**Restful Application Programming**

**Human Resource Management Application**

1. **Managed Example (Only Employee)**
2. **Unmanaged Example (Employee & Timesheet)**

**Database:**

**Main Tables:**

Department (YHRM\_DEPARTMENT)

|  |
| --- |
| @EndUserText.label : 'Department Table'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #ALLOWED  **define** **table** yhrm\_department **{**  **key** client **:** abap**.**clnt **not** **null;**  **key** department\_id **:** yhrm\_department\_id **not** **null;**  department\_name **:** yhrm\_department\_name**;**  @AbapCatalog.foreignKey.screenCheck : false  hod **:** yhrm\_hod  **with** **foreign** **key** **[**0**..\*,**1**]** yhrm\_employee  **where** client **=** yhrm\_department**.**client  **and** emp\_id **=** yhrm\_department**.**hod**;**  @AbapCatalog.foreignKey.screenCheck : false  address\_id **:** yhrm\_address\_id  **with** **foreign** **key** **[**0**..\*,**1**]** yhrm\_address  **where** client **=** yhrm\_department**.**client  **and** address\_id **=** yhrm\_department**.**address\_id**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  local\_last\_changed\_at **:** abp\_locinst\_lastchange\_tstmpl**;**  last\_changed\_at **:** abp\_lastchange\_tstmpl**;**  **}** |

Employee (YHRM\_EMPOYEE)

|  |
| --- |
| @EndUserText.label : 'Employee table'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #ALLOWED  **define** **table** yhrm\_employee **{**  **key** client **:** abap**.**clnt **not** **null;**  **key** emp\_id **:** yhrm\_emp\_id **not** **null;**  first\_name **:** yhrm\_firstname**;**  last\_name **:** yhrm\_lastname**;**  email **:** yhrm\_email**;**  phone\_no **:** yhrm\_phone\_no**;**  dob **:** yhrm\_dob**;**  gender **:** yhrm\_gender**;**  salary **:** yhrm\_salary**;**  hire\_date **:** yhrm\_hire\_date**;**  active **:** abap\_boolean**;**  resign\_date **:** yhrm\_resign\_date**;**  @AbapCatalog.foreignKey.screenCheck : false  address\_id **:** yhrm\_address\_id  **with** **foreign** **key** **[**0**..\*,**1**]** yhrm\_address  **where** client **=** yhrm\_employee**.**client  **and** address\_id **=** yhrm\_employee**.**address\_id**;**  @AbapCatalog.foreignKey.screenCheck : false  job\_id **:** yhrm\_job\_id  **with** **foreign** **key** **[**0**..\*,**1**]** yhrm\_job  **where** client **=** yhrm\_employee**.**client  **and** job\_id **=** yhrm\_employee**.**job\_id**;**  @AbapCatalog.foreignKey.screenCheck : false  department\_id **:** yhrm\_department\_id  **with** **foreign** **key** **[**0**..\*,**1**]** yhrm\_department  **where** client **=** yhrm\_employee**.**client  **and** department\_id **=** yhrm\_employee**.**department\_id**;**  @AbapCatalog.foreignKey.screenCheck : false  supervisor\_id **:** yhrm\_supervisor\_id  **with** **foreign** **key** **[**0**..\*,**1**]** yhrm\_employee  **where** client **=** yhrm\_employee**.**client  **and** emp\_id **=** yhrm\_employee**.**supervisor\_id**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  local\_last\_changed\_at **:** abp\_locinst\_lastchange\_tstmpl**;**  last\_changed\_at **:** abp\_lastchange\_tstmpl**;**  **}** |

Job (YHRM\_JOB)

|  |
| --- |
| @EndUserText.label : 'Job Table'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #ALLOWED  **define** **table** yhrm\_job **{**  **key** client **:** abap**.**clnt **not** **null;**  **key** job\_id **:** yhrm\_job\_id **not** **null;**  job\_title **:** yhrm\_job\_title**;**  job\_type **:** yhrm\_job\_type**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  local\_last\_changed\_at **:** abp\_locinst\_lastchange\_tstmpl**;**  last\_changed\_at **:** abp\_lastchange\_tstmpl**;**  **}** |

