Rap app

13 January 2026 15:35

```
@EndUserText.label : 'Purchase Requisition Header'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #ALLOWED
define table ymm_pr_hdr {

key client          : abap.clnt not null;
key sid             : sysuuid_x16 not null;
key pr_id           : banfn not null;
pr_type             : bsart;
requester           : ernam;
plant               : werks_d;
status              : ymm_pr_status_de;
@Semantics.amount.currencyCode : 'ymm_pr_hdr.currency'
total_value         : wrbtr;
currency            : waers
with value help zcuky_640
where waers = ymm_pr_hdr.currency;
sendmail            : abap.char(3);
location            : stras;
created_by          : ernam;
created_on          : erdat;
locallastchangd_by  : abp_locinst_lastchange_user;
locallastchangd_at  : abp_locinst_lastchange_tstmpl;

}
```

```
@EndUserText.label : 'Purchase Requisition Item'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #ALLOWED
define table ymm_pr_itm {

key client    : abap.clnt not null;
key sid       : sysuuid_x16 not null;
key pr_id     : banfn not null;
key item_no   : bnfpo not null;
material      : matnr;
@Semantics.quantity.unitOfMeasure : 'ymm_pr_itm.uom'
qty           : menge_d;
uom           : meins;
@Semantics.amount.currencyCode : 'ymm_pr_hdr.currency'
price         : netpr;
@Semantics.amount.currencyCode : 'ymm_pr_hdr.currency'
item_value    : wrbtr;
vendor        : lifnr;
delivery_date : eindt;

}
```

```
@EndUserText.label : 'Purchase Order Header'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table ymm_po_hdr {

key client        : abap.clnt not null;
key sid           : sysuuid_x16 not null;
key poid          : ebeln not null;
prid              : banfn;
vendor            : lifnr;
docdate           : bedat;
status            : ymm_po_status_de;
@Semantics.amount.currencyCode : 'ymm_po_hdr.currency'
totalvalue        : wrbtr;
currency          : waers;
porg              : ekorg;
pgroup            : bkgrp;
ccode             : bukrs;
prid2             : banfn;
createdby         : ernam;
createdon         : erdat;
locallastchangdby : abp_locinst_lastchange_user;
locallastchangdat : abp_locinst_lastchange_tstmpl;

}
```

```
@EndUserText.label : 'Purchase Order Item'
```

```
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table ymm_po_itm {

key client   : abap.clnt not null;
key sid      : sysuuid_x16 not null;
key prid     : banfn not null;
key poid     : ebeln not null;
key itemno   : ebelp not null;
material     : matnr;
mtext        : txz01;
prrid        : banfn;
pritemno     : bnfpo;
@Semantics.quantity.unitOfMeasure : 'ymm_po_itm.uom'
qty          : menge_d;
uom          : meins;
@Semantics.amount.currencyCode : 'ymm_po_hdr.currency'
price        : netpr;
@Semantics.amount.currencyCode : 'ymm_po_hdr.currency'
itemvalue    : wrbtr;
deliverydate : eindt;

}
```

```
@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'I view For PR Header'
@Metadata.ignorePropagatedAnnotations: true
define root view entity YI_MM_PR_HDR
as select from ymm_pr_hdr
composition [0..*] of YI_MM_PR_ITM   as _PR_Item
composition [1..*] of yi_po_hdr      as _PO_Header
association [0..*] to ZPR_status_VH  as _status     on
$projection.Status = _status.DomvalueL
association [1]    to Zplant1_vh        as _Plant   on
$projection.Plant = _Plant.werks
association [0..*] to ZOrder_type_VH as _Order_type on
$projection.PrType = _Order_type.bsart
{
key sid                 as Sid,
key pr_id               as PrId,
pr_type            as PrType,
requester          as Requester,
plant              as Plant,
status             as Status,
@Semantics.amount.currencyCode: 'Currency'
total_value        as TotalValue,
currency           as Currency,
@Semantics.user.createdBy: true
created_by         as CreatedBy,
@Semantics.systemDate.createdAt: true
created_on         as CreatedOn,
@Semantics.user.localInstanceLastChangedBy: true
locallastchangd_by as LocallastchangdBy,
@Semantics.systemDateTime.localInstanceLastChangedAt: true
locallastchangd_at as LocallastchangdAt,
case status
when 'A' then 3
when 'R' then 1
when 'S' then 2
else 1
end                as Status_color,
sendmail           as sendmail,
case sendmail
when 'NO' then 2
when 'YES' then 3
else 1 end         as status_mail_color,
location           as Location,
_PR_Item,
_status,
_Order_type,
_Plant,
_PO_Header
//      @ObjectModel.virtualElementCalculatedBy: 'ABAP:ZCL_CUSTOM_VH'
//      @Semantics.amount.currencyCode: 'Currency'
//      cast( '0' as wrbtr ) as Remain_amt

}
```

```
@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'I view For PR Item'
@Metadata.ignorePropagatedAnnotations: true
define view entity YI_MM_PR_ITM
as select from ymm_pr_itm
association     to parent YI_MM_PR_HDR as _PR_Header on
$projection.Sid  = _PR_Header.Sid
```

```
and $projection.PrId = _PR_Header.PrId
association [1] to Zmaterial_VH        as _materil   on
$projection.Material = _materil.Matnr
association [1] to ZSupplier_VH        as _Supplier  on
$projection.Vendor = _Supplier.lifnr
{
key sid                  as Sid,
key pr_id                as PrId,
key item_no              as ItemNo,
material             as Material,
@Semantics.quantity.unitOfMeasure: 'Uom'
qty                  as Qty,
uom                  as Uom,
@Semantics.amount.currencyCode: 'Currency'
price                as Price,
@Semantics.amount.currencyCode: 'Currency'
item_value           as ItemValue,
vendor               as Vendor,
delivery_date        as DeliveryDate,
_PR_Header.Currency as Currency,
//       _PR_Header.Requester as requestor,
_PR_Header,
_materil,
_Supplier
}
```

```
@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'I View For PO Header'
@Metadata.ignorePropagatedAnnotations: true
define view entity yi_po_hdr
as select from ymm_po_hdr
association        to parent YI_MM_PR_HDR as _PR_header    on
$projection.Prid = _PR_header.PrId
and $projection.Sid  = _PR_header.Sid
association [1..1] to Zpgroup_vh          as _PurchseGroup on
$projection.Pgroup = _PurchseGroup.Ekgrp
association [1..1] to Zporg_VH            as _PurchaseOrg  on
$projection.Porg = _PurchaseOrg.Ekorg
//  association [1..*] to yi_po_itm           as _PO_Item      on
$projection.Poid = _PO_Item.Poid
composition [1..*] of yi_po_itm           as _PO_Item

{
key   sid               as Sid,
key   poid              as Poid,
key   prid              as Prid,
vendor            as Vendor,
docdate           as Docdate,
porg              as Porg,
pgroup            as Pgroup,
prid2             as Prid2,
ccode             as Ccode,
status            as Status,
@Semantics.amount.currencyCode: 'Currency'
totalvalue        as Totalvalue,
currency          as Currency,
@Semantics.user.createdBy: true
createdby         as Createdby,
@Semantics.systemDate.createdAt: true
createdon         as Createdon,
@Semantics.user.localInstanceLastChangedBy: true
locallastchangdby as Locallastchangdby,
@Semantics.systemDate.localInstanceLastChangedAt: true
locallastchangdat as Locallastchangdat,
_PR_header,
_PurchseGroup,
_PurchaseOrg,
_PO_Item
}
```

```
@AbapCatalog.viewEnhancementCategory:
[#NONE]
@AccessControl.authorizationCheck:
#NOT_REQUIRED
@EndUserText.label: 'I View For PO Item'
@Metadata.ignorePropagatedAnnotations: true
define view entity yi_po_itm
as select from ymm_po_itm
association        to parent yi_po_hdr as
_PO_Header on  $projection.Poid =
_PO_Header.Poid
and $projection.Sid  = _PO_Header.Sid
and $projection.Prid = _PO_Header.Prid
association [1..1] to Zmaterial_VH     as
_Material  on  $projection.Material =
_Material.Matnr
association        to YI_MM_PR_HDR     as
_PR_Header on  $projection.Sid  =
_PR_Header.Sid
```

```
and $projection.Prid = _PR_Header.PrId
{
key sid                 as Sid,
key prid                as Prid,
key poid                as Poid,
key itemno              as Itemno,

material            as Material,
mtext               as Mtext,
prrid               as Prrid,
pritemno            as Pritemno,
@Semantics.quantity.unitOfMeasure: 'Uom'
qty                 as Qty,
uom                 as Uom,
@Semantics.amount.currencyCode: 'Currency'
price               as Price,
@Semantics.amount.currencyCode: 'Currency'
itemvalue           as Itemvalue,
deliverydate        as Deliverydate,
_PO_Header.Currency as Currency,
_PO_Header,
_Material,
_PR_Header
}
```

```
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'P view For PR Header'
@Metadata.ignorePropagatedAnnotations: true
@Metadata.allowExtensions: true
@ObjectModel.resultSet.sizeCategory: #XS
define root view entity YP_mm_pr_hdr
provider contract transactional_query
as projection on YI_MM_PR_HDR
{
key     Sid,
key     PrId,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZOrder_type_VH',
element: 'bsart'
} }]
PrType,
Requester,
@ObjectModel.text.element: [ 'Plandesc' ]
@Consumption.valueHelpDefinition: [{ entity: {
name: 'Zplant1_vh',
element: 'werks'
} }]
Plant,
_Plant.name1        as Plandesc,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZPR_STATUS_VH',
element: 'DomvalueL'
} }]
@ObjectModel.text.element: [ 'descr' ]
Status,
@Semantics.amount.currencyCode: 'Currency'
TotalValue,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZCurr_cuky_VH',
element: 'Waers'
} }]
Currency,
CreatedBy,
CreatedOn,
sendmail,
status_mail_color,
LocallastchangdBy,
LocallastchangdAt,
Status_color,
Location,
/* Associations */
//      _PO_Header,
_Order_type,
_Plant,
_PO_Header : redirected to composition child yP_po_hdr,
_status.description as descr : localized,
_PR_Item   : redirected to composition child YP_MM_PR_ITM,
@ObjectModel.virtualElementCalculatedBy: 'ABAP:ZCL_CUSTOM_VH'
@Semantics.amount.currencyCode: 'Currency'
virtual Remain_amt     : abap.curr( 15, 2 ),
@ObjectModel.virtualElementCalculatedBy: 'ABAP:ZCL_CUSTOM_VH'
@Semantics.amount.currencyCode: 'Currency'
virtual total_Item_sum : abap.curr( 15, 2 )


}
```

```
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'P view For PR Item'
@Metadata.ignorePropagatedAnnotations: true
@Metadata.allowExtensions: true
define view entity YP_MM_PR_ITM
as projection on YI_MM_PR_ITM
{
key     Sid,
key     PrId,
key     ItemNo,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'Zmaterial_VH',
element: 'Matnr'
} }]
//          @ObjectModel.text.element: [ 'description' ]
Material,
_materil.Maktx       as description,
@Semantics.quantity.unitOfMeasure: 'Uom'
Qty,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZI_VH_UnitOfMeasure',
element: 'UnitOfMeasure'
} }]
Uom,
@Semantics.amount.currencyCode: 'Currency'
Price,
@Semantics.amount.currencyCode: 'Currency'
ItemValue,
@ObjectModel.text.element: [ 'Suppliername' ]
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZSupplier_VH',
element: 'lifnr'
} }]
Vendor,
_Supplier.name1      as Suppliername,
DeliveryDate,
Currency,
_PR_Header.Requester as requestor,
/* Associations */
@ObjectModel.virtualElementCalculatedBy: 'ABAP:YCL_HELPER_PR_PO'
@Semantics.amount.currencyCode: 'Currency'
virtual total  : abap.curr( 15,2 ),
_PR_Header : redirected to parent YP_mm_pr_hdr,
_Supplier,
@ObjectModel.virtualElementCalculatedBy: 'ABAP:ZCL_CUSTOM_MAIL'
virtual isHide : abap_boolean
}



@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'P View For Po Header'
@Metadata.ignorePropagatedAnnotations: true
@Metadata.allowExtensions: true
define view entity yP_po_hdr
as projection on yi_po_hdr
{
key   Sid,
key   Poid,
key   Prid,
//        @ObjectModel.text.element: [ 'Suppliername' ]
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZSupplier_VH',
element: 'lifnr'
} }]
Vendor,
Docdate,
@ObjectModel.text.element: [ 'PorgName' ]
@Consumption.valueHelpDefinition:
[{ entity: {
name: 'Zporg_VH',
element: 'Ekorg'
},
additionalBinding: [ { localElement: 'Ccode' ,element: 'Bukrs' }] }]
Porg,
@ObjectModel.text.element: [ 'Pgname' ]
@Consumption.valueHelpDefinition: [{ entity: {
name: 'Zpgroup_vh',
element: 'Ekgrp'
} }]
Pgroup,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZCE_CCcode_VH',
element: 'bukrs'
} }]
Ccode,
Prid2,
Status,
@Semantics.amount.currencyCode: 'Currency'
Totalvalue,
Currency,
Createdby,
Createdon,
Locallastchangdby,
Locallastchangdat,
/* Associations */
_PR_header : redirected to parent YP_mm_pr_hdr,
_PurchseGroup.Eknam as Pgname,
_PurchaseOrg.Ekotx  as PorgName,
_PO_Item   : redirected to composition child YP_PO_ITM

}
```

```
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'P View For PO Item'
@Metadata.ignorePropagatedAnnotations: true
@Metadata.allowExtensions: true
define view entity YP_PO_ITM
//  provider contract transactional_query
as projection on yi_po_itm
{
key Sid,
key Prid,
key Poid,
key Itemno,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'Zmaterial_VH',
element:   'Matnr'
},
additionalBinding: [{ localElement: 'Mtext'
,element: 'Maktx'}] }]
Material,
Mtext,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZPR_VH',
element: 'PrId'
} }]
Prrid,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZPR_Itm_VH',
element: 'ItemNo'
},additionalBinding: [{ localElement: 'Prrid',element: 'PrId' }] }]
Pritemno,
@Semantics.quantity.unitOfMeasure: 'Uom'
Qty,
Uom,
@Semantics.amount.currencyCode: 'Currency'
Price,
@Semantics.amount.currencyCode: 'Currency'
Itemvalue,
Deliverydate,
Currency,
/* Associations */
_Material,
_PO_Header : redirected to parent yP_po_hdr,
_PR_Header : redirected to YP_mm_pr_hdr
}
```

```
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Approver Projection View'
@Metadata.ignorePropagatedAnnotations: true
@Metadata.allowExtensions: true

define root view entity YP_PR_Approver
provider contract transactional_query
as projection on YI_MM_PR_HDR
{
key Sid,
@Consumption.valueHelpDefinition: [{ entity: {
name: 'Zcustom_entity_VH',
element: 'pr_id'
} }]
key PrId,
PrType,
Requester,
Plant,
@ObjectModel.text.element: [ 'descr' ]
Status,
@Semantics.amount.currencyCode: 'Currency'
TotalValue,
Currency,
CreatedBy,
CreatedOn,
LocallastchangdBy,
LocallastchangdAt,
Status_color,
sendmail,
status_mail_color,
Location,
/* Associations */
_Order_type,
_Plant,
_PO_Header,
_PR_Item,
_status,
_status.description as descr : localized
}
```

```
@EndUserText.label: 'Abstarct Entity'
define abstract entity Zabs_entity
//  with parameters parameter_name : parameter_type
{
@EndUserText.label: 'Enter Mail'
email     : abap.char( 20 );
@EndUserText.label: 'Enter Name'
Name      : abap.char(10);
//  Pr_Id      : banfn;
//    prtype     : bsart;
@EndUserText.label: 'Requested    By'
requester : ernam;
//  @ObjectModel.readOnly
//  plant      : werks_d;
//  @Consumption.valueHelpDefinition: [{ entity: {
//    name    : 'ZPR_STATUS_VH',
//    element : 'DomvalueL'
//  } }]
//  status     : ymm_pr_status_de;
//  @Semantics.amount.currencyCode : 'currency'
//  total_value : wrbtr;
//  currency  : waers;
//  createdby : ernam;
//  @ObjectModel.readOnly: true
//  created_on : erdat;

}
```

```
@EndUserText.label: 'Approvel Abstract Entity'
@ObjectModel.resultSet.sizeCategory: #XS
@ObjectModel.dataCategory: #VALUE_HELP
define abstract entity Zabs_entity1

//  with parameters parameter_name : parameter_type
{
//  @Consumption.valueHelpDefinition: [{ entity: {
//      name : 'Zabs_entity_Approval',
//      element: 'Approval'
//  } }]
@Consumption.valueHelpDefinition: [{ entity: {
name: 'ZPR_STATUS_VH',
element: 'DomvalueL'
} }]
key Status : ymm_pr_status_de;
}
```

```
@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Help For Company Codes'
@Metadata.ignorePropagatedAnnotations: true
define view entity Zccode_vh
as select from t001
{
key bukrs as Bukrs,
butxt as Butxt,
ort01 as Ort01,
land1 as Land1

}
```

```
@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Material Help'
@Metadata.ignorePropagatedAnnotations: true
define view entity Zmaterial_VH
as select from makt

{
//      @ObjectModel.text.element: [ 'Maktx' ]
key matnr as Matnr,
maktx as Maktx
}
where
matnr > '000000000000100100'
and matnr < '000000000000100122'
```

