULT KEY.
```

```
    DATA: lv_bin        TYPE lgpla,
```

```

lv_stock  TYPE char1,
lv_unit   TYPE char10.

" Sample config values
lt_bins  = VALUE #(( 'BIN001' ) ( 'BIN002' ) ( 'BIN003' ) ( 'BIN004' )).
lt_stocks = VALUE #(( 'Q' ) ( 'S' ) ( 'E' )).
lt_units  = VALUE #(( 'HU1001' ) ( 'HU1002' ) ( 'HU1003' ) ( 'HU1004' )).

" Cleanup old items
DELETE FROM zwh_mov_items.
out->write( 'Old item records deleted.' && cl_abap_char_utilities=>newline ).

" Read all header movement_ids
SELECT movement_id FROM zwh_movements INTO TABLE @DATA(lt_movements).

LOOP AT lt_movements INTO DATA(ls_movement).

" Generate 1 to 3 item rows per movement
DATA(lv_count) = 1 + sy-tabix MOD 3.

DO lv_count TIMES.

  lv_item_id = sy-index.

  lv_bin  = lt_bins[ sy-index MOD lines( lt_bins ) + 1 ].
  lv_stock = lt_stocks[ sy-index MOD lines( lt_stocks ) + 1 ].
  lv_unit  = lt_units[ sy-index MOD lines( lt_units ) + 1 ].

  CLEAR ls_item.
  ls_item-movement_id  = ls_movement-movement_id.
  ls_item-item_id       = lv_item_id.
  ls_item-storage_bin   = lv_bin.
  ls_item-special_stock = lv_stock.
  ls_item-handling_unit = lv_unit.
  ls_item-item_quantity = 5 + ( sy-tabix MOD 20 ).
  ls_item-item_uom      = 'EA'.

  APPEND ls_item TO lt_items.

  IF lines( lt_items ) >= 1000.
    INSERT zwh_mov_items FROM TABLE @lt_items.
    CLEAR lt_items.
  ENDIF.

ENDDO.

ENDLOOP.

IF lt_items IS NOT INITIAL.
  INSERT zwh_mov_items FROM TABLE @lt_items.
ENDIF.

```

```
out->write( 'Warehouse movement item records inserted successfully.' ).
```

```
ENDMETHOD.
```

```
ENDCLASS.
```

Create CDS View for Item Table

```
@AbapCatalog.sqlViewName: 'ZVWHMOVITEMS'  
@AbapCatalog.compiler.compareFilter: true  
@AccessControl.authorizationCheck: #NOT_REQUIRED  
@EndUserText.label: 'Warehouse Movement Item View'  
@VDM.viewType: #BASIC  
@Metadata.allowExtensions: true  
define view ZI_WH_MOV_ITEMS  
  as select from zwh_mov_items  
  association to parent ZI_WH_MOVEMENTS as _Parent  
    on $projection.movement_id = _Parent.movement_id  
{  
  @EndUserText.label: 'Movement ID'  
  key movement_id,  
  
  @EndUserText.label: 'Item ID'  
  key item_id,  
  
  @EndUserText.label: 'Storage Bin'  
  storage_bin,  
  
  @EndUserText.label: 'Special Stock Indicator'  
  special_stock,  
  
  @EndUserText.label: 'Handling Unit'  
  handling_unit,  
  
  @EndUserText.label: 'Quantity'  
  item_quantity,  
  
  @EndUserText.label: 'Unit of Measure'  
  item_uom,  
  _Parent  
}
```

Metadata Extension for Item table

```
@Metadata.layer: #CORE  
annotate entity ZI_WH_MOV_ITEMS  
  with  
{  
  @UI.linelitem: [{ position: 1 }]
```

```

item_id;

@UI.lineItem: [{ position: 2 }]
movement_id;
@UI.lineItem: [{ position: 10 }]
storage_bin;

@UI.lineItem: [{ position: 20 }]
special_stock;

@UI.lineItem: [{ position: 30 }]
handling_unit;

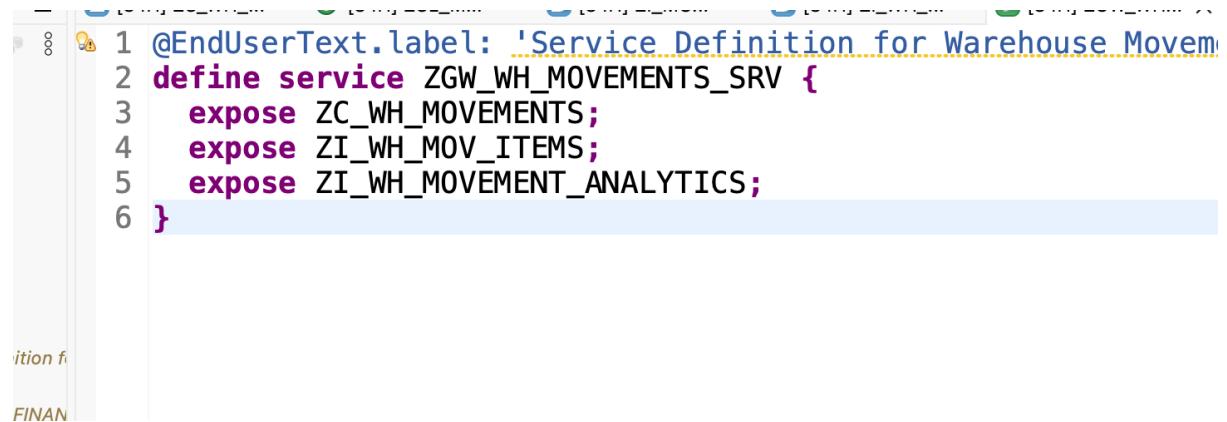
@UI.lineItem: [{ position: 40 }]
item_quantity;