Address (YHRM\_ADDRESS)

|  |
| --- |
| @EndUserText.label : 'Address table'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #ALLOWED  **define** **table** yhrm\_address **{**  **key** client **:** abap**.**clnt **not** **null;**  **key** address\_id **:** yhrm\_address\_id **not** **null;**  street\_add1 **:** yhrm\_street\_add**;**  street\_add2 **:** yhrm\_street\_add**;**  street\_add3 **:** yhrm\_street\_add**;**  pin\_code **:** yhrm\_pin\_code**;**  city **:** yhrm\_city**;**  state **:** yhrm\_state**;**  country **:** yhrm\_country**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  local\_last\_changed\_at **:** abp\_locinst\_lastchange\_tstmpl**;**  last\_changed\_at **:** abp\_lastchange\_tstmpl**;**  **}** |

Timesheet (YHRM\_TIMESHEET)

|  |
| --- |
| @EndUserText.label : 'Timesheet table'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #ALLOWED  **define** **table** yhrm\_timesheet **{**  **key** client **:** abap**.**clnt **not** **null;**  **key** empid **:** yhrm\_emp\_id **not** **null;**  **key** ydate **:** yhrm\_date **not** **null;**  available **:** abap\_boolean**;**  workinghours **:** yhrm\_hours**;**  leavetype **:** yhrm\_leavetype**;**  overtime\_hrs **:** yhrm\_hours**;**  @AbapCatalog.foreignKey.screenCheck : false  approved\_by **:** yhrm\_emp\_id  **with** **foreign** **key** **[**0**..\*,**1**]** yhrm\_employee  **where** client **=** yhrm\_timesheet**.**client  **and** emp\_id **=** yhrm\_timesheet**.**approved\_by**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  local\_last\_changed\_at **:** abp\_locinst\_lastchange\_tstmpl**;**  last\_changed\_at **:** abp\_lastchange\_tstmpl**;**  **}** |

A screenshot of a computer program

Description automatically generated

**Draft Tables:** (Can create manually or using Quick Fix in Behavior definition)

YHRM\_DEPART\_D (Used in Managed app)

|  |
| --- |
| @EndUserText.label : 'Draft Department Table'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #RESTRICTED  **define** **table** yhrm\_depart\_d **{**  **key** mandt **:** mandt **not** **null;**  **key** departmentid **:** yhrm\_department\_id **not** **null;**  departmentname **:** yhrm\_department\_name**;**  hod **:** yhrm\_hod**;**  addressid **:** yhrm\_address\_id**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  local\_last\_changed\_at **:** abp\_locinst\_lastchange\_tstmpl**;**  last\_changed\_at **:** abp\_lastchange\_tstmpl**;**  "%admin" **:** **include** sych\_bdl\_draft\_admin\_inc**;**  **}** |

YHRM\_EMPLOYEE\_D (Used in Managed app)

|  |
| --- |
| @EndUserText.label : 'Draft Employee Table'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #RESTRICTED  **define** **table** yhrm\_employee\_d **{**  **key** mandt **:** mandt **not** **null;**  **key** empid **:** yhrm\_emp\_id **not** **null;**  employeename **:** abap**.**char**(**130**);**  firstname **:** yhrm\_firstname**;**  lastname **:** yhrm\_lastname**;**  email **:** yhrm\_email**;**  phoneno **:** yhrm\_phone\_no**;**  dob **:** yhrm\_dob**;**  gender **:** yhrm\_gender**;**  salary **:** yhrm\_salary**;**  hiredate **:** yhrm\_hire\_date**;**  active **:** abap\_boolean**;**  resigndate **:** yhrm\_resign\_date**;**  addressid **:** yhrm\_address\_id**;**  jobid **:** yhrm\_job\_id**;**  departmentid **:** yhrm\_department\_id**;**  supervisorid **:** yhrm\_supervisor\_id**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  local\_last\_changed\_at **:** abp\_locinst\_lastchange\_tstmpl**;**  last\_changed\_at **:** abp\_lastchange\_tstmpl**;**  "%admin" **:** **include** sych\_bdl\_draft\_admin\_inc**;**  **}** |