metadata

```
@Metadata.layer:  #CORE
@UI.headerInfo: {
typeName: 'PR Header',
typeNamePlural: 'Purchase Requisitions',
typeImageUrl: 'sap-icon://activity-items',

//    imageUrl: '',
title: {
type: #STANDARD,
label: 'Purchase Requisition',
value: 'PrId'
}
}
annotate entity YP_mm_pr_hdr with
{

@UI.facet: [
{ id: 'Header1',
type: #DATAPOINT_REFERENCE,
purpose: #HEADER,
targetQualifier: 'h1'
},
{ id: 'Header2',
type: #DATAPOINT_REFERENCE,
purpose: #HEADER,
targetQualifier: 'h2'
},
{ id: 'Header3',
type: #DATAPOINT_REFERENCE,
purpose: #HEADER,
targetQualifier: 'h3'
},
{
id: 'PR_collections',
purpose: #STANDARD,
position: 10 ,
label: 'Purchase Requisition',
type: #COLLECTION
},
{

id: 'PR_header',
parentId: 'PR_collections',
purpose: #STANDARD,
position: 10 ,
label: 'PR Header',
type: #IDENTIFICATION_REFERENCE,
targetQualifier: 'PR_Header'
},
{
id: 'PR_Basic',
parentId: 'PR_collections',
purpose: #STANDARD,
position: 20 ,
label: 'Basic Data',
type: #IDENTIFICATION_REFERENCE,
targetQualifier: 'PR_basic'
},
{
id: 'PR_Item',
purpose: #STANDARD,
position: 20 ,
label: 'Purchase Requesition Item',
type: #LINEITEM_REFERENCE,
targetElement: '_PR_Item'
},{
id: 'PO_Header',
purpose: #STANDARD,
position: 30 ,
label: 'Purchase Order Header',
type: #LINEITEM_REFERENCE,
targetElement: '_PO_Header'
}
]
//  @UI.hidden: true
@UI.lineItem: [{ type: #FOR_ACTION,dataAction: 'Copy_PR',label: 'Copy
PR' }]
Sid;
@UI : { lineItem: [{ position: 10},
{type: #FOR_ACTION ,dataAction: 'send_popup',label: 'Send Mail' },
{ type: #FOR_INTENT_BASED_NAVIGATION,semanticObject: 'show'  }],

identification: [{ position: 10 ,qualifier: 'PR_Header'},{ type:
#FOR_ACTION,dataAction: 'Copy_PR',label: 'Copy PR' }],
selectionField: [{ position: 10 }],
dataPoint : { qualifier: 'h1',title: 'Requisition Document' } }
@EndUserText.label: 'Requisition Document'
PrId;
@UI : { lineItem: [{ position: 20 }],
identification: [{ position: 20 ,qualifier: 'PR_Header'}] }
PrType;
@UI : { lineItem: [{ position: 30 }],
dataPoint : { qualifier: 'h2',title: 'Requestor' },
identification: [{ position: 30 ,qualifier: 'PR_Header'}] }
@EndUserText.label: 'Requestor'
Requester;
@UI : { lineItem: [{ position: 40 }],
textArrangement: #TEXT_ONLY,
identification: [{ position: 40 ,qualifier: 'PR_Header'}] }
Plant;
@UI : { lineItem: [{ position: 50 ,importance: #HIGH,criticality:
'Status_color',criticalityRepresentation: #WITHOUT_ICON}],
textArrangement: #TEXT_ONLY,
identification: [{ position: 50 ,qualifier: 'PR_Header',importance:
#HIGH,criticality: 'Status_color'}] }
Status;
@UI : { lineItem: [{ position: 60 }],
identification: [{ position: 60 ,qualifier: 'PR_Header'}] }
TotalValue;
@UI : { lineItem: [{ position: 70 }],
identification: [{ position: 70 ,qualifier: 'PR_Header'}] }

Currency;
@UI : { lineItem: [{ position: 80 ,importance: #HIGH,criticality:
'status_mail_color'}]}
@EndUserText.label: 'Status mail'
sendmail;
@UI.identification: [{ position: 80 ,qualifier: 'PR_Header'}]
@EndUserText.label: 'Plant Location'
Location;
@UI : { identification: [{ position: 10,qualifier: 'PR_basic'  }] }
CreatedBy;
//  @UI : { identification: [{ position: 20,qualifier:
'PR_basic'  }] }
@UI.hidden: true
CreatedOn;
@UI : { identification: [{ position: 30,qualifier: 'PR_basic'  }] }
LocallastchangdBy;
@UI : { identification: [{ position: 30,qualifier: 'PR_basic'  }] }
LocallastchangdAt;

@UI.dataPoint : { title: 'Remaining Budget',qualifier: 'h3',
targetValueElement: 'total_Item_sum'   ,visualization: #PROGRESS  }
Remain_amt;

}
```

```
@Metadata.layer:  #CORE
@UI.headerInfo: {
typeName: 'PR Item',
typeNamePlural: 'PR Items',
typeImageUrl: 'sap-icon://bullet-text',
//    imageUrl: '',
title: {
type: #STANDARD
//        label: 'Purchase Requisition',
//        value: 'PrId'
}
}
@Search.searchable: true
annotate entity YP_MM_PR_ITM with

{
@UI.facet: [
{ id: 'Header1',
type: #DATAPOINT_REFERENCE,
purpose: #HEADER,
targetQualifier: 'h1'
},
{ id: 'Header2',
type: #DATAPOINT_REFERENCE,
purpose: #HEADER,
targetQualifier: 'h2'
},
{
id: 'pritemcollection',
type: #COLLECTION,
label: 'Purchase Requisition',
position: 10
},
{

id: 'PR_Item',
parentId: 'pritemcollection',
purpose: #STANDARD,
position: 10 ,
label: 'Requisition Item',
type: #IDENTIFICATION_REFERENCE,
targetQualifier: 'pritem'
},
{
id: 'vendor',
parentId: 'pritemcollection',
purpose: #STANDARD,
position: 20 ,
label: 'Vendor Details',
type: #IDENTIFICATION_REFERENCE,
targetQualifier: 'vendor'
}]
//  @UI.hidden: true
@UI.lineItem: [{ type: #FOR_ACTION ,dataAction: 'Copy_PR_Item',label: '' }]
Sid;
@UI : { lineItem: [{ position: 10 }],
identification: [{ position: 10,qualifier: 'pritem' }],
dataPoint : { qualifier: 'h1',title: 'Purchase Reuisition Doc' },
selectionField: [{ position: 10 }] }
@Search.defaultSearchElement: true
@Search.fuzzinessThreshold: 0.8
PrId;
@UI : { lineItem: [{ position: 20 }],
identification: [{ position: 20,qualifier: 'pritem' }],
selectionField: [{ position: 20 }] }
@Search.defaultSearchElement: true
@Search.fuzzinessThreshold: 0.8
ItemNo;
@UI : { lineItem: [{ position: 30 }],
identification: [{ position: 30,qualifier: 'pritem' }] }
Material;
@UI : { lineItem: [{ position: 40 }],
identification: [{ position: 40,qualifier: 'pritem',hidden: #( isHide ) }] }
Qty;
@UI : { identification: [{ position: 50,qualifier: 'pritem',hidden: #( isHide ) }] }
Uom;
@UI : { lineItem: [{ position: 60 }],
identification: [{ position: 60 ,qualifier: 'pritem'}] }
Price;
//  @UI : { lineItem: [{ position: 70 }],
//  identification: [{ position: 70 }] }
//  ItemValue;
@UI : { lineItem: [{ position: 80 ,importance: #HIGH}],
textArrangement: #TEXT_ONLY,
identification: [{ position: 80,qualifier: 'vendor' }] }
//  @UI.dataPoint:{ qualifier: 'h2' ,title: 'Vendor' }
Vendor;
@UI : { lineItem: [{ position: 90}],
identification: [{ position: 90 ,qualifier: 'vendor'}] }
DeliveryDate;
@UI : { lineItem: [{ position: 70}],
identification: [{ position:100 ,qualifier: 'vendor'}] }
@EndUserText.label: 'Total Amount'
total;
@UI.hidden: true
Currency;
@UI.dataPoint:{ qualifier: 'h2' ,title: 'Requested By' }
requestor;
/* Associations */
//  _PR_Header;

}
```

```
@Metadata.layer: #CORE
@UI.headerInfo: {
typeNamePlural: 'Purchase Requsition'
}
@UI.presentationVariant: [{
sortOrder: [
{ by: 'PrId', direction: #DESC }
]
}]
annotate entity YP_PR_Approver with
{
@UI.facet: [{
id: 'PR',
purpose: #STANDARD,
position: 10 ,
importance: #HIGH,
label: 'Purchase Requisition',
type: #IDENTIFICATION_REFERENCE
}]
@UI.hidden: true
Sid;
@UI.lineItem: [{ position: 10 },
{ type: #FOR_ACTION,dataAction: 'Reject_status',label: ' Reject ' },
{ type: #FOR_ACTION,dataAction: 'Approve_status',label: ' Approve ' }

]

@UI.selectionField: [{ position: 10 }]
@UI.identification: [{ position: 10 }]
PrId;
@UI.lineItem: [{ position: 20 }]
```

```
   //  @UI.identification: [{ position: 20 }]
   PrType;
   @UI.lineItem: [{ position: 30 }]
   @EndUserText.label: 'Requested By'
   //  @UI.identification: [{ position: 30 }]
   Requester;
   @UI.lineItem: [{ position: 40 }]
   //  @UI.identification: [{ position: 40 }]
   Plant;
   @UI.lineItem: [{ position: 50,criticality: 'Status_color',
   criticalityRepresentation: #WITHOUT_ICON }]
   @UI.textArrangement: #TEXT_ONLY
   @UI.identification: [{ position: 50,criticality: 'Status_color',criticalityRepresentation:
   #WITHOUT_ICON }]
   Status;
   @UI.lineItem: [{ position: 60 }]
   TotalValue;
   @UI.hidden: true
   Currency;
   @UI.hidden: true
   //  @UI.identification: [{ position: 70 }]
   CreatedBy;
   @UI.hidden: true
   //  @UI.identification: [{ position: 80 }]
   CreatedOn;
   @UI.hidden: true
   LocallastchangdBy;
   @UI.hidden: true
   LocallastchangdAt;
   @UI.lineItem: [{ position: 70,criticality: 'status_mail_color' }]
   //  @UI.identification: [{ position:90 }]
   @EndUserText.label: 'Received Mail'
   sendmail;


   }




@Metadata.layer: #CORE
@UI.headerInfo: {
typeName: 'PO',
typeNamePlural: 'Purchase Documents',
typeImageUrl: 'https://cdn-icons-png.flaticon.com/128/10329/10329719.png',
//    imageUrl: '',
title: {  value: 'Poid'  }
}
@Search.searchable: true
annotate entity yP_po_hdr with
{

@UI.facet: [{
id: 'Header1',
position: 10,
purpose: #HEADER,
label: 'Purchase Requisition',
type: #DATAPOINT_REFERENCE,
targetQualifier: 'h1'
},{
id: 'Header2',
position: 20,
purpose: #HEADER,
label: 'Purchase Document',
type: #DATAPOINT_REFERENCE,
targetQualifier: 'h2'
},{
id: 'POcollections',
position: 10,
label: 'Purchase Document',
type: #COLLECTION
},
{
id: 'POHeader',
parentId: 'POcollections',
position: 10,
label: 'PO Header',
type: #IDENTIFICATION_REFERENCE,
targetQualifier: 'PO_header'
},
{
id: 'Basic',
parentId: 'POcollections',
position: 20,
label: 'Suppiers Data',
type: #IDENTIFICATION_REFERENCE,
targetQualifier: 'Basic'
},
{
id: 'PO_Item',
position: 30,
purpose: #STANDARD,
label: 'Purchase Document Item',
type: #LINEITEM_REFERENCE,
targetElement: '_PO_Item'
}]
@UI.hidden: true
@UI.lineItem: [{ type: #FOR_ACTION,dataAction: 'Copy_PO',label: '' }]
Sid;
@UI :{ lineItem: [{ position: 10}],
identification: [{ position: 10,qualifier: 'PO_header' }],

dataPoint :{ qualifier: 'h2' }
}
@EndUserText.label: 'Document Number'
@Search.defaultSearchElement: true
@Search.fuzzinessThreshold: 0.8
Poid;
@UI :{
identification: [{ position: 20,qualifier: 'PO_header' }],
dataPoint :{ qualifier: 'h1' }
}
@EndUserText.label: 'Requisition Number'
@Search.defaultSearchElement: true
@Search.fuzzinessThreshold: 0.8
Prid;
@UI :{ lineItem: [{ position: 21 }],
identification: [{ position: 21,qualifier: 'PO_header' }] }
Porg;
@UI :{ lineItem: [{ position: 22 }],
identification: [{ position: 22,qualifier: 'PO_header' }] }
Pgroup;
@UI :{ lineItem: [{ position: 23 }],
identification: [{ position: 23,qualifier: 'PO_header' }] }
Ccode;
//  @UI :{ lineItem: [{ position: 25 }],
//  identification: [{ position: 25,qualifier: 'PO_header' }] }
//  Status;
@UI :{ lineItem: [{ position: 24 }],
identification: [{ position: 24,qualifier: 'PO_header' }] }
@EndUserText.label: 'Budget'
Totalvalue;
@UI :{
identification: [{ position: 26,qualifier: 'PO_header' }] }
Currency;
@UI :{
identification: [{ position: 30,qualifier: 'Basic' }] }
@EndUserText.label: 'Supplier'
Vendor;
@UI :{ lineItem: [{ position: 40 }],
identification: [{ position: 40,qualifier: 'Basic' }] }
@EndUserText.label: 'Order Date'
Docdate;
//  @UI :{
//  identification: [{ position: 50,qualifier: 'Basic' }] }
//  Totalvalue;
//  @UI :{ identification: [{ position: 60,qualifier: 'Basic' }] }
//  Currency;
@UI :{ identification: [{ position: 60,qualifier: 'PO_header' }] }
Createdby;
//  @UI :{ identification: [{ position: 70,qualifier: 'PO_header' }] }
//  Createdon;
@UI.hidden: true
Locallastchangdby;
@UI.hidden: true
Locallastchangdat;


}
```

```
@Metadata.layer: #CORE
@UI.headerInfo: {
typeName: 'Item',
typeNamePlural: 'Purchase Items',
typeImageUrl: 'https://user-images.githubusercontent.com/12494082/115613230-d25d9380-a2f4-11eb-9082-0de47de7b3bc.png',
//    imageUrl: '',
title: {
type: #STANDARD,
label: 'Purchase Doc Item'
//        value: ''
}
}
annotate entity YP_PO_ITM with
{

@UI.facet: [
//  {
//      id: 'h1',
//      purpose: #HEADER,
//      position: 10,
//      type: #DATAPOINT_REFERENCE,
//      targetQualifier: 'h1'
//  },
{
id: 'h2',
purpose: #HEADER,
position: 20,
type: #DATAPOINT_REFERENCE,
targetQualifier: 'h2'
},
{ id: 'POItem',
type: #IDENTIFICATION_REFERENCE,
label: 'Purchase Item'

}]
//  @UI.hidden: true
@UI.lineItem: [{ type: #FOR_ACTION,dataAction: 'Copy_POItem',label: '' }]
Sid;
@UI:{
identification: [{ position: 10  }]
//  dataPoint:{ title: 'Requisition',qualifier: 'h1'}
}
@UI.hidden: true
Prid;
@UI:{
//  identification: [{ position: 20  }],
dataPoint:{ title: 'Purchase Document',qualifier: 'h2'}}
Poid;
@UI:{ lineItem: [{ position: 30 }],
identification: [{ position: 30  }] }
@EndUserText.label: 'Item Number'

Itemno;
@UI:{ lineItem: [{ position: 40 }],
identification: [{ position: 40  }] }
Material;
@UI:{ lineItem: [{ position: 50 }],
identification: [{ position: 50  }] }
Mtext;
@UI:{ lineItem: [{ position: 60 }],
identification: [{ position: 60  }] }
Prrid;
@UI:{ lineItem: [{ position: 70 }],
identification: [{ position: 70  }] }
Pritemno;
@UI:{ lineItem: [{ position: 80 }],
identification: [{ position: 80  }] }
Qty;
@UI:{ lineItem: [{ position: 90 }],
identification: [{ position: 90  }] }
Uom;
@UI:{ lineItem: [{ position: 100 }],
identification: [{ position: 100  }] }
Price;
@UI:{ lineItem: [{ position: 110 }],
identification: [{ position: 110  }] }
Itemvalue;
@UI:{ lineItem: [{ position: 120 }],
identification: [{ position: 120  }] }
Deliverydate;
@UI.hidden: true
Currency;


}
```

BDEF

```
unmanaged implementation in class zbp_i_mm_pr_hdr unique;
strict ( 2 );

with draft;

define behavior for YI_MM_PR_HDR alias PR_Header
implementation in class zcl_pr_hdr unique
draft table ypr_hdr_d
//late numbering
early numbering
lock master total etag LocallastchangdAt
authorization master ( instance )
etag master LocallastchangdAt
{
create;
update;
delete;
association _PR_Item { create; with draft; }
association _PO_Header { create ( features : instance ); with draft; }

field ( readonly ) Sid, PrId, Status, CreatedBy, CreatedOn,
LocallastchangdAt, LocallastchangdBy;
field ( features : instance, mandatory ) Requester, PrType, Plant;
field ( mandatory ) TotalValue;
field ( readonly ) Location;
```

```
draft action Activate optimized;
draft action Discard;
draft action Edit;
draft action Resume;
draft determine action Prepare
{
validation ( always ) PR_Item~validate_netprice;
validation ( always ) Check_data;
validation ( always ) validate_plant;
validation ( always ) PO_Header~PurhcaseORG_validate;
validation ( always ) PO_Header~PurhcaseGroup_validate;
validation ( always ) PO_Header~CCode_validate;
validation PO_Item~material_vaidate;
}


factory action Copy_PR [1];

action ( features : instance ) send_popup parameter Zabs_entity result
[1] $self;
determination FillPlantLocation on modify { field Plant; }
side effects
{ field Plant affects field Location;
action Reject_status affects field Status;
action Approve_status affects field Status;
action send_popup affects field sendmail; }

validation Check_data on save { create; field Plant, Requester; }
validation validate_plant on save { create; field Plant; }

action ( features : instance ) Approve_status result [1] $self;
action ( features : instance ) Reject_status result [1] $self;


mapping for ymm_pr_hdr
{
Sid               = sid;
PrId              = pr_id;
PrType            = pr_type;
Requester         = requester;
Plant             = plant;
Status            = status;
TotalValue        = total_value;
Currency          = currency;
sendmail          = sendmail;
Location          = location;
CreatedBy         = created_by;
CreatedOn         = created_on;
LocallastchangdAt = locallastchangd_at;
LocallastchangdBy = locallastchangd_by;
}
}

define behavior for YI_MM_PR_ITM alias PR_Item
implementation in class zcl_pr_itm unique
draft table ypr_itm_d
late numbering
lock dependent by _PR_Header
authorization dependent by _PR_Header
etag dependent by _PR_Header
{
update ( features : instance );
delete;
field ( readonly ) Sid, PrId, ItemNo, DeliveryDate;
validation validate_netprice on save { create; update; field Price; }

factory action Copy_PR_Item [1];

association _PR_Header { with draft; }
mapping for ymm_pr_itm
{
Sid          = sid;
PrId         = pr_id;
ItemNo       = item_no;
Material     = material;
Qty          = qty;
Uom          = uom;
Price        = price;
ItemValue    = item_value;
Vendor       = vendor;
DeliveryDate = delivery_date;
}
}

define behavior for yi_po_hdr alias PO_Header
implementation in class zcl_po_hdr unique
draft table ypo_hdr_d
late numbering
//early numbering
lock dependent by _PR_Header
authorization dependent by _PR_Header
etag master Locallastchangdat
{
update;
delete;
```

```
   field ( readonly ) Sid, PrId, Poid, Createdby, Createdon,
   Locallastchangdat, Locallastchangdby
   ;

   field ( mandatory ) Porg, Pgroup, Ccode;

   determination provide_docdate on modify { create; update; field
   Prid; }
   side effects { field Prid affects field Docdate; }

   validation PurhcaseORG_validate on save { create; update; field
   Porg; }
   validation PurhcaseGroup_validate on save { create; update; field
   Pgroup; }
   validation CCode_validate on save { create; update; field Pgroup; }


   factory action ( authorization : instance ) Copy_PO [1];

   association _PR_Header { with draft; }
   association _PO_Item { create; with draft; }
   mapping for ymm_po_hdr
   {
   Sid               = sid;
   Poid              = poid;
   Prid              = prid;
   Vendor            = vendor;
   Docdate           = docdate;
   Pgroup            = pgroup;
   Porg              = porg;
   Ccode             = ccode;
   Prid2             = prid2;
   Status            = status;
   Totalvalue        = totalvalue;
   Currency          = currency;
   Createdby         = createdby;
   Createdon         = createdon;
   Locallastchangdat = locallastchangdat;
   Locallastchangdby = locallastchangdby;
   }
   }


   define behavior for yi_po_itm alias PO_Item
   implementation in class zcl_po_itm unique
   draft table ypo_itm_d
   late numbering
   //early numbering
   lock dependent by _PR_Header
   authorization dependent by _PR_Header
   etag dependent by _PO_Header
   {
   update;
   delete;
   field ( readonly ) Sid, Prid, Poid, Itemno, Deliverydate;
   field ( mandatory ) Material, Prrid, Pritemno;


   validation material_vaidate on save { create; update; field
   Material; }

   determination AutoFillData on modify { update; field Pritemno; }
   side effects
   { field Pritemno affects field Qty; }

   factory action Copy_POItem [1];

   association _PO_Header { with draft; }
   association _PR_Header { with draft; }
   mapping for ymm_po_itm
   {
   Sid          = sid;
   Prid         = prid;
   Poid         = poid;
   Itemno       = itemno;
   Material     = material;
   Mtext        = mtext;
   Prrid        = prrid;
   Pritemno     = pritemno;
   Qty          = qty;
   Uom          = uom;
   Price        = price;
   Itemvalue    = itemvalue;
   Deliverydate = deliverydate;
   }
   }
```

```
projection;
strict ( 2 );
use draft;
use side effects;
```

```
define behavior for YP_mm_pr_hdr alias PR_Header
use etag
{
use create;
use update;
use delete;

use action Activate;
use action Discard;
use action Edit;
use action Resume;
use action Prepare;

use action send_popup;
use action Copy_PR;

use association _PR_Item { create; with draft; }
use association _PO_Header { create; with draft; }
}

define behavior for YP_MM_PR_ITM alias PR_Item
{
use update;
use delete;

use association _PR_Header { with draft; }

use action Copy_PR_Item;
}

define behavior for yP_po_hdr alias PO_Header
{
use update;
use delete;

use association _PR_Header { with draft; }
use association _PO_Item { create; with draft; }

use action Copy_PO;

}

define behavior for YP_PO_ITM alias PO_Item
{
use update;
use delete;

use association _PO_Header { with draft; }
use association _PR_Header { with draft; }

use action Copy_POItem;
}



projection;
strict ( 2 );
//use draft;
use side effects;
define behavior for YP_PR_Approver alias PR_Approver
use etag
{
//  use create;
//  use update;
//  use delete;

use action Reject_status;
use action Approve_status;

//  use action Activate;
//  use action Discard;
//  use action Edit;
//  use action Resume;
//  use action Prepare;
//
//  use association _PR_Item;
//  use association _PO_Header;

}
```