@UI.lineItem: [{ position: 50 }]
item_uom;

}

```

Create Service Definition and Service Binding



```

1 @EndUserText.label: 'Service Definition for Warehouse Movements'
2 define service ZGW_WH_MOVEMENTS_SRV {
3   expose ZC_WH_MOVEMENTS;
4   expose ZI_WH_MOV_ITEMS;
5   expose ZI_WH_MOVEMENT_ANALYTICS;
6 }

```

The screenshot shows a code editor window with the following content:

```

1 @EndUserText.label: 'Service Definition for Warehouse Movements'
2 define service ZGW_WH_MOVEMENTS_SRV {
3   expose ZC_WH_MOVEMENTS;
4   expose ZI_WH_MOV_ITEMS;
5   expose ZI_WH_MOVEMENT_ANALYTICS;
6 }

```

The code is written in OpenEdge Language. It starts with an annotation `@EndUserText` followed by a label. Then it defines a service named `ZGW_WH_MOVEMENTS_SRV` using the `define service` keyword. Inside the service block, there are three `expose` statements, each followed by an interface name: `ZC_WH_MOVEMENTS`, `ZI_WH_MOV_ITEMS`, and `ZI_WH_MOVEMENT_ANALYTICS`. The code editor has syntax highlighting and a code completion feature.

Note: There are additional views I have exposed. In a later post, we will explore how to use them for **Analytics** purposes.

Service Binding: ZSB_WH_MOVEMENTS_V2

General Information
This section describes general information about this service binding
Binding Type: OData V2 - UI

Service Versions
Define service versions associated with the service binding
type filter text

Version	API & Service Definition	Add...
1.0.0	Not F ZGW_WH_MOVEMENTS_SRV	Remove

Service Version Details
View information on selected service version
Local Service Endpoint: Published Unpublish

Service Information
Service URL: /sap/opu/odata/sap/ZSB_WH_MOVEMENTS_V2

Entity Set and Association
ZC_WH_MOVEMENTS
 -> to_Analytics
 -> to_Items
 -> ZI_MOVTYP
 -> ZI_WH_MOVEMENT_ANALYTICS
 -> ZI_WH_MOV_ITEMS

Preview and Final Output for List View

Standard ▾

Material: From Location: To Location: Movement Type: Movement Date:

Search Adapt Filters

Movements (10,000)

Material	Batch	From Location	To Location	Movement Type	Movement Date	Quantity	UoM	Movement Criticality
MAT1002	BATCH1	PL2	PL3	✖ Goods Issue	May 30, 2025	2	EA	>
MAT1003	BATCH2	PL3	PL4	⚠ Transfer Posting	May 29, 2025	3.000	L	>
MAT1004	BATCH3	PL4	PL1	Return Delivery	May 28, 2025	4.000	KG	>
MAT1001	BATCH4	PL1	PL2	✓ Goods Receipt	May 27, 2025	5	EA	>
MAT1002	BATCH5	PL2	PL3	✖ Goods Issue	May 26, 2025	6.000	L	>
MAT1003	BATCH6	PL3	PL4	⚠ Transfer Posting	May 25, 2025	7.000	KG	>
MAT1004	BATCH7	PL4	PL1	Return Delivery	May 24, 2025	8	EA	>
MAT1001	BATCH8	PL1	PL2	✓ Goods Receipt	May 23, 2025	9.000	L	>

Final Output for Object Page

MAT1003

2d55b414-4abe-1fe0-8fc4-0780617233e8

Movement Items

Standard ▾

Item ID	Movement ID	Storage Bin	Special Stoc...	Handling Unit	Quantity	Unit of Mea...	
1	2d55b414-4abe-1fe0-8fc4-0780617233e8	BIN002	S	HU1002	7.000	EA	>
2	2d55b414-4abe-1fe0-8fc4-0780617233e8	BIN003	E	HU1003	8.000	EA	>
3	2d55b414-4abe-1fe0-8fc4-0780617233e8	BIN004	Q	HU1004	9.000	EA	>

Final Output for Value Help

Movement Type: Movement Date:

101 Goods Receipt 122 Return Delivery 261 Goods Issue 311 Transfer Posting

[Adapt Filters](#)

ate
025 2 EA EA
125 3.000 I