YHEM\_EMP\_D\_UN (Used in unmanaged app)

|  |
| --- |
| @EndUserText.label : 'Draft table for entity YHRM\_U\_EMPLOYEE unmanaged'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #RESTRICTED  **define** **table** yhrm\_emp\_d\_un **{**  **key** mandt **:** mandt **not** **null;**  **key** empid **:** yhrm\_emp\_id **not** **null;**  firstname **:** yhrm\_firstname**;**  lastname **:** yhrm\_lastname**;**  email **:** yhrm\_email**;**  phoneno **:** yhrm\_phone\_no**;**  dob **:** yhrm\_dob**;**  gender **:** yhrm\_gender**;**  salary **:** yhrm\_salary**;**  hiredate **:** yhrm\_hire\_date**;**  active **:** abap\_boolean**;**  resigndate **:** yhrm\_resign\_date**;**  addressid **:** yhrm\_address\_id**;**  jobid **:** yhrm\_job\_id**;**  departmentid **:** yhrm\_department\_id**;**  supervisorid **:** yhrm\_supervisor\_id**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  locallastchangedat **:** abp\_locinst\_lastchange\_tstmpl**;**  lastchangedat **:** abp\_lastchange\_tstmpl**;**  jobtitle **:** yhrm\_job\_title**;**  departmentname **:** yhrm\_department\_name**;**  supervisorname **:** yhrm\_firstname**;**  "%admin" **:** **include** sych\_bdl\_draft\_admin\_inc**;**  **}** |

YHEM\_TIMESHEET\_D (Used in unmanaged app)

|  |
| --- |
| @EndUserText.label : 'Draft table for entity YHRM\_U\_TIMESHEET'  @AbapCatalog.enhancement.category : #EXTENSIBLE\_ANY  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #RESTRICTED  **define** **table** yhrm\_timesheet\_d **{**  **key** mandt **:** mandt **not** **null;**  **key** empid **:** yhrm\_emp\_id **not** **null;**  **key** ydate **:** yhrm\_date **not** **null;**  available **:** abap\_boolean**;**  workinghours **:** yhrm\_hours**;**  leavetype **:** yhrm\_leavetype**;**  overtimehrs **:** yhrm\_hours**;**  approvedby **:** yhrm\_emp\_id**;**  created\_by **:** syuname**;**  created\_at **:** timestampl**;**  last\_changed\_by **:** syuname**;**  local\_last\_changed\_by **:** abp\_locinst\_lastchange\_user**;**  locallastchangedat **:** abp\_locinst\_lastchange\_tstmpl**;**  lastchangedat **:** abp\_lastchange\_tstmpl**;**  "%admin" **:** **include** sych\_bdl\_draft\_admin\_inc**;**  **}** |

**Class for insert the data:**  
Class ZDP\_CL\_POULAE\_EMPLOYEE ([link](https://github.com/Smitpadshala99/YHRM_RAP_APPLICATION/blob/main/YHRM%20RAP%20Application/Source%20Code%20Library/Classes/zdp_cl_populate_employee.txt))

**Help View F4 CDS:**

YHRM\_F4\_ADDRESS

|  |
| --- |
| @AbapCatalog.viewEnhancementCategory: [#NONE]  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Help view F4 for address'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #X,  sizeCategory: #S,  dataClass: #MIXED  }  @Search.searchable: true  **define** **view** **entity** YHRM\_F4\_ADDRESS **as** **select** **from** yhrm\_address  **{**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  **key** address\_id **as** AddressId**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  street\_add1 **as** StreetAdd1**,**  @Search.defaultSearchElement: true  pin\_code **as** PinCode**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  city **as** City**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  state **as** State**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  country **as** Country  **}** |

YHRM\_F4\_DEPARTMENT

|  |
| --- |
| @AbapCatalog.viewEnhancementCategory: [#NONE]  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Help view F4 for department'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #X,  sizeCategory: #S,  dataClass: #MIXED  }  @ObjectModel.resultSet.sizeCategory: #XS  **define** **view** **entity** YHRM\_F4\_DEPARTMENT **as** **select** **from** yhrm\_department  **{**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  **key** department\_id **as** DepartmentId**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  department\_name **as** DepartmentName  **,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  hod **as** HOD  **}** |