SD

```
@EndUserText.label: 'Service def  For PR & PO'
define service YSD_PR_PO {
expose YP_mm_pr_hdr as PR_Header;
expose YP_MM_PR_ITM as PR_Item;
expose yP_po_hdr    as PO_Header;
expose YP_PO_ITM    as PO_Item;
}
```

```
@EndUserText.label: 'Service def  Approve PR'
define service YSD_PR_Approver {
expose YP_PR_Approver as PR_Approver;
}
```

ABAP CLASSES

```
CLASS lhc_PR_Header DEFINITION INHERITING FROM cl_abap_behavior_handler.
  PRIVATE SECTION.

    METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
      IMPORTING keys REQUEST requested_authorizations FOR PR_Header RESULT result.

    METHODS create FOR MODIFY
      IMPORTING entities FOR CREATE PR_Header.

    METHODS update FOR MODIFY
      IMPORTING entities FOR UPDATE PR_Header.

    METHODS delete FOR MODIFY
      IMPORTING keys FOR DELETE PR_Header.

    METHODS read FOR READ
      IMPORTING keys FOR READ PR_Header RESULT result.

    METHODS lock FOR LOCK
      IMPORTING keys FOR LOCK PR_Header.

    METHODS rba_Pr_item FOR READ
      IMPORTING keys_rba FOR READ PR_Header\_Pr_item FULL result_requested RESULT result LINK
association_links.

    METHODS cba_Pr_item FOR MODIFY
      IMPORTING entities_cba FOR CREATE PR_Header\_Pr_item.

    METHODS get_instance_features FOR INSTANCE FEATURES
      IMPORTING keys REQUEST requested_features FOR PR_Header RESULT result.

    METHODS send_popup FOR MODIFY
      IMPORTING keys FOR ACTION PR_Header~send_popup RESULT result.

    METHODS FillPlantLocation FOR DETERMINE ON MODIFY
      IMPORTING keys FOR PR_Header~FillPlantLocation.
    METHODS ChecK_data FOR VALIDATE ON SAVE
      IMPORTING keys FOR PR_Header~Check_data.
    METHODS validate_plant FOR VALIDATE ON SAVE
      IMPORTING keys FOR PR_Header~validate_plant.
    METHODS rba_Po_header FOR READ
      IMPORTING keys_rba FOR READ PR_Header\_Po_header FULL result_requested RESULT result LINK
association_links.

    METHODS cba_Po_header FOR MODIFY
      IMPORTING entities_cba FOR CREATE PR_Header\_Po_header.
    METHODS Approve_status FOR MODIFY
      IMPORTING keys FOR ACTION PR_Header~Approve_status RESULT result.

    METHODS Reject_status FOR MODIFY
      IMPORTING keys FOR ACTION PR_Header~Reject_status RESULT result.
    METHODS Copy_PR FOR MODIFY
      IMPORTING keys FOR ACTION PR_Header~Copy_PR.

*    METHODS earlynumbering_cba_Po_header FOR NUMBERING
*      IMPORTING entities FOR CREATE PR_Header\_Po_header.

*    METHODS draft_status FOR DETERMINE ON MODIFY
*      IMPORTING keys FOR PR_Header~draft_status.


    METHODS earlynumbering_create FOR NUMBERING
      IMPORTING entities FOR CREATE PR_Header.

ENDCLASS.

CLASS lhc_PR_Header IMPLEMENTATION.

  METHOD get_instance_authorizations.
  ENDMETHOD.

  METHOD create.
    ycl_helper_pr_po=>get_instance( )->create_pr_hdr(
      EXPORTING
        entities = entities
      CHANGING
        mapped   = mapped
        failed   = failed
        reported = reported
    ).
  ENDMETHOD.

  METHOD update.
```

```
    ycl_helper_pr_po=>get_instance( )->update_pr_hdr(
      EXPORTING
        entities = entities
      CHANGING
        mapped   = mapped
        failed   = failed
        reported = reported
    ).
  ENDMETHOD.

  METHOD delete.
    ycl_helper_pr_po=>get_instance( )->delete_pr_hdr(
      EXPORTING
        keys     = keys
      CHANGING
        mapped   = mapped
        failed   = failed
        reported = reported
    ).
  ENDMETHOD.

  METHOD read.
    ycl_helper_pr_po=>get_instance( )->read_pr_hdr(
      EXPORTING
        keys     = keys
      CHANGING
        result   = result
        failed   = failed
        reported = reported
    ).
  ENDMETHOD.

  METHOD lock.
  ENDMETHOD.

  METHOD rba_Pr_item.


    DATA(lv_sid) = keys_rba[ 1 ]-Sid.
    DATA(lv_pid) = keys_rba[ 1 ]-PrId.
*    DATA(lv_itumnum) = keys[ 1 ]-ItemNo.

    SELECT * FROM ymm_pr_itm
    WHERE sid = @lv_sid AND pr_id = @lv_pid
    INTO TABLE @DATA(lt_read).

*    LOOP AT keys INTO DATA(ls_key).
*      result = VALUE #( FOR ls_data IN lt_read
*      ( %is_draft = ls_key-%is_draft
*       %data = CORRESPONDING #( ls_data MAPPING Sid = sid PrId = pr_id
*                                                ItemNo = item_no
*                                                ItemValue = item_value
*                                                DeliveryDate = delivery_date
*                                 ) ) ).
*    ENDLOOP.\

    result = CORRESPONDING #( lt_read MAPPING Sid = sid PrId = pr_id
                                            ItemNo = item_no
                                            ItemValue = item_value
                                            DeliveryDate = delivery_date
                             ).


  ENDMETHOD.

  METHOD cba_Pr_item.
    ycl_helper_pr_po=>get_instance( )->cba_pr_item(
      EXPORTING
        entities_cba = entities_cba
      CHANGING
        mapped       = mapped
        failed       = failed
        reported     = reported
    ).
  ENDMETHOD.

  METHOD earlynumbering_create.
    ycl_helper_pr_po=>get_instance( )->earlynumbering_create(
      EXPORTING
        entities = entities
      CHANGING
        mapped   = mapped
        failed   = failed
        reported = reported
    ).
  ENDMETHOD.


  METHOD get_instance_features.
    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
      ENTITY PR_Header
      FIELDS ( Status )
      WITH CORRESPONDING #( keys )
      RESULT DATA(lt_header).
```

```
*  Fileds Are Readonly During Update

    IF lt_header IS NOT INITIAL.
      DATA(lv_pnum) = lt_header[ 1 ]-PrId.
      DATA(lv_sid) = lt_header[ 1 ]-Sid.
      SELECT SINGLE * FROM ymm_pr_hdr WHERE sid = @lv_sid AND pr_id = @lv_pnum INTO @DATA(wa_data).
      IF sy-subrc = 0.
        result = VALUE #( FOR ls_data IN lt_header
                        LET lv_let = if_abap_behv=>perm-f-read_only_update
                        IN ( %tky = ls_data-%tky Sid = ls_data-Sid PrId = ls_data-PrId
                             %action-send_popup = COND #( WHEN ls_data-Status = 'A' OR ls_data-Status = 'R'
                                                                THEN if_abap_behv=>fc-o-disabled
                                                                ELSE if_abap_behv=>fc-o-enabled )
                             %features-%field-Plant = lv_let
                             %features-%field-PrType = lv_let
                             %features-%field-Requester = lv_let
*                             %assoc-_PO_Header = COND #( WHEN ls_data-Status = 'R' OR ls_data-Status = 'S'
*                                                                 THEN if_abap_behv=>fc-o-disabled
*                                                                 ELSE if_abap_behv=>fc-o-enabled )
) ).
      ENDIF.
    ENDIF.

    SELECT * FROM ymm_pr_hdr INTO TABLE @DATA(lt_data1).
    lt_header = CORRESPONDING #( lt_data1 MAPPING PrId = pr_id ).

**To Hide Send mail action  based on Status
    IF  requested_features-%action-send_popup = 01.
      IF lt_header IS NOT INITIAL.
        LOOP AT lt_header INTO DATA(ls_data1).
          APPEND VALUE #( %tky = ls_data1-%tky
                           sid  = ls_data1-sid
                           prid = ls_data1-PrId
                            %action-send_popup = COND #( WHEN ls_data1-Status = 'A' OR ls_data1-Status = 'R'
                                                               THEN if_abap_behv=>fc-o-disabled
                                                             ELSE if_abap_behv=>fc-o-enabled )
                              )  TO result.
          CLEAR : ls_data1.


        ENDLOOP.
      ENDIF.
    ENDIF.
*  Actions buttons are Enable or disable For Admin
    IF  requested_features-%action-Approve_status = 01 OR  requested_features-%action-Reject_status = 01.
      LOOP AT lt_header INTO DATA(ls_approve).
        IF ls_approve-sendmail = 'YES'.
          IF ls_approve-Status = 'A' OR ls_approve-Status = 'R'.
            APPEND VALUE #( %tky = ls_approve-%tky
                            %action-Approve_status = if_abap_behv=>fc-o-disabled
                            %action-Reject_status = if_abap_behv=>fc-o-disabled
                          )  TO result.
          ENDIF.
        ELSE.
          APPEND VALUE #( %tky = ls_approve-%tky
            %action-Approve_status = if_abap_behv=>fc-o-disabled
            %action-Reject_status = if_abap_behv=>fc-o-disabled
              )  TO result.
        ENDIF.
      ENDLOOP.
    ENDIF.

*  Disable PO Creation based on PR Reject status
*    IF  requested_features-%assoc-_PO_Header = 01.
*      LOOP AT lt_header INTO DATA(ls_header).
*        IF ls_header-Status = 'R' OR ls_header-Status = 'S'.
*          result = VALUE #( ( %tky = ls_header-%tky
*                              %assoc-_PO_Header = if_abap_behv=>fc-o-disabled   ) ).
*        ENDIF.
*      ENDLOOP.
*    ENDIF.

  ENDMETHOD.

  METHOD send_popup.
    DATA : tt_keys_header    TYPE  TABLE FOR ACTION IMPORT yi_mm_pr_hdr\\pr_header~send_popup,
           tt_result_header  TYPE TABLE FOR ACTION RESULT yi_mm_pr_hdr\\pr_header~send_popup,
           tt_mapped_early   TYPE RESPONSE FOR MAPPED EARLY yi_mm_pr_hdr,
           tt_reported_early TYPE RESPONSE FOR REPORTED EARLY yi_mm_pr_hdr,
           tt_falied_early   TYPE  RESPONSE FOR FAILED EARLY yi_mm_pr_hdr.
    LOOP AT keys INTO DATA(ls_key).
      DATA(lv_mail) =  ls_key-%param-email.
      SPLIT lv_mail AT '@' INTO DATA(user) DATA(domain).
      domain = |@{ to_lower( domain ) }|.
      IF domain NE '@gmail.com'.
        reported-pr_header = VALUE #( ( %key = ls_key-%key
                                        %msg = new_message_with_text(
                                                                    severity = if_abap_behv_message=>severity-
error
                                                                    text     = |Invalid Mail { lv_mail }|
                                                                  )
                                          ) ).
      ENDIF.
    ENDLOOP.
    ycl_helper_mail_send=>get_instance(  )->send_mail(
      EXPORTING
        keys     = keys
      IMPORTING
        result   = tt_result_header
        mapped   = tt_mapped_early
        failed   = tt_falied_early
        reported = tt_reported_early
        ).

    IF tt_reported_early-pr_header IS NOT INITIAL.
      result = VALUE #( FOR ls IN  tt_result_header (  %key = ls-%key %param = CORRESPONDING #( ls ) ) ).
*      mapped = CORRESPONDING #( tt_mapped_early ).
*      failed = tt_falied_early.
*      reported-pr_header = VALUE #( BASE reported-pr_header ( %key = tt_reported_early-pr_header[ 1 ]-%key
*                                                              %msg = new_message_with_text(
*                                                                       severity = if_abap_behv_message=>
severity-success
*                                                                       text     = |Mail Sent Successfully
{ keys[ 1 ]-%param-Name }|
*                                                                     ) ) ).

      APPEND VALUE #( %key = tt_reported_early-pr_header[ 1 ]-%key
                      %msg = new_message_with_text(
                                                   severity = if_abap_behv_message=>severity-success
                                                   text     = |Mail Sent Successfully { keys[ 1 ]-%param-
Name }|
                                                 ) ) TO reported-pr_header.
    ENDIF.


  ENDMETHOD.

*  METHOD draft_status.
**    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
**    ENTITY PR_Header
**    FIELDS ( Status ) WITH CORRESPONDING #( keys )
**    RESULT DATA(lt_header).
**
**    LOOP AT lt_header ASSIGNING FIELD-SYMBOL(<fs_header>).
**
**      SELECT SINGLE * FROM ypr_hdr_d WHERE prid = @<fs_header>-PrId  INTO @DATA(ls_draft).
**      IF sy-subrc EQ 0.
**        <fs_header>-Status = 'D'.
**        MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
**        ENTITY PR_Header
**        UPDATE FIELDS ( Status )
**        WITH VALUE #( (  %is_draft = <fs_header>-%is_draft
**                         %data = CORRESPONDING #( <fs_header> ) ) ).
**
**
**      ENDIF.
**    ENDLOOP.
*
*  ENDMETHOD.

  METHOD FillPlantLocation.

    READ ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PR_Header
    FIELDS ( Plant )
    WITH CORRESPONDING #( keys )
    RESULT DATA(lt_pr_header).

    LOOP AT lt_pr_header ASSIGNING FIELD-SYMBOL(<fs_header>).

      SELECT SINGLE FROM t001w FIELDS stras
      WHERE werks = @<fs_header>-Plant
      INTO @<fs_header>-Location.

      MODIFY ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
      ENTITY PR_Header
      UPDATE FIELDS ( Location )
      WITH CORRESPONDING #( lt_pr_header ).

      reported-pr_header = VALUE #( ( %tky = <fs_header>-%tky
                                      %msg = new_message_with_text(
                                               severity = if_abap_behv_message=>severity-success
                                               text     = 'Storage or Plant Location Filled'
                                             ) ) ).

    ENDLOOP.

  ENDMETHOD.

  METHOD CheckK_data.
    READ ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PR_Header
    ALL FIELDS WITH
    CORRESPONDING #( keys )
    RESULT DATA(lt_pr_header).

    READ TABLE lt_pr_header INTO DATA(ls_header) INDEX 1.
    IF sy-subrc EQ 0.
    reported-pr_header = VALUE #( ( %tky = ls_header-%tky %state_area = 'ORDER_TYPE')
                                  ( %tky = ls_header-%tky %state_area = 'REQUESTOR' ) ).
    IF ls_header-PrType IS INITIAL.
      failed-pr_header = CORRESPONDING #( lt_pr_header ).
      reported-pr_header = VALUE #( (
      %tky = ls_header-%tky
      %msg = new_message_with_text(
               severity = if_abap_behv_message=>severity-error
               text     = 'Enter Order Type *' )
      %state_area = 'ORDER_TYPE'
      %element-prtype = if_abap_behv=>mk-on
               ) ).
    ELSEIF ls_header-PrType IS NOT INITIAL.
      SELECT SINGLE * FROM ZOrder_type_VH WHERE bsart = @ls_header-PrType INTO @DATA(ls_data2).
      IF sy-subrc NE 0.
        failed-pr_header = CORRESPONDING #( lt_pr_header ).
        reported-pr_header = VALUE #( (
        %tky = ls_header-%tky
        %msg = new_message_with_text(
                 severity = if_abap_behv_message=>severity-error
                 text     = |{ ls_header-PrType } is invalid| )
        %state_area = 'ORDER_TYPE'
        %element-prtype = if_abap_behv=>mk-on
                 ) ).

      ENDIF.
    ENDIF.
    IF ls_header-Requester IS INITIAL.
      failed-pr_header = CORRESPONDING #( BASE ( failed-pr_header ) lt_pr_header ).
      reported-pr_header = VALUE #( BASE reported-pr_header (
      %tky = ls_header-%tky
      %msg = new_message_with_text(
               severity = if_abap_behv_message=>severity-error
               text     = 'Enter Requestor Name *' )
      %state_area = 'REQUESTOR'
       %element-requester = if_abap_behv=>mk-on

               ) ).

    ENDIF.
  ENDIF.

ENDMETHOD.

METHOD validate_plant.
  READ ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PR_Header
    FIELDS ( Plant )
    WITH CORRESPONDING #( keys )
    RESULT DATA(lt_pr_header).

  LOOP AT lt_pr_header INTO DATA(ls_header).
    SELECT SINGLE * FROM t001w
    WHERE werks = @ls_header-Plant
    INTO @DATA(lv_loc).
    IF sy-subrc NE 0.
      failed-pr_header = CORRESPONDING #( lt_pr_header ).
      reported-pr_header = VALUE #( ( %tky = ls_header-%tky
                          %msg = new_message_with_text(
                                   severity = if_abap_behv_message=>severity-error
                                   text     = 'Invalid Plant. Please enter a valid Plant.'
                                 )
                                 %element-plant = if_abap_behv=>mk-on ) ).

    ENDIF.
  ENDLOOP.
ENDMETHOD.

METHOD rba_Po_header.

ENDMETHOD.

METHOD cba_Po_header.
  DATA: lv_max_pr1   TYPE string,
        lv_max_pr2   TYPE string,
        lv_num1      TYPE i,
        lv_num2      TYPE i,
        lv_next_no   TYPE i,
        lv_po_id     TYPE ebeln,
        lv_numc(6)   TYPE n,
        lt_po_header TYPE TABLE OF ymm_po_hdr,
        ls_msg       TYPE bapiret2.

  DATA(lv_prid) = entities_cba[ 1 ]-PrId.

  SELECT FROM ymm_po_hdr FIELDS MAX( poid ) WHERE prid = @lv_prid INTO @lv_max_pr1.
  SELECT FROM ypo_hdr_d FIELDS MAX( poid ) WHERE prid = @lv_prid INTO @lv_max_pr2.

  lv_num1 = COND i(  WHEN lv_max_pr1 IS INITIAL THEN 0 ELSE lv_max_pr1+2(6) ).
  lv_num2 = COND i( WHEN lv_max_pr2 IS INITIAL THEN 0 ELSE lv_max_pr2+2(6) ).

  " Get next number
  lv_next_no = COND i( WHEN lv_num1 > lv_num2 THEN lv_num1 + 1 ELSE lv_num2 + 1 ).
  lv_numc = lv_next_no.
```

```
    LOOP AT entities_cba INTO DATA(ls_header).
      lt_po_header = CORRESPONDING #( ls_header-%target ).
    ENDLOOP.
    LOOP AT lt_po_header ASSIGNING FIELD-SYMBOL(<fs_data>).
      lv_po_id = |PO{ lv_numc }|.
      <fs_data>-poid = lv_po_id.
      <fs_data>-docdate = sy-datum.
      lv_numc += 1.
    ENDLOOP.

    CALL FUNCTION 'YBAPI_PO'
      EXPORTING
        po_create_h = lt_po_header
      IMPORTING
        return      = ls_msg.

    IF ls_msg IS NOT INITIAL.

      reported-po_header = VALUE #( ( Sid = entities_cba[ 1 ]-Sid
                                      Prid = entities_cba[ 1 ]-PrId
                                      Poid = ls_msg-message_v1
                                            %msg = new_message_with_text(
                                               severity = COND #( WHEN ls_msg-type = 'S'
                                                                     THEN   if_abap_behv_message=>severity-
success
                                                                     ELSE if_abap_behv_message=>severity-
error )
                                               text     = COND #( WHEN ls_msg-type = 'S'
                                                                 THEN |PO { ls_msg-message_v1 } Created|
                                                                 ELSE |PO { ls_msg-message_v1 } Failed To
Create| )
                                              ) ) ).
    ENDIF.


  ENDMETHOD.

*  METHOD earlynumbering_cba_Po_header.
*
*    ycl_helper_pr_po=>get_instance( )->earlynumbering_cba_po_header(
*      EXPORTING
*        entities = entities
*      CHANGING
*        mapped   = mapped
*        failed   = failed
*        reported = reported
*    ).
*
*  ENDMETHOD.


  METHOD Approve_status.
    READ ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PR_Header
    FIELDS ( Plant )
    WITH CORRESPONDING #( keys )
    RESULT DATA(lt_pr_header).
    LOOP AT lt_pr_header ASSIGNING FIELD-SYMBOL(<fs_header>).
      <fs_header>-Status = 'A'.

      MODIFY ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
      ENTITY PR_Header
      UPDATE FIELDS ( Status )
      WITH CORRESPONDING #( lt_pr_header )
      MAPPED DATA(lt_mapped)
      FAILED DATA(lt_faliled).

      result = VALUE #( BASE result ( %tky = <fs_header>-%tky %param = CORRESPONDING #( <fs_header> ) ) ).
      mapped-pr_header = VALUE #( BASE mapped-pr_header ( %tky = <fs_header>-%tky ) ).
      reported-pr_header = VALUE #( ( %tky = <fs_header>-%tky
                            %msg = new_message_with_text(
                                     severity = if_abap_behv_message=>severity-success
                                     text     = |{ <fs_header>-PrId } is Approved Successfully|
                                   )       ) ).

    ENDLOOP.

  ENDMETHOD.

  METHOD Reject_status.

    READ ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
      ENTITY PR_Header
      FIELDS ( Plant )
      WITH CORRESPONDING #( keys )
      RESULT DATA(lt_pr_header).
    LOOP AT lt_pr_header ASSIGNING FIELD-SYMBOL(<fs_header>).
      <fs_header>-Status = 'R'.

      MODIFY ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
      ENTITY PR_Header
      UPDATE FIELDS ( Status )
      WITH CORRESPONDING #( lt_pr_header )
      MAPPED DATA(lt_mapped)
      FAILED DATA(lt_faliled).
      result = VALUE #( BASE result ( %tky = <fs_header>-%tky %param = CORRESPONDING #( <fs_header> ) ) ).
      mapped-pr_header = VALUE #( BASE mapped-pr_header ( %tky = <fs_header>-%tky ) ).
      reported-pr_header = VALUE #( ( %tky = <fs_header>-%tky
                            %msg = new_message_with_text(
                                     severity = if_abap_behv_message=>severity-success
                                     text     = |{ <fs_header>-PrId } is Rejected|
                                   )       ) ).

    ENDLOOP.

  ENDMETHOD.

  METHOD Copy_PR.

*    Only PR Header Copy
    DATA : lt_copy  TYPE TABLE FOR CREATE yi_mm_pr_hdr.

    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PR_Header
    ALL FIELDS WITH CORRESPONDING #( keys )
    RESULT DATA(lt_prheader).

    lt_copy = VALUE #( FOR ls IN lt_prheader
    ( %cid = keys[ KEY entity %key = ls-%key ]-%cid
      %is_draft = keys[ KEY entity %key = ls-%key ]-%param-%is_draft
      %data = CORRESPONDING #( ls EXCEPT Sid PrId )
    ) ).

    MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PR_Header
    CREATE FIELDS ( PrType Requester Plant TotalValue Currency Location )
    WITH lt_copy
    MAPPED DATA(lt_mapped).

    mapped-pr_header = lt_mapped-pr_header.


*  Copy Pr Header and Pr Item

*    DATA : lt_prheader TYPE TABLE FOR CREATE yi_mm_pr_hdr,
*           lt_pritem   TYPE TABLE FOR CREATE yi_mm_pr_hdr\_PR_Item..
*
*    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*    ENTITY PR_Header
*    ALL FIELDS WITH CORRESPONDING #( keys )
*    RESULT DATA(lt_header_PR).
*
*    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*    ENTITY PR_Header BY \_PR_Item
*    ALL FIELDS WITH CORRESPONDING #( keys )
*    RESULT DATA(lt_item_PR).
*
*    LOOP AT lt_header_PR ASSIGNING FIELD-SYMBOL(<fs_prheader>).
*
*      APPEND VALUE #( %cid = keys[ KEY entity %key = <fs_prheader>-%key  ]-%cid
*                      %is_draft = <fs_prheader>-%is_draft
*                      %data = CORRESPONDING #( <fs_prheader> EXCEPT sid prid ) )
*             TO lt_prheader ASSIGNING FIELD-SYMBOL(<ls_header>).
*
*      APPEND VALUE #( %cid_ref = <ls_header>-%cid %is_draft = <fs_prheader>-%is_draft )
*             TO lt_pritem ASSIGNING FIELD-SYMBOL(<ls_item>).
*
*      LOOP AT lt_item_PR ASSIGNING FIELD-SYMBOL(<fs_pritem>)
*            USING KEY entity WHERE Sid = <fs_prheader>-Sid AND PrId = <fs_prheader>-PrId.
*
*        APPEND VALUE #( %cid = <ls_header>-%cid && <fs_pritem>-ItemNo
*                        %is_draft = <fs_prheader>-%is_draft
*                        %data = CORRESPONDING #( <fs_pritem> EXCEPT sid prid itemno   )
*                         )
*                TO <ls_item>-%target .
**        <ls_item>-Sid = <ls_header>-Sid.
**        <ls_item>-PrId = <ls_header>-prid.
*
*      ENDLOOP.
*    ENDLOOP.
*
*    MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*      ENTITY PR_Header
*      CREATE FIELDS (  PrType Requester Plant TotalValue Currency Location )
*      WITH lt_prheader
*
*      ENTITY PR_Header
*      CREATE BY \_PR_Item
*      FIELDS ( Material Qty Uom Price Currency Vendor )
*      WITH lt_pritem
*
*      MAPPED DATA(lt_mapped1).
*
*    mapped-pr_header = lt_mapped1-pr_header.
*    mapped-pr_item = lt_mapped1-pr_item.
```

```
  ENDMETHOD.

ENDCLASS.



CLASS lhc_PR_Item DEFINITION INHERITING FROM cl_abap_behavior_handler.
  PRIVATE SECTION.

    METHODS update FOR MODIFY
      IMPORTING entities FOR UPDATE PR_Item.

    METHODS delete FOR MODIFY
      IMPORTING keys FOR DELETE PR_Item.

    METHODS read FOR READ
      IMPORTING keys FOR READ PR_Item RESULT result.

    METHODS rba_Pr_header FOR READ
      IMPORTING keys_rba FOR READ PR_Item\_Pr_header FULL result_requested RESULT result LINK
association_links.

    METHODS validate_netprice FOR VALIDATE ON SAVE
      IMPORTING keys FOR PR_Item~validate_netprice.

    METHODS get_instance_features FOR INSTANCE FEATURES
      IMPORTING keys REQUEST requested_features FOR PR_Item RESULT result.

    METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
      IMPORTING keys REQUEST requested_authorizations FOR pr_item RESULT result.

    METHODS copy_pr_item FOR MODIFY
      IMPORTING keys FOR ACTION pr_item~copy_pr_item.

ENDCLASS.

CLASS lhc_PR_Item IMPLEMENTATION.

  METHOD update.
    ycl_helper_pr_po=>get_instance(  )->update_pr_item(
      EXPORTING
        entities = entities
      CHANGING
        mapped   = mapped
        failed   = failed
        reported = reported
    ).
  ENDMETHOD.

  METHOD delete.
    ycl_helper_pr_po=>get_instance(  )->delete_pr_itm(
      EXPORTING
        keys     = keys
      CHANGING
        mapped   = mapped
        failed   = failed
        reported = reported
    ).
  ENDMETHOD.

  METHOD read.
    DATA(lv_sid) = keys[ 1 ]-Sid.
    DATA(lv_pid) = keys[ 1 ]-PrId.
    DATA(lv_itumnum) = keys[ 1 ]-ItemNo.

    SELECT * FROM ymm_pr_itm
    WHERE sid = @lv_sid AND pr_id = @lv_pid AND item_no = @lv_itumnum
    INTO TABLE @DATA(lt_read).

    LOOP AT keys INTO DATA(ls_key).
      result = VALUE #( FOR ls_data IN lt_read
      ( %is_draft = ls_key-%is_draft
       %data = CORRESPONDING #( ls_data MAPPING Sid = sid PrId = pr_id
                                                ItemNo = item_no
                                                ItemValue = item_value
                                                DeliveryDate = delivery_date
                                 ) ) ).
    ENDLOOP.

  ENDMETHOD.

  METHOD rba_Pr_header.

  ENDMETHOD.

  METHOD validate_netprice.
    DATA: lv_cost  TYPE p,
          lv_total TYPE p,
          lt_read  TYPE TABLE FOR VALIDATION yi_mm_pr_hdr\\pr_item~validate_netprice.

    LOOP AT keys INTO DATA(ls_key).
      IF ls_key-%pid IS NOT INITIAL.
        lt_read = VALUE #( BASE lt_read ( %pid = ls_key-%pid %tky = ls_key-%tky ) ).
      ENDIF.
    ENDLOOP.

    READ ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PR_Item
    FIELDS ( Price )
    WITH CORRESPONDING #( lt_read )
    RESULT DATA(lt_item).

    LOOP AT lt_item INTO DATA(ls_item).
      lv_total += ls_item-Price.
    ENDLOOP.

    SELECT FROM ymm_pr_hdr FIELDS SUM( total_value )
    WHERE pr_id = @ls_item-PrId
    INTO @DATA(lv_sum1).

    SELECT FROM ymm_pr_itm FIELDS SUM( price )
    WHERE sid = @ls_item-Sid AND pr_id = @ls_item-PrId
    INTO @DATA(lv_sum2).

    IF lv_sum1 LT lv_total + lv_sum2 .
      lv_cost = COND #( WHEN lv_sum2 = 0 THEN lv_sum1
                        WHEN lv_sum1 LT lv_sum2 + lv_total  THEN  lv_sum1 - lv_sum2  ).

      failed-pr_item = VALUE #( ( %tky = ls_item-%tky ) ).
      reported-pr_item = VALUE #( ( %tky = ls_item-%tky
                                    %msg = new_message_with_text(
                                             severity = if_abap_behv_message=>severity-error
                                             text     = |Available Budget is { lv_cost } |
                                           ) ) ).
    ENDIF.


  ENDMETHOD.

  METHOD get_instance_features.
    DATA: lv_cost  TYPE p,
          lv_total TYPE p.

    IF requested_features-%update = if_abap_behv=>mk-on.
*      READ ENTITIES OF  yi_mm_pr_hdr IN LOCAL MODE
*        ENTITY PR_Item
*        FIELDS ( Price )
*        WITH CORRESPONDING #( keys )
*        RESULT DATA(lt_item).
*
*      LOOP AT lt_item INTO DATA(ls_item).
*        result = VALUE #( BASE result ( %tky = ls_item-%tky %update = if_abap_behv=>fc-o-
disabled ) ).
*      ENDLOOP.
*
*      LOOP AT lt_item INTO DATA(ls_item).
*        lv_total += ls_item-Price.
*      ENDLOOP.
*
*      SELECT FROM ymm_pr_hdr FIELDS SUM( total_value )
*        WHERE pr_id = @ls_item-PrId
*        INTO @DATA(lv_sum1).
*
*      SELECT FROM ymm_pr_itm FIELDS SUM( price )
*      WHERE sid = @ls_item-Sid AND pr_id = @ls_item-PrId
*      INTO @DATA(lv_sum2).>


    ENDIF.

  ENDMETHOD.

  METHOD get_instance_authorizations.
  ENDMETHOD.

  METHOD Copy_PR_Item.


*    DATA:
*      lt_prheader TYPE TABLE FOR CREATE yi_mm_pr_hdr,
*      lt_pritem   TYPE TABLE FOR CREATE yi_mm_pr_hdr\_PR_Item.
*
*    "--- Read PR Header
*    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*      ENTITY PR_Header
*      ALL FIELDS
*      WITH CORRESPONDING #( keys )
*      RESULT DATA(lt_header_pr).
*
*    "--- Read PR Items
*    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*      ENTITY PR_Item
*      ALL FIELDS
*      WITH CORRESPONDING #( keys )
*      RESULT DATA(lt_item_pr).
*
*    LOOP AT lt_header_pr ASSIGNING FIELD-SYMBOL(<fs_prheader>).
*
*      "Create Header
*      APPEND VALUE #(
*        %cid  = keys[ 1 ]-%cid
*        %data = CORRESPONDING #( <fs_prheader> EXCEPT sid prid )
*      ) TO lt_prheader ASSIGNING FIELD-SYMBOL(<ls_header>).
*
*      "Prepare Item container
*      APPEND VALUE #(
*        %cid_ref = <ls_header>-%cid
*      ) TO lt_pritem ASSIGNING FIELD-SYMBOL(<ls_item>).
*
*      "Attach Items
*      LOOP AT lt_item_pr ASSIGNING FIELD-SYMBOL(<fs_pritem>) USING KEY entity
*           WHERE Sid  = <fs_prheader>-Sid
*             AND PrId = <fs_prheader>-PrId.
*
*        APPEND VALUE #(
*          %cid  = <ls_header>-%cid && <fs_pritem>-Sid && <fs_pritem>-PrId && <fs_pritem>-ItemNo
*          %data = CORRESPONDING #( <fs_pritem> EXCEPT itemno )
*        ) TO <ls_item>-%target.
*
*      ENDLOOP.
*
*    ENDLOOP.

*    "--- Create Header + Items
*    MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*      ENTITY PR_Header
*      CREATE FIELDS ( PrType Requester Plant TotalValue Currency Location )
*      WITH lt_prheader
*      MAPPED DATA(lt_mapped).
*
*    "--- Correct mapping assignment
*    mapped-pr_header = lt_mapped-pr_header.


*    MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*          ENTITY PR_Header
*          CREATE BY \_PR_Item
*          FIELDS ( Material Qty Uom Price Currency Vendor )
*          WITH lt_pritem
*          MAPPED DATA(lt_mapped1).
*
*    mapped-pr_item = lt_mapped1-pr_item.


*    DATA : lt_prheader TYPE TABLE FOR CREATE yi_mm_pr_hdr,
*           lt_pritem   TYPE TABLE FOR CREATE yi_mm_pr_hdr\_PR_Item..
*
*    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*    ENTITY PR_Header
*    ALL FIELDS WITH CORRESPONDING #( keys )
*    RESULT DATA(lt_header_PR).
*
*    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*    ENTITY PR_Item
*    ALL FIELDS WITH VALUE #( FOR ls IN keys ( %key = ls-%key ) )
*    RESULT DATA(lt_item_PR).
*
*    LOOP AT lt_header_PR ASSIGNING FIELD-SYMBOL(<fs_prheader>).
*
*      APPEND VALUE #( %cid = keys[ KEY entity %key = <fs_prheader>-%key  ]-%cid
*                      %data = CORRESPONDING #( <fs_prheader> EXCEPT sid prid ) )
*             TO lt_prheader ASSIGNING FIELD-SYMBOL(<ls_header>).
*
*             lt_prheader = vaLUE #( base lt_prheader (  ) ).
*
*      APPEND VALUE #( %cid_ref = <ls_header>-%cid )
*             TO lt_pritem ASSIGNING FIELD-SYMBOL(<ls_item>).
*
*      LOOP AT lt_item_PR ASSIGNING FIELD-SYMBOL(<fs_pritem>)
*                   USING KEY entity
*                   WHERE Sid = <ls_header>-Sid AND PrId = <ls_header>-PrId.
*
*        APPEND VALUE #( %cid = <ls_header>-%cid && <fs_pritem>-ItemNo
*                        %data = CORRESPONDING #( <fs_pritem> EXCEPT  itemno  ) )
*                TO <ls_item>-%target .
*
*      ENDLOOP.
*    ENDLOOP.
*
*    MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*      ENTITY PR_Header
*      CREATE FIELDS ( PrType Requester Plant TotalValue Currency Location )
*      WITH lt_prheader
*      MAPPED DATA(lt_mapped).
*    mapped-pr_header = lt_mapped-pr_header.

*    LOOP AT keys INTO DATA(ls_data).
**      SELECT * FROM ymm_pr_itm
**      WHERE sid = @ls_data-Sid AND pr_id = @ls_data-PrId AND item_no = @ls_data-ItemNo
**      INTO TABLE @DATA(lt_data).
*
*      lt_pritem  = VALUE #( ( %is_draft = ls_data-%is_draft
*
*                            %cid_ref = ls_data-%cid
*                           Sid = ls_data-Sid
*                           PrId = ls_data-PrId
*                           %target = CORRESPONDING #( lt_item_PR EXCEPT ItemNo ) ) ).
*
*    ENDLOOP.
*
*    MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*          ENTITY PR_Header
*          CREATE BY \_PR_Item
*          FIELDS ( Material Qty Uom Price Currency Vendor )
*          WITH lt_pritem
*          MAPPED DATA(lt_mapped1).
*
*    mapped-pr_item = lt_mapped1-pr_item.

  ENDMETHOD.

ENDCLASS.




CLASS lhc_PO_Header DEFINITION INHERITING FROM cl_abap_behavior_handler.
  PRIVATE SECTION.

    METHODS update FOR MODIFY
      IMPORTING entities FOR UPDATE PO_Header.

    METHODS delete FOR MODIFY
      IMPORTING keys FOR DELETE PO_Header.

    METHODS read FOR READ
      IMPORTING keys FOR READ PO_Header RESULT result.

    METHODS rba_Pr_header FOR READ
      IMPORTING keys_rba FOR READ PO_Header\_Pr_header FULL result_requested RESULT result LINK
association_links.

    METHODS provide_docdate FOR DETERMINE ON MODIFY
      IMPORTING keys FOR PO_Header~provide_docdate.
    METHODS CCode_validate FOR VALIDATE ON SAVE
      IMPORTING keys FOR PO_Header~CCode_validate.

    METHODS PurhcaseGroup_validate FOR VALIDATE ON SAVE
      IMPORTING keys FOR PO_Header~PurhcaseGroup_validate.

    METHODS PurhcaseORG_validate FOR VALIDATE ON SAVE
      IMPORTING keys FOR PO_Header~PurhcaseORG_validate.
    METHODS rba_Po_item FOR READ
      IMPORTING keys_rba FOR READ PO_Header\_Po_item FULL result_requested RESULT result LINK association_links.

    METHODS cba_Po_item FOR MODIFY
      IMPORTING entities_cba FOR CREATE PO_Header\_Po_item.

    METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
      IMPORTING keys REQUEST requested_authorizations FOR PO_Header RESULT result.

    METHODS Copy_PO FOR MODIFY
      IMPORTING keys FOR ACTION PO_Header~Copy_PO.

ENDCLASS.

CLASS lhc_PO_Header IMPLEMENTATION.

  METHOD update.
    DATA:lt_po_update TYPE STANDARD TABLE OF ymm_po_hdr,
         ls_msg       TYPE bapiret2.
    lt_po_update = CORRESPONDING #( entities ).

    CALL FUNCTION 'YBAPI_PO'
      EXPORTING
        po_update_h = lt_po_update
      IMPORTING
        return      = ls_msg.


  ENDMETHOD.

  METHOD delete.
    DATA:lt_po_delete TYPE STANDARD TABLE OF ymm_po_hdr,
         ls_msg       TYPE bapiret2.
    lt_po_delete  = CORRESPONDING #( keys ).

    CALL FUNCTION 'YBAPI_PO'
      EXPORTING
        po_delete_h = lt_po_delete
      IMPORTING
        return      = ls_msg.

  ENDMETHOD.

  METHOD read.
  ENDMETHOD.
  METHOD rba_Pr_header.
  ENDMETHOD.

  METHOD provide_docdate.
    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PO_Header
    ALL FIELDS WITH CORRESPONDING #( keys )
    RESULT DATA(po_header).
    LOOP AT po_header INTO DATA(ls_header).
      IF ls_header-Docdate IS INITIAL .
        ls_header-Docdate = sy-datum.
        MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
        ENTITY PO_Header
        UPDATE FIELDS ( Docdate )
        WITH VALUE #( ( %tky = ls_header-%tky  %data = CORRESPONDING #( ls_header ) ) ).
      ENDIF.
    ENDLOOP.
  ENDMETHOD.

  METHOD CCode_validate.
  ENDMETHOD.

  METHOD PurhcaseGroup_validate.
    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
    ENTITY PO_Header FIELDS ( Pgroup )
    WITH CORRESPONDING #( keys )
    RESULT DATA(po_header).

    LOOP AT po_header INTO DATA(ls_header).
      APPEND VALUE #(  %tky = ls_header-%tky  %state_area = 'PGROUP'  ) TO reported-po_header.

      IF ls_header-Pgroup IS INITIAL.
        failed-po_header = VALUE #(  ( %tky = ls_header-%tky ) ).
        reported-po_header = VALUE #(  (
                       %tky = ls_header-%tky
                       %msg = new_message_with_text(
                                severity = if_abap_behv_message=>severity-error
                                text     =  'Enter Purchase Group'
                              )
                              %element-pgroup = if_abap_behv=>mk-on
                      %state_area = 'PGROUP'
                      %path-pr_header = CORRESPONDING #( ls_header )
                        ) ).

      ELSE.

        SELECT SINGLE FROM t024 FIELDS ( ekgrp ) WHERE ekgrp = @ls_header-Pgroup INTO @DATA(lv_group).
        IF sy-subrc NE 0.
          failed-po_header = VALUE #( ( %tky = ls_header-%tky ) ).
          reported-po_header = VALUE #(  (
                              %tky = ls_header-%tky
                              %msg = new_message_with_text(
                                       severity = if_abap_behv_message=>severity-error
                                       text     =  |{ ls_header-Pgroup } is in valid|
                                     )
                             %element-pgroup = if_abap_behv=>mk-on
                             %state_area = 'PGROUP'
                             %path-pr_header = CORRESPONDING #( ls_header )
                               ) ).

        ENDIF.
      ENDIF.
    ENDLOOP.
  ENDMETHOD.

  METHOD PurhcaseORG_validate.
    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
      ENTITY PO_Header FIELDS ( Porg )
      WITH CORRESPONDING #( keys )
      RESULT DATA(po_header).

*    APPEND VALUE #(  %tky = po_header[ 1 ]-%tky %state_area = 'PORG'   ) TO reported-po_header.

    LOOP AT po_header INTO DATA(ls_header).
      APPEND VALUE #(  %tky = ls_header-%tky  %state_area = 'PORG'   ) TO reported-po_header.

      IF ls_header-Porg IS INITIAL.
        failed-po_header = VALUE #(  ( %tky = ls_header-%tky ) ).
        reported-po_header = VALUE #(  (
                       %tky = ls_header-%tky
                       %msg = new_message_with_text(
                                severity = if_abap_behv_message=>severity-error
                                text     =  'Enter Purchase Organization'
                              )
                              %element-porg = if_abap_behv=>mk-on
                      %state_area = 'PORG'
                      %path-pr_header = CORRESPONDING #( ls_header )
                        ) ).

      ELSE.
        SELECT SINGLE FROM t024e FIELDS ( ekorg ) WHERE ekorg = @ls_header-Porg INTO @DATA(lv_org).
        IF sy-subrc NE 0.
          failed-po_header = VALUE #( ( %tky = ls_header-%tky ) ).
          reported-po_header = VALUE #(  (
                              %tky = ls_header-%tky
                              %msg = new_message_with_text(
                                      severity = if_abap_behv_message=>severity-error
                                      text     =  |{ ls_header-Porg } is in valid|
                                    )
                             %element-porg = if_abap_behv=>mk-on
                             %state_area = 'PORG'
                             %path-pr_header = CORRESPONDING #( ls_header )
                               ) ).
        ENDIF.
      ENDIF.
    ENDLOOP.


  ENDMETHOD.

  METHOD rba_Po_item.
  ENDMETHOD.

  METHOD cba_Po_item.
    DATA: lv_max_pi1 TYPE ebelp,
          lv_max_pi2 TYPE ebelp,
          lv_next_no TYPE ebelp,
          lt_po_item TYPE TABLE OF ymm_po_itm,
          ls_msg     TYPE bapiret2.

    DATA(lv_ponum) = entities_cba[ 1 ]-Poid.

    SELECT FROM ymm_po_itm FIELDS MAX( itemno ) WHERE poid = @lv_ponum INTO @lv_max_pi1.
    SELECT FROM ypo_itm_d FIELDS MAX( itemno ) WHERE poid = @lv_ponum INTO @lv_max_pi2.

    " Get next number
    lv_next_no = COND i( WHEN lv_max_pi1 > lv_max_pi2 THEN lv_max_pi1 + 10 ELSE lv_max_pi2 + 10 ).


    LOOP AT entities_cba INTO DATA(ls_header).
      lt_po_item = CORRESPONDING #( ls_header-%target MAPPING sid = Sid prid = Prid poid = Poid ).
    ENDLOOP.
    LOOP AT lt_po_item ASSIGNING FIELD-SYMBOL(<fs_data>).
      <fs_data>-sid = entities_cba[ 1 ]-Sid.
      <fs_data>-prid = entities_cba[ 1 ]-Prid.
      <fs_data>-poid = entities_cba[ 1 ]-Poid.
      <fs_data>-itemno = lv_next_no.
      <fs_data>-deliverydate = sy-datum + 10.
      lv_next_no += 10.
    ENDLOOP.

    CALL FUNCTION 'YBAPI_PO'
      EXPORTING
        po_create_i = lt_po_item
      IMPORTING
        return      = ls_msg.

    IF ls_msg IS NOT INITIAL.

      reported-po_item = VALUE #( ( Sid = entities_cba[ 1 ]-Sid
                                      Prid = entities_cba[ 1 ]-PrId
                                      Poid = entities_cba[ 1 ]-Poid
                                      Itemno = ls_msg-message_v1
                                            %msg = new_message_with_text(
                                               severity = COND #( WHEN ls_msg-type = 'S'
                                                                    THEN    if_abap_behv_message=>severity-
success
                                                                    ELSE if_abap_behv_message=>severity-error )
                                               text     = COND #( WHEN ls_msg-type = 'S'
                                                                THEN |PO Items Created|
                                                                ELSE |PO Item Failed Create| )
                                             ) ) ).
    ENDIF.


  ENDMETHOD.

  METHOD get_instance_authorizations.

  ENDMETHOD.

  METHOD Copy_PO.

    DATA : lt_poheader TYPE TABLE FOR CREATE yi_mm_pr_hdr\_PO_Header,
           lt_poitem   TYPE TABLE FOR CREATE yi_po_hdr\_PO_Item.

*    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*      ENTITY PO_Header
*      ALL FIELDS
*      WITH CORRESPONDING #( keys )
*      RESULT DATA(lt_PO).
*
*    IF lt_PO IS NOT INITIAL.
*
*      lt_poheader = CORRESPONDING #( lt_PO EXCEPT PrId ).
*
*      MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*      ENTITY PR_Header
*      CREATE BY \_PO_Header FROM lt_poheader
*      MAPPED DATA(lt_mapped).
```

```
   *    ENDIF.


   *   DATA : it_travel        TYPE TABLE FOR CREATE yi_travel1_m,
   *           it_booking_cba   TYPE TABLE FOR CREATE YI_travel1_M\_booking,
   *           it_booksuppl_cba TYPE TABLE FOR CREATE yi_booking1_m\_bookingsuppl.

       READ TABLE keys ASSIGNING FIELD-SYMBOL(<ls_without_cid>) WITH KEY %cid = ' '.
       ASSERT <ls_without_cid> IS NOT ASSIGNED.

       READ ENTITIES OF yi_mm_pr_hdr  IN LOCAL MODE
           ENTITY PO_Header
           ALL FIELDS WITH CORRESPONDING #( keys )
           RESULT DATA(lt_POH)
           FAILED DATA(lt_failed).

       READ ENTITIES OF yi_mm_pr_hdr  IN LOCAL MODE
           ENTITY PO_Header BY \_PO_Item
           ALL FIELDS WITH CORRESPONDING #( lt_poh )
           RESULT DATA(lt_POI).



   *    LOOP AT lt_POH ASSIGNING FIELD-SYMBOL(<ls_poh>).
   *
   *      APPEND VALUE #( %cid_ref = keys[ KEY entity Prid = <ls_poh>-Prid ]-%cid_ref
   *                      %data = CORRESPONDING #( <ls_travel_r> EXCEPT travelid ) )
   *
   *                   TO lt_poheader ASSIGNING FIELD-SYMBOL(<ls_travel>).
   *
   *      <ls_travel>-BeginDate = cl_abap_context_info=>get_system_date(  ).
   *      <ls_travel>-EndDate = cl_abap_context_info=>get_system_date(  ) + 30.
   *      <ls_travel>-OverallStatus = 'O'.
   *
   *      APPEND VALUE #( %cid_ref = <ls_travel>-%cid )
   *                TO it_booking_cba ASSIGNING FIELD-SYMBOL(<it_booking>).
   *
   *      LOOP AT lt_booking_r ASSIGNING FIELD-SYMBOL(<ls_booking_r>)
   *                                                 USING KEY entity
   *                                                 WHERE TravelId = <ls_travel_r>-TravelId.
   *        APPEND VALUE #( %cid = <ls_travel>-%cid && <ls_booking_r>-BookingId
   *                        %data = CORRESPONDING #( <ls_booking_r> EXCEPT travelid ) )
   *                   TO <it_booking>-%target ASSIGNING FIELD-SYMBOL(<ls_booking_n>).
   *        <ls_booking_n>-BookingStatus = 'N'.
   *
   *        APPEND VALUE #( %cid_ref = <ls_booking_n>-%cid )
   *                TO it_booksuppl_cba ASSIGNING FIELD-SYMBOL(<ls_booksupp>).
   *
   *
   *        LOOP AT it_booksuppl_cba ASSIGNING FIELD-SYMBOL(<ls_booksuppl_r>)
   *                                                         USING KEY entity
   *                                                         WHERE TravelId = <ls_travel_r>-TravelId
   *                                                           AND BookingId = <ls_booking_r>-BookingId.
   *
   *          APPEND VALUE #( %cid = <ls_travel>-%cid && <ls_booksuppl_r>-TravelId && <ls_booksuppl_r>-BookingId )
   *                 TO <ls_booksupp>-%target.
   *
   *        ENDLOOP.
   *      ENDLOOP.
   *    ENDLOOP.
   *
   *    MODIFY ENTITIES OF yi_travel1_m IN LOCAL MODE
   *        ENTITY yi_travel1_m
   *        CREATE FIELDS ( AgencyId CustomerId BeginDate EndDate TotalPrice CurrencyCode OverallStatus
   Description  )
   *        WITH it_travel
   *
   *        ENTITY yi_travel1_m
   *        CREATE BY \_booking
   *        FIELDS ( BookingId BookingDate CustomerId CarrierId ConnectionId FlightDate FlightPrice CurrencyCode
   BookingStatus )
   *        WITH it_booking_cba
   *
   *        ENTITY YI_booking1_m
   *        CREATE BY \_bookingsuppl
   *        FIELDS ( BookingSupplementId SupplementId Price CurrencyCode )
   *        WITH it_booksuppl_cba
   *        MAPPED DATA(it_mapped).
   *
   *    mapped-yi_travel1_m = it_mapped-yi_travel1_m.



     ENDMETHOD.

   ENDCLASS.




CLASS lhc_PO_Item DEFINITION INHERITING FROM cl_abap_behavior_handler.
  PRIVATE SECTION.
```

```
    METHODS update FOR MODIFY
      IMPORTING entities FOR UPDATE PO_Item.

    METHODS delete FOR MODIFY
      IMPORTING keys FOR DELETE PO_Item.

    METHODS read FOR READ
      IMPORTING keys FOR READ PO_Item RESULT result.

    METHODS rba_Po_header FOR READ
      IMPORTING keys_rba FOR READ PO_Item\_Po_header FULL result_requested RESULT result LINK
association_links.

    METHODS rba_Pr_header FOR READ
      IMPORTING keys_rba FOR READ PO_Item\_Pr_header FULL result_requested RESULT result LINK
association_links.
    METHODS material_vaidate FOR VALIDATE ON SAVE
      IMPORTING keys FOR PO_Item~material_vaidate.
    METHODS AutoFillData FOR DETERMINE ON MODIFY
      IMPORTING keys FOR PO_Item~AutoFillData.
    METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
      IMPORTING keys REQUEST requested_authorizations FOR PO_Item RESULT result.

    METHODS Copy_POItem FOR MODIFY
      IMPORTING keys FOR ACTION PO_Item~Copy_POItem.



ENDCLASS.

CLASS lhc_PO_Item IMPLEMENTATION.

  METHOD update.
    DATA : lt_po_update TYPE TABLE OF ymm_po_itm,
           ls_msg       TYPE bapiret2.
    lt_po_update = CORRESPONDING #( entities ).
    IF lt_po_update IS NOT INITIAL.
      CALL FUNCTION 'YBAPI_PO'
        EXPORTING
          po_update_i = lt_po_update
        IMPORTING
          return      = ls_msg.

      IF ls_msg IS NOT INITIAL.

        reported-po_item = VALUE #( ( Sid = entities[ 1 ]-Sid
                                        Prid = entities[ 1 ]-PrId
                                        Poid = entities[ 1 ]-Poid
                                        Itemno = ls_msg-message_v1
                                              %msg = new_message_with_text(
                                                 severity = COND #( WHEN ls_msg-type = 'S'
                                                                      THEN   if_abap_behv_message=>severity-
success
                                                                      ELSE if_abap_behv_message=>severity-
error )
                                                 text     = COND #( WHEN ls_msg-type = 'S'
                                                                  THEN |PO Items Updated|
                                                                  ELSE |PO Item Failed To Update| )
                                               ) ) ).
      ENDIF.
    ENDIF.
  ENDMETHOD.

  METHOD delete.

    DATA : lt_po_delete TYPE TABLE OF ymm_po_itm,
           ls_msg       TYPE bapiret2.
    lt_po_delete = CORRESPONDING #( keys ).
    IF lt_po_delete IS NOT INITIAL.
      CALL FUNCTION 'YBAPI_PO'
        EXPORTING
          po_delete_i = lt_po_delete
        IMPORTING
          return      = ls_msg.

      IF ls_msg IS NOT INITIAL.

        reported-po_item = VALUE #( ( Sid = lt_po_delete[ 1 ]-Sid
                                        Prid = lt_po_delete[ 1 ]-PrId
                                        Poid = lt_po_delete[ 1 ]-Poid
                                        Itemno = ls_msg-message_v1
                                              %msg = new_message_with_text(
                                                 severity = COND #( WHEN ls_msg-type = 'S'
                                                                      THEN   if_abap_behv_message=>severity-
success
                                                                      ELSE if_abap_behv_message=>severity-
error )
                                                 text     = COND #( WHEN ls_msg-type = 'S'
                                                                  THEN |PO Items Deleted|
                                                                  ELSE |PO Item Failed To Delete| )
                                               ) ) ).
      ENDIF.
    ENDIF.

  ENDMETHOD.
```

```
    METHOD read.
    ENDMETHOD.

    METHOD rba_Po_header.
    ENDMETHOD.

    METHOD rba_Pr_header.
    ENDMETHOD.

    METHOD material_vaidate.

      READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
      ENTITY PO_Item
      FIELDS ( Material )
      WITH CORRESPONDING #( keys )
      RESULT DATA(lt_po_item).
      LOOP AT lt_po_item INTO DATA(ls_data).
        IF ls_data-Material IS INITIAL.
          failed-po_item = VALUE #( ( %key = ls_data-%key ) ).
          reported-po_item = VALUE #( ( %tky = ls_data-%tky
                                        %msg = new_message_with_text(
                                                 severity = if_abap_behv_message=>severity-error
                                                 text     = 'Material Should not Be Initial'
                                               )
                                       %element-material = if_abap_behv=>mk-on
                                       %state_area = 'material'
                                       %path-po_header = CORRESPONDING #( ls_data ) ) ).
        ENDIF.
      ENDLOOP.

    ENDMETHOD.

    METHOD AutoFillData.


  *    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
  *      ENTITY PO_Item
  *      ALL FIELDS
  *      WITH CORRESPONDING #( keys )
  *      RESULT DATA(lt_item_pr).
  *
  *    LOOP AT lt_item_pr ASSIGNING FIELD-SYMBOL(<fs_data>).
  *
  **        SELECT SINGLE FROM ymm_pr_itm FIELDS qty,uom,price
  **        WHERE sid = @<fs_data>-Sid AND pr_id = @<fs_data>-Prrid AND item_no = @<fs_data>-Pritemno
  **        INTO @DATA(ls_item).
  *
  *
  *      <fs_data>-Qty = 20.
  *      MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
  *        ENTITY PO_Item
  *        UPDATE FIELDS ( Qty  )
  *        WITH VALUE #( FOR ls_data IN lt_item_pr ( %tky = ls_data-%tky