YHRM\_F4\_EMPLOYEE

|  |
| --- |
| @AbapCatalog.viewEnhancementCategory: [#NONE]  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Help view F4 for employee'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #X,  sizeCategory: #S,  dataClass: #MIXED  }  @Search.searchable: true  **define** **view** **entity** YHRM\_F4\_EMPLOYEE **as** **select** **from** yhrm\_employee  **{**  @Search.defaultSearchElement: true  @ObjectModel.text.element: ['FirstName']  @EndUserText.label: 'Employee Id'  **key** emp\_id **as** EmpId**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  first\_name **as** FirstName**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  last\_name **as** LastName**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  gender **as** Gender**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Active'  active **as** Active**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Supervisor Id'  supervisor\_id **as** SupervisorId  **}** |

YHRM\_F4\_GENDER

|  |
| --- |
| @AbapCatalog.viewEnhancementCategory: [#NONE]  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Help view F4 for Gender'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #X,  sizeCategory: #S,  dataClass: #MIXED  }  @ObjectModel.resultSet.sizeCategory: #XS  **define** **view** **entity** YHRM\_F4\_GENDER **as** **select** **from** DDCDS\_CUSTOMER\_DOMAIN\_VALUE\_T**(** p\_domain\_name**:** 'YHRM\_GENDER' **)**  **{**  @UI.hidden: true  **key** domain\_name**,**  @UI.hidden: true  **key** value\_position**,**  @UI.hidden: true  @Semantics.language: true  **key** language**,**  value\_low **as** Value**,**  @Semantics.text: true  text **as** Description  **}** |

YHRM\_F4\_JOB

|  |
| --- |
| @AbapCatalog.viewEnhancementCategory: [#NONE]  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Help view F4 for job'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #X,  sizeCategory: #S,  dataClass: #MIXED  }  @ObjectModel.resultSet.sizeCategory: #XS  **define** **view** **entity** YHRM\_F4\_JOB **as** **select** **from** yhrm\_job  **{**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  **key** job\_id **as** JobId**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  job\_title **as** JobTitle**,**  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  job\_type **as** JobType  **}** |

YHRM\_F4\_LEAVETYPE

|  |
| --- |
| @AbapCatalog.viewEnhancementCategory: [#NONE]  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Help view F4 for Leave Type'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #X,  sizeCategory: #S,  dataClass: #MIXED  }  @ObjectModel.resultSet.sizeCategory: #XS  **define** **view** **entity** YHRM\_F4\_LEAVETYPE **as** **select** **from** DDCDS\_CUSTOMER\_DOMAIN\_VALUE\_T**(** p\_domain\_name**:** 'YHRM\_LEAVETYPE' **)**  **{**  @UI.hidden: true  **key** domain\_name**,**  @UI.hidden: true  **key** value\_position**,**  @UI.hidden: true  @Semantics.language: true  **key** language**,**  @ObjectModel.text.element: [ 'Description' ]  value\_low **as** Value**,**  @Semantics.text: true  @UI.hidden: true  text **as** Description  **}** |

NOTE:

* **@Search.searchable: true** Makes the element searchable.
* **@Search.defaultSearchElement: true** Sets the element as the default for searches.
* **@Search.fuzzinessThreshold: 0.8:** Sets the similarity threshold for search matches to 80%.
* **@ObjectModel.text.element: ['FirstName']:** Specifies ‘FirstName’ as the text element in the object model.
* **@EndUserText.label: 'Employee Id':** Sets the label of the element for end users as ‘Employee Id’.
* **@ObjectModel.resultSet.sizeCategory: #XS:** Specifies the expected size of the result set as ‘Extra Small’.
* select from **DDCDS\_CUSTOMER\_DOMAIN\_VALUE\_T( p\_domain\_name: 'YHRM\_GENDER' ):** Selects data from the DDCDS\_CUSTOMER\_DOMAIN\_VALUE\_T table where the domain name is ‘YHRM\_GENDER’.
* **@UI.hidden: true:** Hides the associated element in the UI.
* **@Semantics.language: true:** Indicates that the associated element represents a language.
* **@Semantics.text: true:** Indicates that the associated element represents a text.