  *                                                  %data = CORRESPONDING #(  <fs_data> )
  *
  *
  *                                                    ) )
  *        MAPPED DATA(lt_mapped).
  *
  *    ENDLOOP.


    ENDMETHOD.



    METHOD get_instance_authorizations.

    ENDMETHOD.

    METHOD Copy_POItem.


    ENDMETHOD.

  ENDCLASS.
```

```
CLASS ycl_helper_pr_po DEFINITION  PUBLIC  FINAL  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES : if_oo_adt_classrun,
      if_sadl_exit_calc_element_read.

    TYPES : tt_entities_pr    TYPE TABLE FOR CREATE yi_mm_pr_hdr\\pr_header,
            tt_mapped_early   TYPE RESPONSE FOR MAPPED EARLY yi_mm_pr_hdr,
            tt_failed_early   TYPE RESPONSE FOR FAILED EARLY yi_mm_pr_hdr,
            tt_reported_early TYPE RESPONSE FOR REPORTED EARLY yi_mm_pr_hdr,
            tt_reported_late  TYPE RESPONSE FOR REPORTED LATE yi_mm_pr_hdr,
            tt_pr_hdr_delete  TYPE TABLE FOR DELETE yi_mm_pr_hdr\\pr_header,
            tt_pr_hdr_update  TYPE TABLE FOR UPDATE yi_mm_pr_hdr\\pr_header,
            tt_pr_hdr_read    TYPE TABLE FOR READ IMPORT yi_mm_pr_hdr\\pr_header,
            tt_pr_hdr_result  TYPE TABLE FOR READ RESULT yi_mm_pr_hdr\\pr_header,
            entities_cba      TYPE TABLE FOR CREATE yi_mm_pr_hdr\\pr_header\_pr_item,
            tt_mapped_late    TYPE RESPONSE FOR MAPPED LATE yi_mm_pr_hdr,
            tt_pr_itm_en      TYPE TABLE FOR UPDATE yi_mm_pr_hdr\\pr_item,
            tt_pr_itm_delete  TYPE TABLE FOR DELETE yi_mm_pr_hdr\\pr_item,
            tt_entities_po    TYPE TABLE FOR CREATE yi_mm_pr_hdr\\pr_header\_po_header.

    DATA :gt_pr_header_create TYPE TABLE OF ymm_pr_hdr,
          gt_pr_header_delete TYPE TABLE OF ymm_pr_hdr,
          gt_pr_header_update TYPE TABLE OF ymm_pr_hdr,
          gt_pr_item_create   TYPE TABLE OF ymm_pr_itm,
          gt_pr_item_delete   TYPE TABLE OF ymm_pr_itm,
          gt_pr_item_update   TYPE TABLE OF ymm_pr_itm.

    CLASS-METHODS : get_instance RETURNING VALUE(r_obj) TYPE REF TO ycl_helper_pr_po,
      get_next_id         RETURNING VALUE(rv_id) TYPE sysuuid_x16.

    METHODS : earlynumbering_create
      IMPORTING entities TYPE tt_entities_pr
      CHANGING  mapped   TYPE  tt_mapped_early
                failed   TYPE tt_failed_early
                reported TYPE tt_reported_early,
      create_PR_HDR
        IMPORTING entities TYPE tt_entities_pr
        CHANGING  mapped   TYPE tt_mapped_early
                  failed   TYPE tt_failed_early
                  reported TYPE tt_reported_early,
      delete_PR_HDR
        IMPORTING keys     TYPE  tt_pr_hdr_delete
        CHANGING  mapped   TYPE tt_mapped_early
                  failed   TYPE tt_failed_early
                  reported TYPE tt_reported_early,
      update_PR_HDR
        IMPORTING entities TYPE  tt_pr_hdr_update
        CHANGING  mapped   TYPE tt_mapped_early
                  failed   TYPE tt_failed_early
                  reported TYPE tt_reported_early,
      read_PR_HDR
        IMPORTING keys     TYPE   tt_pr_hdr_read
        CHANGING  result   TYPE tt_pr_hdr_result
                  failed   TYPE tt_failed_early
                  reported TYPE tt_reported_early,
      save_data
        CHANGING reported TYPE  tt_reported_late,

      cba_pr_item
        IMPORTING entities_cba TYPE  entities_cba
        CHANGING  mapped       TYPE tt_mapped_early
                  failed       TYPE tt_failed_early
                  reported     TYPE tt_reported_early,
      adjust_numbers
        CHANGING mapped   TYPE  tt_mapped_late
                 reported TYPE   tt_reported_late ,
      update_pr_item
        IMPORTING entities TYPE tt_pr_itm_en
        CHANGING  mapped   TYPE tt_mapped_early
                  failed   TYPE tt_failed_early
                  reported TYPE tt_reported_early,
      delete_pr_itm
        IMPORTING keys     TYPE  tt_pr_itm_delete
        CHANGING  mapped   TYPE tt_mapped_early
                  failed   TYPE tt_failed_early
                  reported TYPE tt_reported_early,
      earlynumbering_cba_po_header
        IMPORTING entities TYPE  tt_entities_po
        CHANGING  mapped   TYPE tt_mapped_early
                  failed   TYPE tt_failed_early
                  reported TYPE tt_reported_early.

  PROTECTED SECTION.
  PRIVATE SECTION.
    CLASS-DATA : g_obj TYPE REF TO ycl_helper_pr_po.
ENDCLASS.

CLASS ycl_helper_pr_po IMPLEMENTATION.
  METHOD IF_oo_adt_classrun~main.
*    DELETE FROM ypr_hdr_d.
*    DELETE FROM ymm_pr_hdr.
*    DELETE FROM ymm_pr_itm.
  ENDMETHOD.
  METHOD get_instance.
    r_obj = g_obj = COND #( WHEN g_obj IS BOUND THEN g_obj
                                ELSE NEW #(  ) ).
  ENDMETHOD.
  METHOD get_next_id.
    TRY.
        rv_id = cl_uuid_factory=>create_system_uuid(  )->create_uuid_x16(  ).
      CATCH cx_uuid_error.
    ENDTRY.
  ENDMETHOD.
  METHOD earlynumbering_create.
    DATA: lv_max_pr1 TYPE string,
          lv_max_pr2 TYPE string,
          lv_num1    TYPE i,
          lv_num2    TYPE i,
          lv_next_no TYPE i,
          lv_pr_id   TYPE banfn,
          lv_numc(6) TYPE n.
    DATA(s_id) =  get_next_id(  ).

    SELECT SINGLE MAX( pr_id ) FROM ymm_pr_hdr INTO @lv_max_pr1.
    SELECT SINGLE MAX( prid ) FROM ypr_hdr_d  INTO @lv_max_pr2.

    lv_num1 = COND i(  WHEN lv_max_pr1 IS INITIAL THEN 0 ELSE lv_max_pr1+2(6) ).
    lv_num2 = COND i( WHEN lv_max_pr2 IS INITIAL THEN 0 ELSE lv_max_pr2+2(6) ).

    " Get next number
    lv_next_no = COND i( WHEN lv_num1 > lv_num2 THEN lv_num1 + 1 ELSE lv_num2 + 1 ).

    lv_numc = lv_next_no.
    lv_pr_id = |RE{ lv_numc }|.

    mapped-pr_header = VALUE #( FOR ls_entity IN entities ( %cid = ls_entity-%cid
                                                            %is_draft = ls_entity-%is_draft
                                                            Sid = s_id
                                                            PrId = lv_pr_id ) ).

  ENDMETHOD.

  METHOD create_pr_hdr.
    DATA : ls_msg TYPE bapiret2.
    gt_pr_header_create = CORRESPONDING #( entities MAPPING sid = Sid pr_id = PrId
                                           pr_type = PrType total_value = TotalValue
                                           created_by = CreatedBy created_on = CreatedOn
                                           locallastchangd_at = LocallastchangdAt
                                           locallastchangd_by = LocallastchangdBy  ).
    gt_pr_header_create[ 1 ]-created_on = sy-datum.
    gt_pr_header_create[ 1 ]-status = 'S'.
    gt_pr_header_create[ 1 ]-sendmail = 'NO'.

  ENDMETHOD.

  METHOD save_data.
*    IF gt_pr_header_create IS NOT INITIAL.
*      INSERT ymm_pr_hdr FROM TABLE gt_pr_header_create.
*    ENDIF.
*    IF gt_pr_header_delete IS NOT INITIAL.
*      DELETE ymm_pr_hdr FROM TABLE gt_pr_header_delete.
*    ENDIF.
*    IF  gt_pr_header_update IS NOT INITIAL.
*      UPDATE ymm_pr_hdr FROM TABLE  gt_pr_header_update.
*    ENDIF.

*    IF gt_pr_item_create IS NOT INITIAL.
*      INSERT ymm_pr_itm FROM TABLE gt_pr_item_create.
*    ENDIF.
*    IF gt_pr_item_update IS NOT INITIAL.
*      UPDATE ymm_pr_itm FROM TABLE gt_pr_item_update.
*    ENDIF.
*    IF gt_pr_item_delete IS NOT INITIAL.
*      DELETE ymm_pr_itm FROM TABLE gt_pr_item_delete.
*    ENDIF.

  ENDMETHOD.

  METHOD delete_pr_hdr.
    gt_pr_header_delete = CORRESPONDING #( keys MAPPING sid = Sid pr_id = PrId ).
  ENDMETHOD.

  METHOD update_pr_hdr.
    gt_pr_header_update = CORRESPONDING #( entities MAPPING sid = Sid pr_id = PrId
    pr_type = PrType total_value = TotalValue created_by = CreatedBy created_on = CreatedOn
    locallastchangd_at = LocallastchangdAt
    locallastchangd_by = LocallastchangdBy ).
  ENDMETHOD.

  METHOD read_pr_hdr.
    DATA(lv_sid) = keys[ 1 ]-Sid.
    DATA(lv_pid) = keys[ 1 ]-PrId.

    SELECT * FROM ymm_pr_hdr
    WHERE sid = @lv_sid AND pr_id = @lv_pid
    INTO TABLE @DATA(lt_read).

    LOOP AT keys INTO DATA(ls_key).
      result = VALUE #( FOR ls_data IN lt_read
      ( %is_draft = ls_key-%is_draft
      %data = CORRESPONDING #( ls_data MAPPING Sid = sid PrId = pr_id
                                PrType = pr_type TotalValue = total_value
                                CreatedBy = created_by CreatedOn = created_on
                                LocallastchangdAt = locallastchangd_at
                                LocallastchangdBy = locallastchangd_by ) ) ).
    ENDLOOP.

  ENDMETHOD.
  METHOD cba_pr_item.
    LOOP AT entities_cba INTO DATA(ls_entity).
      gt_pr_item_create = CORRESPONDING #( ls_entity-%target MAPPING
                                          sid = Sid
                                          pr_id = PrId  item_no = ItemNo   delivery_date = DeliveryDate
                                           item_value = ItemValue  ).
      mapped-pr_item = VALUE #( ( %is_draft = ls_entity-%is_draft Sid = ls_entity-Sid PrId = ls_entity-
PrId ) ).
    ENDLOOP.

  ENDMETHOD.

  METHOD adjust_numbers.
    IF gt_pr_item_create IS NOT INITIAL.
      DATA : lv_itemnum TYPE bnfpo.
      DATA(lv_prid) = gt_pr_item_create[ 1 ]-pr_id.
      DATA(lv_sid) = get_next_id(  ).
      SELECT FROM ymm_pr_itm FIELDS MAX( item_no ) WHERE pr_id = @lv_prid INTO @DATA(lv_max1).
      SELECT FROM ypr_itm_d FIELDS MAX( itemno ) WHERE prid = @lv_prid INTO @DATA(lv_max2).
      lv_itemnum  = COND #( WHEN lv_max1 = 0 AND lv_max2 = 0 THEN 10
                            WHEN lv_max1 < lv_max2 THEN lv_max2 + 10
                            WHEN lv_max1 > lv_max2 THEN lv_max1 + 10
                            ELSE lv_max1 + 10
                            ).

*      lv_itemnum  = COND #( WHEN lv_max1 = 0  THEN 10  ELSE lv_max1 + 10 ).
      LOOP AT gt_pr_item_create ASSIGNING FIELD-SYMBOL(<fs_item>).
        <fs_item>-item_no = lv_itemnum.
        <fs_item>-delivery_date = sy-datum + 10.
        lv_itemnum += 10.
      ENDLOOP.
    ENDIF.

  ENDMETHOD.

  METHOD if_sadl_exit_calc_element_read~calculate.

    DATA : lt_data TYPE TABLE OF yp_mm_pr_itm.
    LT_data = CORRESPONDING #( it_original_data ).
    LOOP AT LT_data ASSIGNING FIELD-SYMBOL(<fs_data>).
      <FS_data>-total = <fs_data>-Qty * <FS_data>-Price.
    ENDLOOP.
    ct_calculated_data = CORRESPONDING #( lt_data ).

  ENDMETHOD.

  METHOD if_sadl_exit_calc_element_read~get_calculation_info.

  ENDMETHOD.

  METHOD delete_pr_itm.
    gt_pr_item_delete = CORRESPONDING #( keys MAPPING sid = Sid pr_id = PrId item_no = ItemNo ).
  ENDMETHOD.

  METHOD update_pr_item.
    gt_pr_item_update = CORRESPONDING #( entities MAPPING
    pr_id = PrId item_no = ItemNo item_value = ItemValue delivery_date = DeliveryDate ).
  ENDMETHOD.


  METHOD earlynumbering_cba_po_header.


*    DATA: lv_max_pr1 TYPE string,
*          lv_max_pr2 TYPE string,
*          lv_num1    TYPE i,
*          lv_num2    TYPE i,
*          lv_next_no TYPE i,
*          lv_po_id   TYPE ebeln,
*          lv_numc(6) TYPE n.
**    DATA(s_id) =  get_next_id(  ).
*
*    SELECT FROM ymm_po_hdr FIELDS MAX( poid ) INTO @lv_max_pr1.
*    SELECT FROM ypo_hdr_d FIELDS MAX( prid ) INTO @lv_max_pr2.
*
*    lv_num1 = COND i(  WHEN lv_max_pr1 IS INITIAL THEN 0 ELSE lv_max_pr1+2(6) ).
*    lv_num2 = COND i( WHEN lv_max_pr2 IS INITIAL THEN 0 ELSE lv_max_pr2+2(6) ).
*
*    " Get next number
*    lv_next_no = COND i( WHEN lv_num1 > lv_num2 THEN lv_num1 + 1 ELSE lv_num2 + 1 ).
*
*    lv_numc = lv_next_no.
*    lv_po_id = |PO{ lv_numc }|.
*
*    mapped-po_header = VALUE #(
*    FOR ls_entity IN entities ( %tky = ls_entity-%tky
*    Prid = ls_entity-PrId
*                               Poid =  lv_po_id
*                                 ) ).

  ENDMETHOD.

ENDCLASS.
```

```
CLASS lsc_YI_MM_PR_HDR DEFINITION INHERITING FROM cl_abap_behavior_saver.
  PROTECTED SECTION.

    METHODS finalize REDEFINITION.

    METHODS check_before_save REDEFINITION.

    METHODS adjust_numbers REDEFINITION.

    METHODS save REDEFINITION.

    METHODS cleanup REDEFINITION.

    METHODS cleanup_finalize REDEFINITION.

ENDCLASS.

CLASS lsc_YI_MM_PR_HDR IMPLEMENTATION.

  METHOD finalize.
  ENDMETHOD.

  METHOD check_before_save.
  ENDMETHOD.

  METHOD save.
    DATA : ls_msg TYPE bapiret2.

*    ycl_helper_pr_po=>get_instance(  )->save_data(
*      CHANGING
*        reported = reported
*    ).


* Create
    DATA(lt_create) =  ycl_helper_pr_po=>get_instance(  )->gt_pr_header_create.
    IF   lt_create IS NOT INITIAL.
      CALL FUNCTION 'YBAPI_PR_CREATE'
        EXPORTING
          pr_create = lt_create
        IMPORTING
          return    = ls_msg
        EXCEPTIONS
          failed    = 1
          OTHERS    = 2.
      IF ls_msg-type = 'S' .
        reported-pr_header = VALUE #( ( Sid = ls_msg-message_v2 PrId = ls_msg-message_v1
                                         %msg = new_message_with_text(
                                                  severity = COND #( WHEN ls_msg-type = 'S'
                                                                        THEN if_abap_behv_message=>
severity-success
                                                                        ELSE if_abap_behv_message=>
severity-error
                                                                        )
                                         text     = COND #( WHEN ls_msg-type = 'S' THEN |PR { ls_msg-
message_v1 } Created Successfully|
                                                            ELSE |PR { ls_msg-message_v1 } Failed To
Create |
                                                           )  ) ) ).
      ENDIF.
    ENDIF.


* Update
    DATA(lt_update) =  ycl_helper_pr_po=>get_instance(  )->gt_pr_header_update.
    IF   lt_update IS NOT INITIAL.
      CALL FUNCTION 'YBAPI_PR_CREATE'
        EXPORTING
          pr_update = lt_update
        IMPORTING
          return    = ls_msg
        EXCEPTIONS
          failed    = 1
          OTHERS    = 2.
      IF ls_msg IS NOT INITIAL .
        reported-pr_header = VALUE #( ( Sid = ls_msg-message_v2 PrId = ls_msg-message_v1
                                         %msg = new_message_with_text(
                                                  severity = COND #( WHEN ls_msg-type = 'S'
                                                                        THEN if_abap_behv_message=>
severity-success
                                                                        ELSE if_abap_behv_message=>
severity-error
                                                                        )
                                         text     = COND #( WHEN ls_msg-type = 'S' THEN |PR { ls_msg-
message_v1 } Updated|
                                                            ELSE |PR { ls_msg-message_v1 } Failed To
Update|
                                                           )  ) ) ).

      ENDIF.
    ENDIF.
```

```
*  Delete
    DATA(lt_delete) =  ycl_helper_pr_po=>get_instance(  )->gt_pr_header_delete.
    IF   lt_delete IS NOT INITIAL.
      CALL FUNCTION 'YBAPI_PR_CREATE'
        EXPORTING
          pr_delete = lt_delete
        IMPORTING
          return    = ls_msg
        EXCEPTIONS
          failed    = 1
          OTHERS    = 2.
      IF ls_msg IS NOT INITIAL .
        reported-pr_header = VALUE #( ( Sid = ls_msg-message_v2 PrId = ls_msg-message_v1
                                         %msg = new_message_with_text(
                                                  severity = COND #( WHEN ls_msg-type = 'S'
                                                                      THEN if_abap_behv_message=>
severity-information
                                                                      ELSE if_abap_behv_message=>
severity-error
                                                                      )
                                         text     = COND #( WHEN ls_msg-type = 'S' THEN |PR
{ lines( lt_delete ) } Records Deleted Successfully|
                                                            ELSE |PR { ls_msg-message_v1 } Failed To
Delete|
                                                           )  ) ) ).

      ENDIF.
    ENDIF.

* Item Create
    DATA(lt_itm_create) = ycl_helper_pr_po=>get_instance(  )->gt_pr_item_create.
    IF lt_itm_create IS NOT INITIAL.
      CALL FUNCTION 'YBAPI_PR_ITM'
        EXPORTING
          pr_itm_create = lt_itm_create
*         pr_itm_update =
*         pr_itm_delete =
        IMPORTING
          return        = ls_msg.
      IF ls_msg IS NOT INITIAL .
        reported-pr_item = VALUE #( ( Sid = ls_msg-message_v1 PrId = ls_msg-message_v2 ItemNo =
ls_msg-message_v3
                                     %msg = new_message_with_text(
                                              severity = COND #( WHEN ls_msg-type = 'S'
                                                                  THEN if_abap_behv_message=>
severity-success
                                                                  ELSE if_abap_behv_message=>
severity-error
                                                                  )
                                     text     = COND #( WHEN ls_msg-type = 'S' THEN |Item Created
Successfully|
                                                        ELSE |PR { ls_msg-message_v1 } Failed To
Create|
                                                       )  ) ) ).

      ENDIF.
    ENDIF.


* Item update
    DATA(lt_itm_update) = ycl_helper_pr_po=>get_instance(  )->gt_pr_item_update.
    IF lt_itm_update IS NOT INITIAL.
      CALL FUNCTION 'YBAPI_PR_ITM'
        EXPORTING
          pr_itm_update = lt_itm_update
*         pr_itm_delete =
        IMPORTING
          return        = ls_msg.
      IF ls_msg IS NOT INITIAL .
        reported-pr_item = VALUE #( ( Sid = ls_msg-message_v1 PrId = ls_msg-message_v2 ItemNo =
ls_msg-message_v3
                                     %msg = new_message_with_text(
                                              severity = COND #( WHEN ls_msg-type = 'S'
                                                                  THEN if_abap_behv_message=>
severity-success
                                                                  ELSE if_abap_behv_message=>
severity-error
                                                                  )
                                     text     = COND #( WHEN ls_msg-type = 'S' THEN |Updated
Successfully|
                                                        ELSE | Failed To Updated|
                                                       )  ) ) ).

      ENDIF.
    ENDIF.


*  Item Delete
    DATA(lt_itm_delete) = ycl_helper_pr_po=>get_instance(  )->gt_pr_item_delete.
    IF lt_itm_delete IS NOT INITIAL.
      CALL FUNCTION 'YBAPI_PR_ITM'
        EXPORTING
          pr_itm_delete = lt_itm_delete
        IMPORTING
```

```
          return        = ls_msg.
      IF ls_msg IS NOT INITIAL .
        reported-pr_item = VALUE #( ( Sid = ls_msg-message_v1 PrId = ls_msg-message_v2 ItemNo =
ls_msg-message_v3
                                     %msg = new_message_with_text(
                                              severity = COND #( WHEN ls_msg-type = 'S'
                                                                    THEN if_abap_behv_message=>
severity-information
                                                                    ELSE if_abap_behv_message=>
severity-error
                                                                    )
                                     text     = COND #( WHEN ls_msg-type = 'S' THEN
|{ lines( lt_itm_delete ) } Records Deleted Successfully|
                                                        ELSE | Failed To Delete|
                                                       )  ) ) ).

      ENDIF.
    ENDIF.

  ENDMETHOD.

  METHOD cleanup.
  ENDMETHOD.

  METHOD cleanup_finalize.
  ENDMETHOD.

  METHOD adjust_numbers.
    ycl_helper_pr_po=>get_instance(  )->adjust_numbers(
      CHANGING
        mapped   = mapped
        reported = reported
    ).
  ENDMETHOD.

ENDCLASS.





CLASS ycl_helper_mail_send DEFINITION  PUBLIC  FINAL  CREATE PUBLIC .
  PUBLIC SECTION.
    TYPES :
      tt_keys_hdr       TYPE TABLE FOR ACTION IMPORT yi_mm_pr_hdr\\pr_header~send_popup,
      tt_hdr_result     TYPE TABLE FOR ACTION RESULT yi_mm_pr_hdr\\pr_header~send_popup,
      tt_mapped_early   TYPE RESPONSE FOR MAPPED EARLY yi_mm_pr_hdr,
      tt_reported_early TYPE RESPONSE FOR REPORTED EARLY yi_mm_pr_hdr,
      tt_failed_early   TYPE  RESPONSE FOR FAILED EARLY yi_mm_pr_hdr.

    CLASS-METHODS : get_instance RETURNING VALUE(r_obj) TYPE REF TO ycl_helper_mail_send.
    METHODS :       send_mail
      IMPORTING keys     TYPE tt_keys_hdr
      EXPORTING result   TYPE tt_hdr_result
                mapped   TYPE tt_mapped_early
                failed   TYPE tt_failed_early
                reported TYPE tt_reported_early .
  PROTECTED SECTION.
  PRIVATE SECTION.
    CLASS-DATA : g_obj TYPE REF TO ycl_helper_mail_send.
ENDCLASS.



CLASS ycl_helper_mail_send IMPLEMENTATION.
  METHOD get_instance.
    r_obj = g_obj = COND #( WHEN g_obj IS BOUND THEN g_obj
                            ELSE NEW #(  ) ).
  ENDMETHOD.

  METHOD send_mail.
*    READ ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*    ENTITY PR_Header
*    ALL FIELDS WITH CORRESPONDING #( keys )
*    RESULT DATA(lt_header).

    DATA : lt_text1  TYPE soli_tab.

    LOOP AT keys INTO DATA(ls_key).
      DATA(lv_mail) = ls_key-%param-email.
      DATA(lv_name) = ls_key-%param-Name.
      SELECT SINGLE * FROM ymm_pr_hdr
      WHERE sid = @ls_key-Sid AND pr_id = @ls_key-PrId
      INTO @DATA(ls_header).
      IF sy-subrc EQ 0.


* 1) Prepare a text format for  ALV Data.

* 2) convert text format to binary format

        " Build HTML body with table

        CLEAR lt_text1.
        APPEND '<html><body>' TO lt_text1.
        APPEND '<p>Dear Sir,</p>' TO lt_text1.
        APPEND '<p>Created Purchase Requsition details as Below :</p>' TO lt_text1.

        APPEND '<table border="1" cellspacing="0" cellpadding="5">' TO lt_text1.
        APPEND '<tr> <th>Purchase Requsition</th><th>Order Type</th><th>Requestor</th><th>Plant</th>
<th>Amount</th><th>Currency</th><th>User Name</th></tr>' TO lt_text1.

        APPEND |<tr><td>{ ls_header-pr_id }</td><td>{ ls_header-pr_type }</td><td>{ ls_header-
requester }</td><td>{ ls_header-plant }</td><td>{ ls_header-total_value  }</td><td>{ ls_header-
currency }</td><td>{ ls_header-created_by }</td></tr> |
        TO  lt_text1.

        APPEND '</table>' TO lt_text1.
        APPEND '<br/><p>Please Approve as soon as possible,<br/></p>' TO lt_text1.
        APPEND '<br/><p>Thanks & Regards,<br/>Vinod </p>' TO lt_text1.
        APPEND '</body></html>' TO lt_text1.


* 3) create a send request
        TRY.
            CALL METHOD cl_bcs=>create_persistent
              RECEIVING
                result = DATA(lo_bcs).
          CATCH cx_send_req_bcs.
        ENDTRY.

*  4) create a sap user / external user (recipient).

*  5) ADD THE RECIPIENT.
*        for Exter Mail connect
        TRY.
            CALL METHOD cl_cam_address_bcs=>create_internet_address
              EXPORTING
                i_address_string = CONV adr6-smtp_addr( lv_mail )
                i_address_name   = CONV adr6-smtp_addr( lv_name )
                i_incl_sapuser   = 'X'
              RECEIVING
                result           = DATA(lo_external_user).
          CATCH cx_address_bcs.
        ENDTRY.

        TRY.
            CALL METHOD lo_bcs->add_recipient
              EXPORTING
                i_recipient = lo_external_user.
          CATCH cx_send_req_bcs.
        ENDTRY.

* 6) create the document
        TRY.
            CALL METHOD cl_document_bcs=>create_document
              EXPORTING
                i_type    = 'HTM'
                i_subject = |Created PR { ls_header-pr_id }  Details|
                i_text    = lt_text1
              RECEIVING
                result    = DATA(lo_document).
          CATCH cx_document_bcs.
        ENDTRY.

* 7) Add the attachment.

* 8) set the document.
        TRY.
            CALL METHOD lo_bcs->set_document
              EXPORTING
                i_document = lo_document.
          CATCH cx_send_req_bcs.
        ENDTRY.

* 9) Activate or Deactivate immediate sending
        TRY.
            CALL METHOD lo_bcs->set_send_immediately
              EXPORTING
                i_send_immediately = 'X'.
          CATCH cx_send_req_bcs.
        ENDTRY.

* 10) send
        TRY.
            CALL METHOD lo_bcs->send
*  EXPORTING
*    i_with_error_screen = SPACE
              RECEIVING
                result = DATA(lv_result).
          CATCH cx_send_req_bcs.
        ENDTRY.

* 11) Commit only for external email not for sap users.( within sap do not require commit work ).
        IF lv_result IS NOT INITIAL.
*          COMMIT WORK.
          UPDATE ymm_pr_hdr SET sendmail = 'YES'
            WHERE sid = ls_header-sid AND pr_id = ls_header-pr_id.
*          ls_header-status = 'YES'.
*          MODIFY ENTITIES OF yi_mm_pr_hdr IN LOCAL MODE
*          ENTITY PR_Header
*          UPDATE FIELDS ( Status )
*          WITH VALUE #( (  Sid = ls_header-sid PrId = ls_header-pr_id
*          %data = CORRESPONDING #( ls_header ) ) ).

     IF sy-subrc EQ 0.
       result = VALUE #( ( %tky = ls_key-%tky
                              %param-sendmail = 'YES' ) ).
       mapped-pr_header = VALUE #( ( %tky = ls_key-%tky ) ).
       result = VALUE #( ( %tky = ls_key-%tky %param-sendmail = 'YES' ) ).
       reported-pr_header = VALUE #( ( %tky = ls_key-%tky ) ).
     ENDIF.
    ENDIF.

   ENDIF.
  ENDLOOP.

 ENDMETHOD.

ENDCLASS.
```

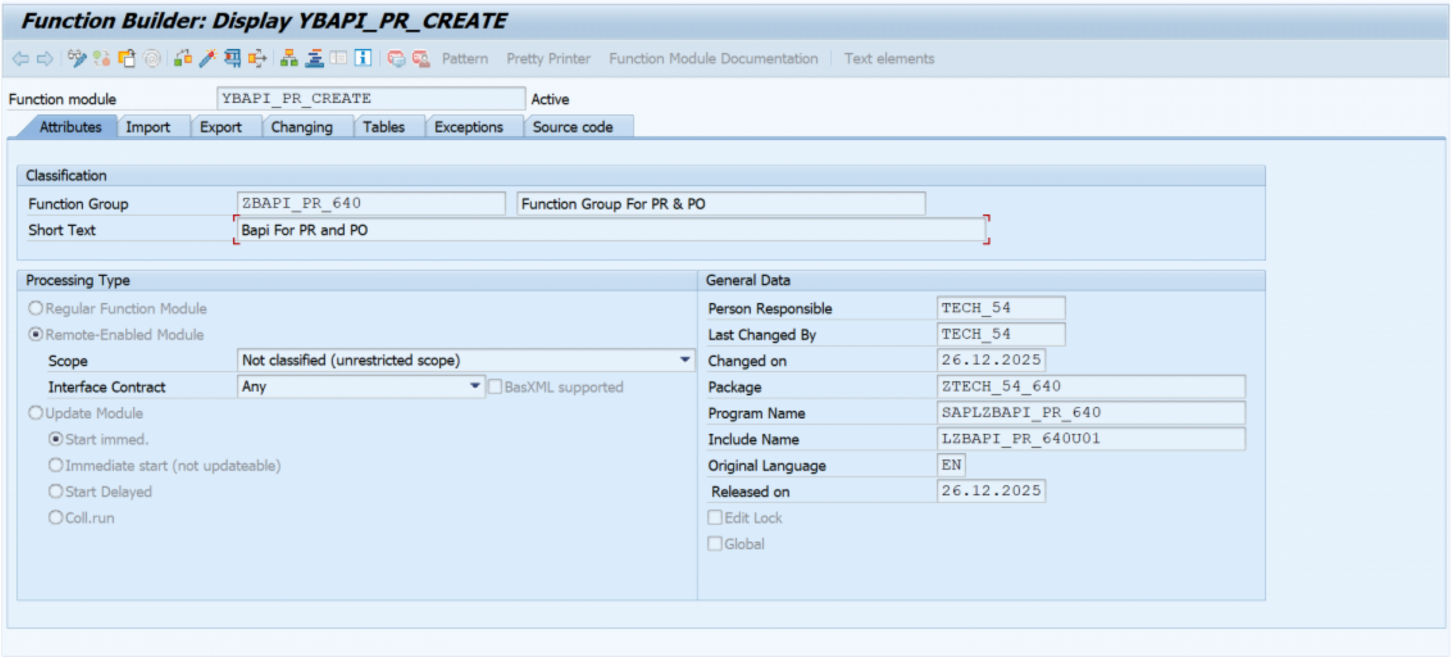
```
@EndUserText.label: 'PR Value Help'
@ObjectModel.query.implementedBy: 'ABAP:ZCL_CUSTOM_VH'
define custom entity Zcustom_entity_VH
{
key pr_id      : banfn;
@UI.hidden : true
pr_type     : bsart;
@EndUserText.label: 'Requestor'
requester  : ernam;
plant      : werks_d;
@UI.selectionField: [{exclude: true}]
location    : stras;
@UI.hidden : true
@UI.selectionField: [{
exclude: true,
element: 'requester'
}]
created_by : ernam;

}
```

```
CLASS zcl_custom_vh DEFINITION  PUBLIC  FINAL  CREATE PUBLIC .

 PUBLIC SECTION.
  INTERFACES : if_rap_query_provider,
   if_sadl_exit_calc_element_read.
 PROTECTED SECTION.
 PRIVATE SECTION.
ENDCLASS.
```

```
CLASS zcl_custom_vh IMPLEMENTATION.


 METHOD if_rap_query_provider~select.
  CASE io_request->get_entity_id( ).
   WHEN 'ZCUSTOM_ENTITY_VH'.
    IF io_request->is_data_requested( ).
     DATA(lv_top) = io_request->get_paging( )->get_page_size( ).
     IF lv_top < 0.
      lv_top = 1.
     ENDIF.

     DATA(lv_skip) = io_request->get_paging( )->get_offset( ).
     DATA(lv_conditions) = io_request->get_filter( )->get_as_sql_string( ).
     DATA(lv_sort) = io_request->get_sort_elements( ).
     DATA : lv_orderby TYPE string.

     LOOP AT lv_sort INTO DATA(ls_sort).
      IF ls_sort-descending = abap_true.
       lv_orderby = |{ lv_orderby }{ ls_sort-element_name } DESCENDING|.
      ELSE.
       lv_orderby = |{ lv_orderby }{ ls_sort-element_name } ASCENDING|.

      ENDIF.
     ENDLOOP.

     IF lv_orderby IS   INITIAL.
      lv_orderby = 'PR_ID'.
     ENDIF.

     SELECT FROM ymm_pr_hdr
       FIELDS pr_id,requester,pr_type,plant,location,created_by
       WHERE sendmail = 'YES' OR
          (lv_conditions)
```

```
               ORDER BY (lv_orderby)
              INTO TABLE @DATA(lt_data)
            UP TO @lv_top ROWS OFFSET @lv_skip.

          IF io_request->is_total_numb_of_rec_requested( ).
            io_response->set_total_number_of_records( lines( lt_data ) ).
            io_response->set_data( lt_data ).

          ENDIF.
        ENDIF.

      WHEN 'ZCE_CCCODE_VH'.
        IF io_request->is_data_requested( ).
          lv_top = io_request->get_paging( )->get_page_size( ).
          IF lv_top < 0.
            lv_top = 1.
          ENDIF.

          lv_skip = io_request->get_paging( )->get_offset( ).
          lv_conditions = io_request->get_filter( )->get_as_sql_string( ).
          lv_sort = io_request->get_sort_elements( ).

          LOOP AT lv_sort INTO ls_sort.
            IF ls_sort-descending = abap_true.
              lv_orderby = |{ lv_orderby }{ ls_sort-element_name } DESCENDING|.
            ELSE.
              lv_orderby = |{ lv_orderby }{ ls_sort-element_name } ASCENDING|.

            ENDIF.
          ENDLOOP.

          IF lv_orderby IS   INITIAL.
            lv_orderby = 'PR_ID'.
          ENDIF.

*          lv_top = 30.

          SELECT FROM t001
             FIELDS bukrs,butxt,ort01,land1
             WHERE (lv_conditions)
               ORDER BY (lv_orderby)
              INTO TABLE @DATA(lt_data1)
            UP TO @lv_top ROWS OFFSET @lv_skip.

          IF io_request->is_total_numb_of_rec_requested( ).
            io_response->set_total_number_of_records( lines( lt_data1 ) ).
            io_response->set_data( lt_data1 ).

          ENDIF.
        ENDIF.
    ENDCASE.


  ENDMETHOD.


  METHOD if_sadl_exit_calc_element_read~calculate.

    DATA : lt_data TYPE TABLE OF YP_mm_pr_hdr.
    lt_data  = CORRESPONDING #( it_original_data ).
    LOOP AT lt_data ASSIGNING FIELD-SYMBOL(<fs_data>).

      SELECT FROM ymm_pr_hdr
      FIELDS SUM( total_value  )
      WHERE sid = @<fs_data>-Sid AND pr_id = @<fs_data>-PrId
      INTO @DATA(lv_sum1).

      SELECT FROM ymm_pr_itm
      FIELDS SUM( price )
      WHERE sid = @<fs_data>-Sid AND pr_id = @<fs_data>-PrId
      INTO @DATA(lv_sum2).
*      <fs_data>-Remain_amt = lv_sum1 - lv_sum2.
      <fs_data>-remain_amt = lv_sum1 - lv_sum2.
      <fs_data>-total_item_sum = lv_sum1.
    ENDLOOP.

    ct_calculated_data = CORRESPONDING #( lt_data ).

  ENDMETHOD.


  METHOD if_sadl_exit_calc_element_read~get_calculation_info.

  ENDMETHOD.
ENDCLASS.
```