**Managed Example (Only Employee)**

A diagram of a company

Description automatically generated

**YHRM\_EMPLOYEE** (Dictionary / Database Tables) (see at Employee (YHRM\_EMPOYEE)

**YHRM\_EMPLOYEE\_D** (Dictionary / Database Tables) (see at (YHRM\_EMPLOYEE\_D (Used in Managed app)

))

**YHRM\_I\_EMPLOYEE\_APP** (CDS / Data Definition)

|  |
| --- |
| @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Employee Transactional App'  **define** **root** **view** **entity** YHRM\_I\_EMPLOYEE\_APP **as** **select** **from** yhrm\_employee  **{**  **key** emp\_id **as** EmpId**,**  concat**(**first\_name**,** last\_name **)** **as** EmployeeName**,**  // concat\_with\_space( first\_name,last\_name,1) as EmployeeName,  first\_name **as** FirstName**,**  last\_name **as** LastName**,**  email **as** Email**,**  phone\_no **as** PhoneNo**,**  dob **as** Dob**,**  gender **as** Gender**,**  salary **as** Salary**,**  hire\_date **as** HireDate**,**  active **as** Active**,**  resign\_date **as** ResignDate**,**  address\_id **as** AddressId**,**  job\_id **as** JobId**,**  department\_id **as** DepartmentId**,**  supervisor\_id **as** SupervisorId**,**    @Semantics.user.createdBy: true  created\_by**,**  @Semantics.systemDateTime.createdAt: true  created\_at**,**  @Semantics.user.lastChangedBy: true  last\_changed\_by**,**  @Semantics.user.localInstanceLastChangedBy: true  local\_last\_changed\_by**,**  @Semantics.systemDateTime.localInstanceLastChangedAt: true  local\_last\_changed\_at**,**  @Semantics.systemDateTime.lastChangedAt: true  last\_changed\_at  **}** |

NOTE:

* **root:** The root keyword is used to define the main or base view entity in the RAP model. It’s the starting point of the model and all other entities are defined in relation to it.
* **concat:** concat is a function used to combine two or more strings into one. In this case, it’s used to combine first\_name and last\_name to form EmployeeName.

**YHRM\_C\_EMPLOYEE\_APP** (CDS / Data Definition)

|  |
| --- |
| @EndUserText.label: 'Projection view of Employee App'  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @Metadata.allowExtensions: true  **define** **root** **view** **entity** YHRM\_C\_EMPLOYEE\_APP **provider** **contract** **transactional\_query** **as** **projection** **on** YHRM\_I\_EMPLOYEE\_APP  **{**  @ObjectModel.text.element: [ 'EmployeeName' ]  **key** EmpId**,**  EmployeeName**,**  FirstName**,**  LastName**,**  Email**,**  PhoneNo**,**  Dob**,**  Gender**,**  Salary**,**  HireDate**,**  Active**,**  ResignDate**,**  AddressId**,**  JobId**,**  DepartmentId**,**  SupervisorId**,**  created\_by**,**  created\_at**,**  last\_changed\_by**,**  local\_last\_changed\_by**,**  local\_last\_changed\_at**,**  last\_changed\_at  **}** |

NOTE:

* **@Metadata.allowExtensions: true** enables browser extensions, enhancing web application functionality.
* **provider contract transactional\_query**: This is a type of provider contract in RAP. It’s used to define the capabilities of the view entity. transactional\_query means the entity supports read and write operations in a transactional context.
* **projection on:** This keyword is used to define a projection view entity. A projection is a subset of a base view entity. In this case, YHRM\_C\_EMPLOYEE\_APP is a projection on YHRM\_I\_EMPLOYEE\_APP.
* **ObjectModel.text.element:** This annotation is used to specify the text elements for a view entity. Text elements are used for UI display purposes. In this case, EmployeeName is defined as a text element.