FM



Function Builder: Display YBAPI_PR_CREATE

Function module YBAPI_PR_CREATE Active

Classification

Function Group	ZBAPI_PR_640	Function Group For PR & PO
Short Text	Bapi For PR and PO	

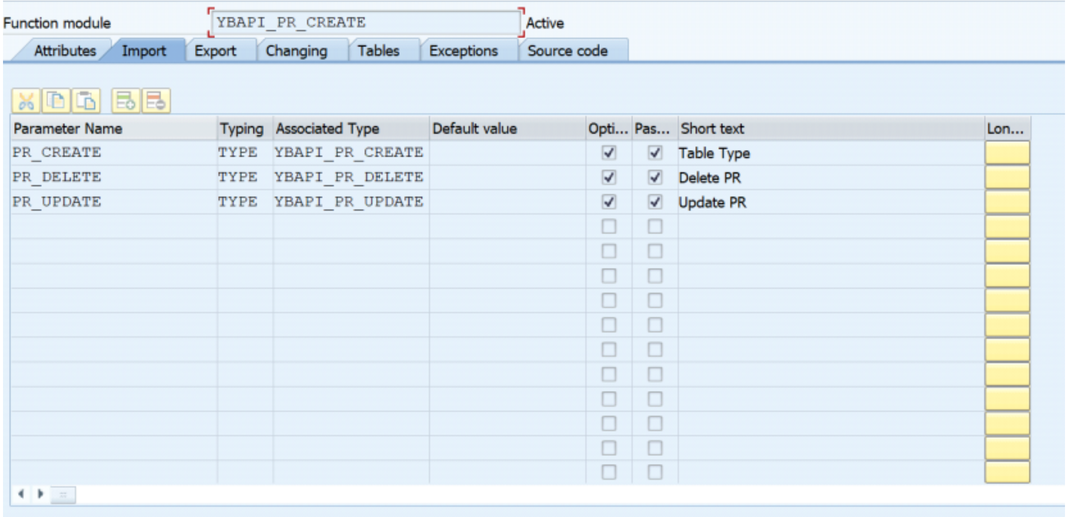
Processing Type

○ Regular Function Module
● Remote-Enabled Module
Scope: Not classified (unrestricted scope)
Interface Contract: Any □ BasXML supported
○ Update Module
● Start immed.
○ Immediate start (not updateable)
○ Start Delayed
○ Coll.run

General Data

Person Responsible	TECH_54
Last Changed By	TECH_54
Changed on	26.12.2025
Package	ZTECH_54_640
Program Name	SAPLZBAPI_PR_640
Include Name	LZBAPI_PR_640U01
Original Language	EN
Released on	26.12.2025

□ Edit Lock
□ Global

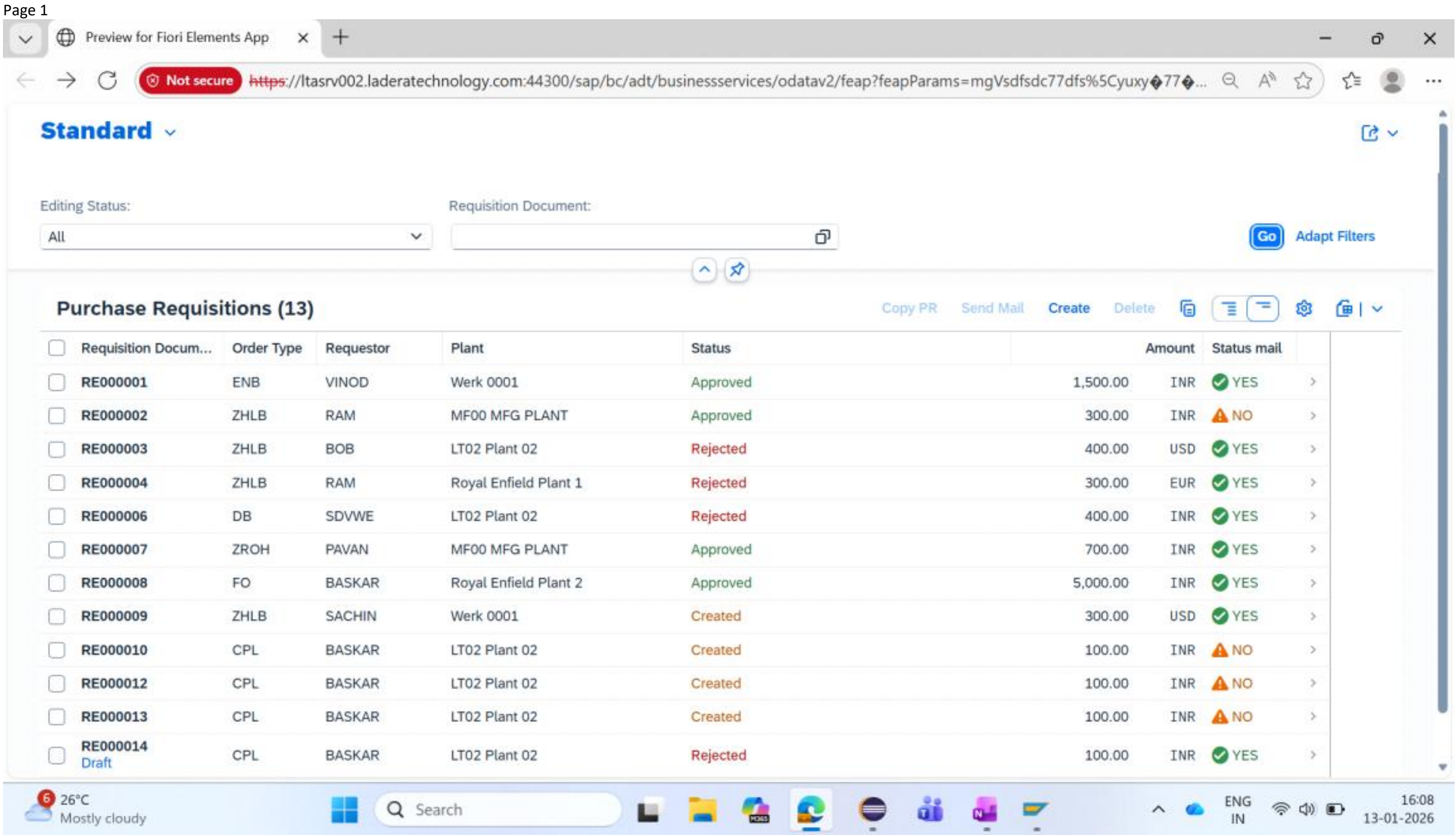


Function module YBAPI_PR_CREATE Active

Parameter Name	Typing	Associated Type	Default value	Opti...	Pas...	Short text	Lon...
PR_CREATE	TYPE	YBAPI_PR_CREATE		☑	☑	Table Type	
PR_DELETE	TYPE	YBAPI_PR_DELETE		☑	☑	Delete PR	
PR_UPDATE	TYPE	YBAPI_PR_UPDATE		☑	☑	Update PR	

```
FUNCTION ybapi_pr_create.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(PR_CREATE) TYPE  YBAPI_PR_CREATE OPTIONAL
*"     VALUE(PR_DELETE) TYPE  YBAPI_PR_DELETE OPTIONAL
*"     VALUE(PR_UPDATE) TYPE  YBAPI_PR_UPDATE OPTIONAL
*"  EXPORTING
*"     VALUE(RETURN) TYPE  BAPIRET2
*"  EXCEPTIONS
*"      FAILED
*"----------------------------------------------------------------------
  IF pr_create IS NOT INITIAL .
    INSERT ymm_pr_hdr FROM TABLE pr_create.
    IF sy-subrc = 0.
      return = VALUE #( type = 'S' id = 'YMSGS_640' number = '003'
                        message_v1 = pr_create[ 1 ]-pr_id message_v2 = pr_create[ 1 ]-sid ).
    ELSE.
      return = VALUE #( type = 'E' id = 'YMSGS_640' number = '004'
                        message_v1 = pr_create[ 1 ]-pr_id message_v2 = pr_create[ 1 ]-sid ).
    ENDIF.

  ENDIF.

  IF pr_update IS NOT INITIAL .
    UPDATE ymm_pr_hdr FROM TABLE pr_update.
    IF sy-subrc = 0.
      return = VALUE #( type = 'S' id = 'YMSGS_640' number = '005'
                        message_v1 = pr_update[ 1 ]-pr_id
                        message_v2 = pr_update[ 1 ]-sid ).
    ELSE.
      return = VALUE #( type = 'E' id = 'ZM_SMSYMSGS_640' number = '006'
                        message_v1 = pr_update[ 1 ]-pr_id
                        message_v2 = pr_update[ 1 ]-sid ).

    ENDIF.
  ENDIF.

  IF pr_delete IS NOT INITIAL .
    DELETE ymm_pr_hdr FROM TABLE pr_delete.
    IF sy-subrc = 0.
      DATA(lv_sid) = pr_delete[ 1 ]-sid.
      DATA(lv_prid) = pr_delete[ 1 ]-pr_id.
      return = VALUE #( type = 'S' id = 'YMSGS_640' number = '007'
                        message_v1 = lv_prid
                        message_v2 = lv_sid ).
      DELETE  FROM ymm_pr_itm WHERE sid = @lv_sid AND pr_id = @lv_prid.
    ELSE.
      return = VALUE #( type = 'E' id = 'YMSGS_640' number = '008'
                        message_v1 = pr_delete[ 1 ]-pr_id
                        message_v2 = pr_delete[ 1 ]-sid ).

    ENDIF.
  ENDIF.


ENDFUNCTION.
```

Page 1


Preview for Fiori Elements App
Not secure
https://ltasrv002.laderatechnology.com:44300/sap/bc/adt/businessservices/odatav2/feap?feapParams=mgVsdfsdc77dfs%5Cyuxy�77�...
Go
Adapt Filters
Copy PR
Send Mail
Create
Delete
26°C
Mostly cloudy
Search
ENG
IN
16:08
13-01-2026

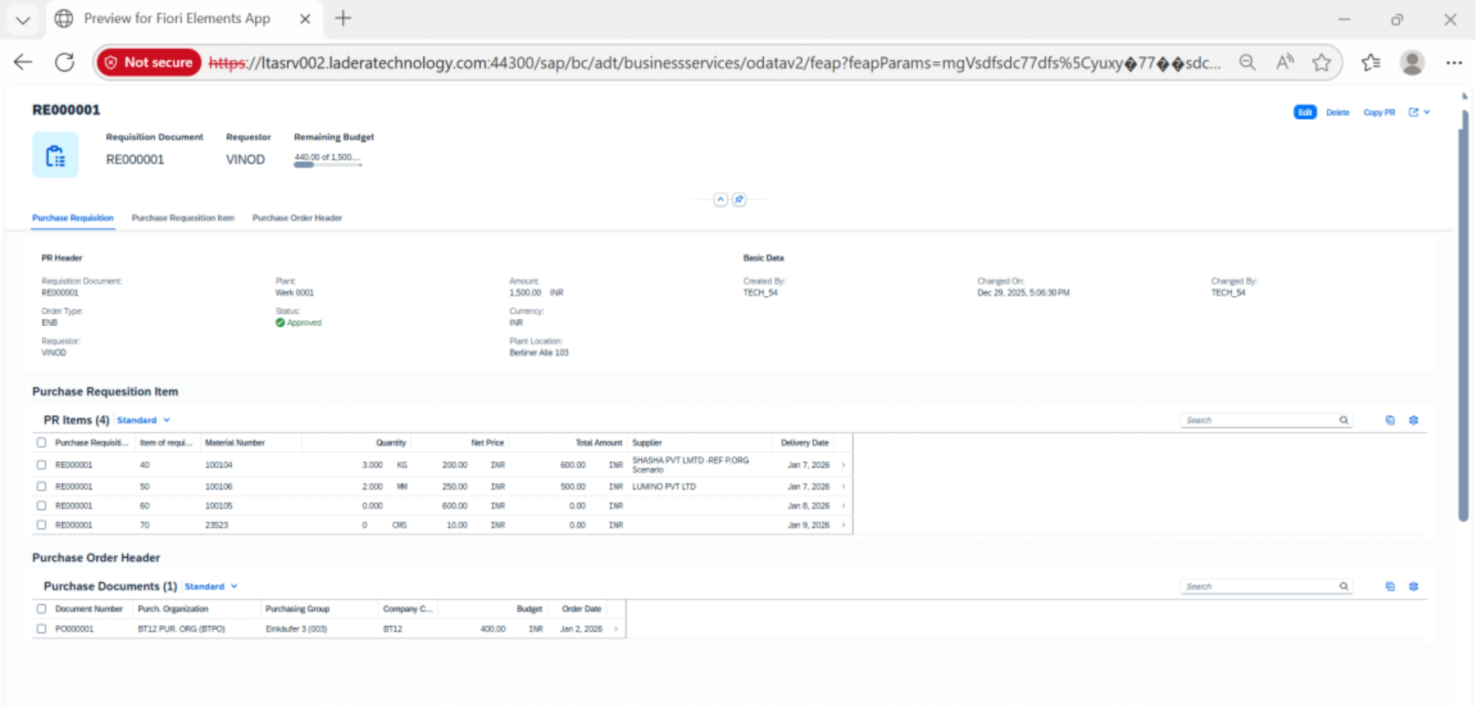
Standard

Editing Status: All

Requisition Document:

Purchase Requisitions (13)

☐	Requisition Docum...	Order Type	Requestor	Plant	Status	Amount		Status mail
☐	RE000001	ENB	VINOD	Werk 0001	Approved	1,500.00	INR	✓ YES
☐	RE000002	ZHLB	RAM	MF00 MFG PLANT	Approved	300.00	INR	⚠ NO
☐	RE000003	ZHLB	BOB	LT02 Plant 02	Rejected	400.00	USD	✓ YES
☐	RE000004	ZHLB	RAM	Royal Enfield Plant 1	Rejected	300.00	EUR	✓ YES
☐	RE000006	DB	SDVWE	LT02 Plant 02	Rejected	400.00	INR	✓ YES
☐	RE000007	ZROH	PAVAN	MF00 MFG PLANT	Approved	700.00	INR	✓ YES
☐	RE000008	FO	BASKAR	Royal Enfield Plant 2	Approved	5,000.00	INR	✓ YES
☐	RE000009	ZHLB	SACHIN	Werk 0001	Created	300.00	USD	✓ YES
☐	RE000010	CPL	BASKAR	LT02 Plant 02	Created	100.00	INR	⚠ NO
☐	RE000012	CPL	BASKAR	LT02 Plant 02	Created	100.00	INR	⚠ NO
☐	RE000013	CPL	BASKAR	LT02 Plant 02	Created	100.00	INR	⚠ NO
☐	RE000014 Draft	CPL	BASKAR	LT02 Plant 02	Rejected	100.00	INR	✓ YES


Preview for Fiori Elements App
Not secure
https://ltasrv002.laderatechnology.com:44300/sap/bc/adt/businessservices/odatav2/feap?feapParams=mgVsdfsdc77dfs%5Cyuxy�77��sdc...
Edit
Delete
Copy PR
Purchase Requisition
Purchase Requesition Item
Purchase Order Header
Search

RE000001

Requisition Document: RE000001
Requestor: VINOD
Remaining Budget: 440.00 of 1,500....

PR Header

Requisition Document: RE000001
Order Type: ENB
Requestor: VINOD
Plant: Werk 0001
Status: ✓ Approved
Amount: 1,500.00 INR
Currency: INR
Plant Location: Berliner Alle 103

Basic Data

Created By: TECH_54
Changed On: Dec 29, 2025, 5:06:30 PM
Changed By: TECH_54

Purchase Requesition Item

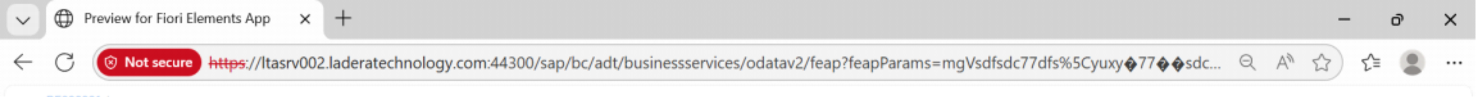
PR Items (4) Standard

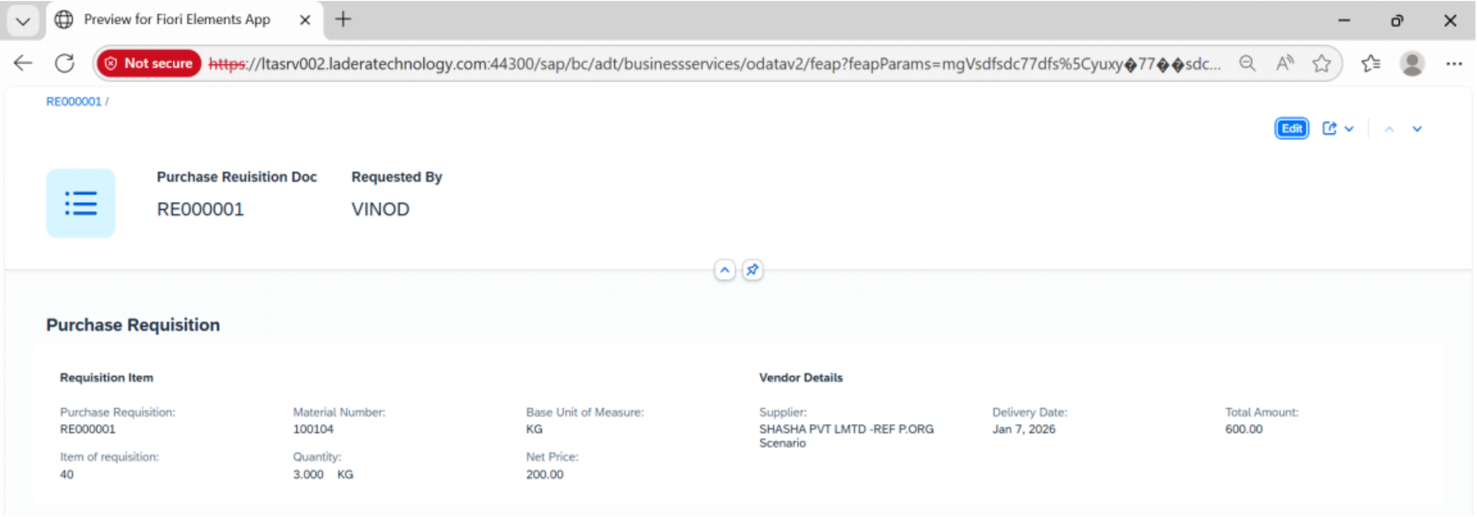
☐	Purchase Requisiti...	Item of requi...	Material Number	Quantity		Net Price		Total Amount		Supplier	Delivery Date
☐	RE000001	40	100104	3.000	KG	200.00	INR	600.00	INR	SHASHA PVT LMTD -REF P.ORG Scenario	Jan 7, 2026
☐	RE000001	50	100106	2.000	MM	250.00	INR	500.00	INR	LUMINO PVT LTD	Jan 7, 2026
☐	RE000001	60	100105	0.000		600.00	INR	0.00	INR		Jan 8, 2026
☐	RE000001	70	23523	0	CMS	10.00	INR	0.00	INR		Jan 9, 2026

Purchase Order Header

Purchase Documents (1) Standard

☐	Document Number	Purch. Organization	Purchasing Group	Company C...	Budget		Order Date
☐	PO000001	BT12 PUR. ORG (BTPO)	Einkäufer 3 (003)	BT12	400.00	INR	Jan 2, 2026


Preview for Fiori Elements App
Not secure
https://ltasrv002.laderatechnology.com:44300/sap/bc/adt/businessservices/odatav2/feap?feapParams=mgVsdfsdc77dfs%5Cyuxy�77��sdc...


Preview for Fiori Elements App
Not secure
https://ltasrv002.laderatechnology.com:44300/sap/bc/adt/businessservices/odatav2/feap?feapParams=mgVsdfsdc77dfs%5Cyuxy�77��sdc...
Edit

RE000001 /

Purchase Reuisition Doc
RE000001

Requested By
VINOD

Purchase Requisition

Requisition Item

Purchase Requisition:
RE000001

Material Number:
100104

Base Unit of Measure:
KG

Item of requisition:
40

Quantity:
3.000 KG

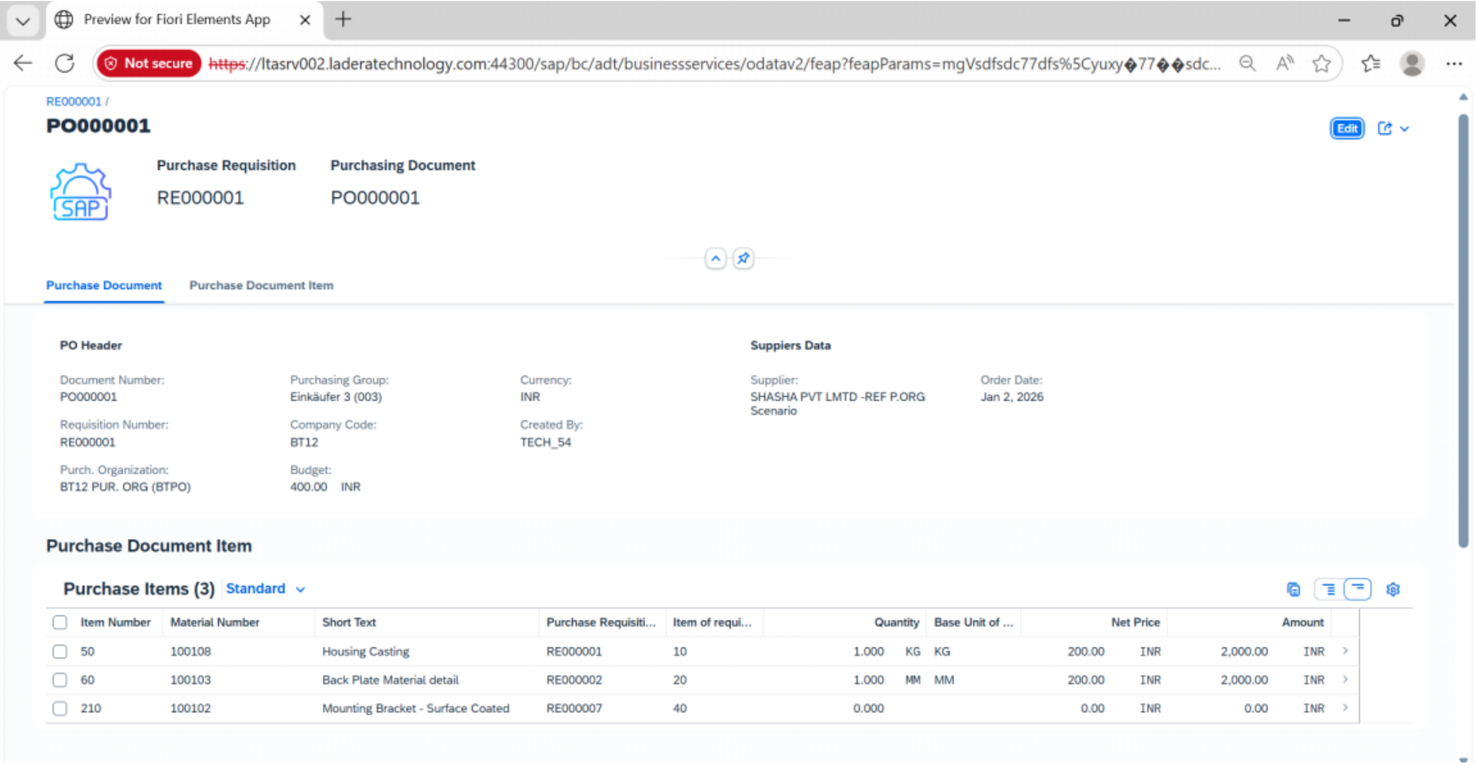
Net Price:
200.00

Vendor Details

Supplier:
SHASHA PVT LMTD -REF P.ORG Scenario

Delivery Date:
Jan 7, 2026

Total Amount:
600.00


Preview for Fiori Elements App
Not secure
https://ltasrv002.laderatechnology.com:44300/sap/bc/adt/businessservices/odatav2/feap?feapParams=mgVsdfsdc77dfs%5Cyuxy�77��sdc...
Edit
SAP

RE000001 /

PO000001

Purchase Requisition
RE000001

Purchasing Document
PO000001

Purchase Document | Purchase Document Item

PO Header

Document Number:
PO000001

Purchasing Group:
Einkäufer 3 (003)

Currency:
INR

Requisition Number:
RE000001

Company Code:
BT12

Created By:
TECH_54

Purch. Organization:
BT12 PUR. ORG (BTPO)

Budget:
400.00 INR

Suppiers Data

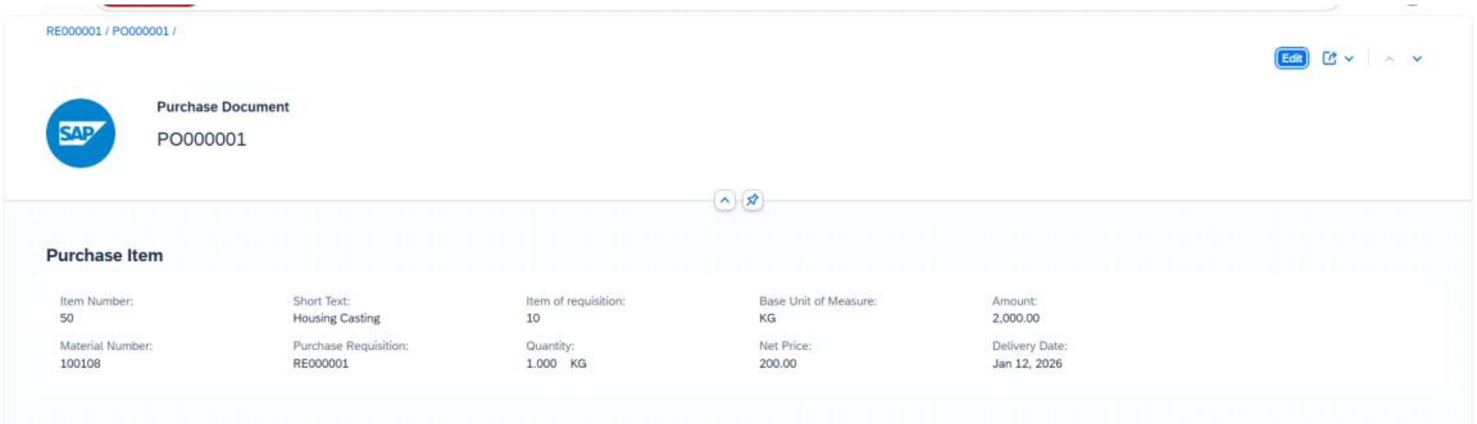
Supplier:
SHASHA PVT LMTD -REF P.ORG Scenario

Order Date:
Jan 2, 2026

Purchase Document Item

Purchase Items (3) Standard

□	Item Number	Material Number	Short Text	Purchase Requisiti...	Item of requi...	Quantity	Base Unit of ...	Net Price	Amount	
□	50	100108	Housing Casting	RE000001	10	1.000 KG	KG	200.00 INR	2,000.00 INR	>
□	60	100103	Back Plate Material detail	RE000002	20	1.000 MM	MM	200.00 INR	2,000.00 INR	>
□	210	100102	Mounting Bracket - Surface Coated	RE000007	40	0.000		0.00 INR	0.00 INR	>



RE000001 / PO000001 /

Purchase Document
PO000001

Purchase Item

Item Number:
50

Short Text:
Housing Casting

Item of requisition:
10

Base Unit of Measure:
KG

Amount:
2,000.00

Material Number:
100108

Purchase Requisition:
RE000001

Quantity:
1.000 KG

Net Price:
200.00

Delivery Date:
Jan 12, 2026