**YHRM\_MDE\_C\_EMP\_APP** (CDS / Metadata Extension)

|  |
| --- |
| @Metadata.layer: #CORE  @UI.headerInfo: {  typeName: 'Employee',  typeNamePlural: 'Employees',  title: { value: 'EmpId'},  description: { value: 'EmployeeName' },  typeImageUrl: 'sap-icon://employee'  }  @UI.presentationVariant: [{  sortOrder: [ { by: 'EmpId', direction: #DESC } ],  visualizations: [{ type: #AS\_LINEITEM}]  }]  @Search.searchable: true  **annotate** **view** YHRM\_C\_EMPLOYEE\_APP  **with**  **{**  @UI.facet: [  {  id: 'Employee',  purpose: #HEADER,  type: #DATAPOINT\_REFERENCE,  position: 10,  targetQualifier: 'Emp'  }  ,  {  id: 'EmployeeEmail',  purpose: #HEADER,  type: #DATAPOINT\_REFERENCE,  position: 11,  targetQualifier: 'Email'  }  ,  {  id: 'EmployeePhoneNo',  purpose: #HEADER,  type: #DATAPOINT\_REFERENCE,  position: 20,  targetQualifier: 'PhoneNo'  },  {  id: 'EmployeeInfo',  type: #COLLECTION,  label: 'Employee Info',  position: 10  },  {  id: 'Employee',  type: #IDENTIFICATION\_REFERENCE,  purpose: #STANDARD,  label: 'Employee Professional Info',  parentId: 'EmployeeInfo',  position: 10  },  {  id: 'EmployeePersonaldata',  type: #FIELDGROUP\_REFERENCE,  purpose: #STANDARD,  label: 'Employee Personal Info',  parentId: 'EmployeeInfo',  position: 20,  targetQualifier: 'Info'  }  ]    @UI.selectionField: [{ position: 10}]  @UI.lineItem: [{ position: 10 ,label: 'Employee Id' , cssDefault.width: '10rem'} ]  @UI.identification: [{ position: 1 , label: 'Employee Id'}]  @Search.defaultSearchElement: true  @EndUserText.label: 'Employee Id'  @Consumption.valueHelpDefinition: [{ entity: { element: 'EmpId' , name: 'YHRM\_F4\_EMPLOYEE'} }]  EmpId**;**  @UI.hidden: true  @EndUserText.label: 'EmployeeName'  EmployeeName**;**  @UI : { lineItem: [ { position: 20 , label: 'First Name' , cssDefault.width: '6rem'} ] }  @UI.fieldGroup: [{ position: 1, qualifier: 'Info', label: 'First Name' }]  @UI.dataPoint: { qualifier: 'Emp' , title: 'FirstName' }  FirstName**;**  @UI : { lineItem: [ { position: 21 , label: 'Last Name' , cssDefault.width: '6rem'} ] }  @UI.fieldGroup: [{ position: 2, qualifier: 'Info', label: 'Last Name' }]  LastName**;**  @UI : { selectionField: [{ position: 30}] ,  lineItem: [ { position: 30 , label: 'Department Id'} ],  identification: [{ position: 10 , label: 'Department Id'}]  }  @Consumption.valueHelpDefinition: [{ entity: { element: 'DepartmentId' , name: 'YHRM\_F4\_DEPARTMENT'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Department Id'  DepartmentId**;**  @UI : { selectionField: [{ position: 40}] ,  lineItem: [ { position: 40 , label: 'Job Id'} ],  identification: [{ position: 20 , label: 'Job Id'}] }  @Consumption.valueHelpDefinition: [{ entity: { element: 'JobId' , name: 'YHRM\_F4\_JOB'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Job Id'  JobId**;**  @UI : { identification: [{ position: 50 , label: 'Supervisor Id'}] ,  lineItem: [ { position: 50 , label: 'Supervisor Id'} ]}  @Consumption.valueHelpDefinition: [{ entity: { element: 'EmpId' , name: 'YHRM\_F4\_EMPLOYEE'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Supervisor Id'  SupervisorId**;**  @UI.lineItem: [{ position: 22 , label: 'Email', cssDefault.width: '12rem'}]  @UI.dataPoint: { qualifier: 'Email' , title: 'Email' }  @UI.fieldGroup: [{ position: 4, qualifier: 'Info' }]  Email**;**  @UI.lineItem: [{ position: 23 , label: 'Phone Number', cssDefault.width: '8rem'}]  @UI.dataPoint: { qualifier: 'PhoneNo' , title: 'Phone Number'}  @UI.fieldGroup: [{ position: 5, qualifier: 'Info' , label: 'Phone Number'}]  PhoneNo**;**  @UI.lineItem: [{ position: 24 , cssDefault.width: '8rem'}]  @UI.fieldGroup: [{ position: 10, qualifier: 'Info' }]  Dob**;**  @UI.lineItem: [{ position: 25 ,label: 'Gender', cssDefault.width: '4rem'}]  @UI.fieldGroup: [{ position: 20, qualifier: 'Info' ,label: 'Gender'}]  @Consumption.valueHelpDefinition: [{ entity: { element: 'Value' , name: 'YHRM\_F4\_GENDER'} }]  Gender**;**  @UI : {  // selectionField: [{ position: 50}] ,  lineItem: [ { position: 29 , label: 'Address Id', cssDefault.width: '5rem'} ],  identification: [{ position: 30, label: 'Address Id' }] }  @Consumption.valueHelpDefinition: [{ entity: { element: 'AddressId' , name: 'YHRM\_F4\_ADDRESS'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Address Id'  AddressId**;**  @UI.lineItem: [{ position: 27 , label: 'Hire Date', cssDefault.width: '8rem'}]  @UI.fieldGroup: [{ position: 30, qualifier: 'Info' , label: 'Hire Date' }]  HireDate**;**  @UI.lineItem: [{ position: 28 ,label: 'Active', cssDefault.width: '5rem'}]  @UI.fieldGroup: [{ position: 40, qualifier: 'Info' ,label: 'Active'}]  Active**;**  @UI.hidden: true  created\_by**;**  @UI.hidden: true  created\_at**;**  @UI.hidden: true  last\_changed\_by**;**  @UI.hidden: true  local\_last\_changed\_by**;**  @UI.hidden: true  local\_last\_changed\_at**;**  @UI.hidden: true  last\_changed\_at**;**  **}** |

NOTE:

* **@UI.headerInfo:** Defines the header information for the entity.
* **@UI.presentationVariant**: Specifies the sort order and visualization type for the entity.
* **@UI.facet:** Outlines the structure and layout of the entity’s user interface.
* **@UI.selectionField:** Determines the fields available for selection in the entity.
* **@UI.lineItem:** Defines the line item ‘Employee Id’ in the ‘Employee’ entity.
* **@UI.identification:** Specifies the identification field ‘Employee Id’ for the ‘Employee’ entity.
* **@Search.defaultSearchElement:** Sets ‘Employee Id’ as the default search element in the ‘Employee’ entity.
* **@EndUserText.label:** Provides a user-friendly label ‘Employee Id’ for the ‘Employee’ entity.
* **@Consumption.valueHelpDefinition:** Defines the value help for the ‘EmpId’ element in the ‘Employee’ entity.
* **@UI.fieldGroup:** Groups related fields under the label ‘First Name’ in the ‘Employee’ entity.
* **@UI.dataPoint:** Defines a data point ‘FirstName’ for the ‘Employee’ entity.
* **cssDefault.width: 5rem:** Sets the default width of the UI element to 5rem, controlling its size for better layout and readability.
* **position: 40:** Determines the position of the UI element in the layout, helping to organize the interface.
* **qualifier: 'Info':** Provides a unique identifier ‘Info’ for the UI element, aiding in its referencing and customization.
* **type: #DATAPOINT\_REFERENCE:** Refers to a specific data point in the ‘Employee’ entity, allowing for targeted data manipulation and display.
* **type: #IDENTIFICATION\_REFERENCE:** Identifies a specific element in the ‘Employee’ entity, aiding in entity navigation and data retrieval.
* **type: #FIELDGROUP\_REFERENCE:** Groups related fields together in the ‘Employee’ entity, enhancing data organization and user interface structure.

**YHRM\_I\_EMPLOYEE\_APP** (CDS / Behavior Definition)

|  |
| --- |
| **managed** **implementation** **in** **class** zbp\_yhrm\_i\_employee\_app **unique;**  **strict** **(** 2 **);**  **with** **draft;**  **define** **behavior** **for** YHRM\_I\_EMPLOYEE\_APP **alias** Employee  **persistent** **table** yhrm\_employee  **draft** **table** yhrm\_employee\_d  **lock** **master** **total** **etag** last\_changed\_at  **authorization** **master** **(** **instance** **)**  **etag** **master** local\_last\_changed\_at  **{**  **create;**  **update;**  **delete;**  **field** **(** **readonly** **)** created\_at**,** created\_by**,** last\_changed\_at**,** last\_changed\_by**,** local\_last\_changed\_at**,** local\_last\_changed\_by**;**  **field** **(** **readonly** **:** **update** **)** EmpId**;**  **field** **(** **mandatory** **:** **create** **)** EmpId**;**  // , FirstName, DepartmentId, Email, HireDate;  **draft** **action** Edit**;**  **draft** **action** Activate **optimized;**  **draft** **action** Discard**;**  **draft** **action** Resume**;**  **draft** **determine** **action** Prepare**;**  **mapping** **for** yhrm\_employee**{**  EmpId **=** emp\_id**;**  FirstName **=** first\_name**;**  LastName **=** last\_name**;**  DepartmentId **=** department\_id**;**  AddressId **=** address\_id**;**  Active **=** active**;**  Dob **=** dob**;**  Email **=** email**;**  Gender **=** gender**;**  HireDate **=** hire\_date**;**  JobId **=** job\_id**;**  PhoneNo **=** phone\_no**;**  ResignDate **=** resign\_date**;**  Salary **=** salary**;**  SupervisorId **=** supervisor\_id**;**  **}**  **}** |

NOTE:

* **managed implementation in class zbp\_yhrm\_i\_employee\_app unique: Specifies** a unique managed implementation for the ‘Employee’ entity.
* **strict ( 2 ):** Enforces strict mode with a level of 2, ensuring rigorous error checking.
* **with draft:** Enables draft capabilities, allowing for changes to be saved without immediate effect.
* **persistent table yhrm\_employee:** Defines the persistent data table for the ‘Employee’ entity.
* **draft table yhrm\_employee\_d:** Specifies the draft data table for the ‘Employee’ entity.
* **lock master total etag last\_changed\_at:** Implements a master lock and uses ‘last\_changed\_at’ as the ETag for concurrency control.
* **authorization master ( instance ):** Sets the authorization at the master level for each instance of the ‘Employee’ entity.
* **create; update; delete;:** Defines the CRUD operations available for the ‘Employee’ entity.
* **field ( readonly ) created\_at, created\_by, ...: Specifies** certain fields as read-only, preventing modification.
* **field ( readonly : update ) EmpId;:** Sets ‘EmpId’ as a read-only field during updates.
* **field ( mandatory : create ) EmpId;:** Makes ‘EmpId’ a mandatory field during creation.
* draft action Edit; Activate; Discard; Resume; Prepare; : Defines the draft actions available for the ‘Employee’ entity.
* **mapping for yhrm\_employee:** Maps the fields of the ‘Employee’ entity to their corresponding database columns.
* For the **managed** scenario **no need to write a logic** in class zbp\_yhrm\_i\_employee\_app

**ZBP\_YHRM\_I\_EMPLOYEE\_APP** (Source Code Library / Classes)

|  |
| --- |
| CLASS zbp\_yhrm\_i\_employee\_app DEFINITION PUBLIC ABSTRACT FINAL FOR BEHAVIOR OF yhrm\_i\_employee\_app.  ENDCLASS.  CLASS zbp\_yhrm\_i\_employee\_app IMPLEMENTATION.  ENDCLASS. |

Local types

|  |
| --- |
| CLASS lhc\_employee DEFINITION INHERITING FROM cl\_abap\_behavior\_handler.  PRIVATE SECTION.  METHODS get\_instance\_authorizations FOR INSTANCE AUTHORIZATION  IMPORTING keys REQUEST requested\_authorizations FOR employee RESULT result.  ENDCLASS.  CLASS lhc\_employee IMPLEMENTATION.  METHOD get\_instance\_authorizations.  ENDMETHOD.  ENDCLASS. |

NOTE:

* For the managed scenario, there is no need to write code. Instead, you can use the ‘Quick Fix’ feature to implement the class.

**YHRM\_C\_EMPLOYEE\_APP** (CDS / Behavior Definition)

|  |
| --- |
| **projection;**  **strict** **(** 2 **);**  **use** **draft;**  **define** **behavior** **for** YHRM\_C\_EMPLOYEE\_APP **alias** Employee  **use** **etag**  **{**  **use** **create;**  **use** **update;**  **use** **delete;**  **use** **action** Edit**;**  **use** **action** Activate**;**  **use** **action** Discard**;**  **use** **action** Resume**;**  **use** **action** Prepare**;**  **}** |

NOTE:

* **projection:** Specifies that the behavior is for a projection view of the ‘Employee’ entity.
* To generate the behavior definition code for the ‘YHRM\_C\_EMPLOYEE\_APP’ CDS view, you can follow these steps:

1. Right-click on ‘YHRM\_C\_EMPLOYEE\_APP’ (which is a CDS/Data Definition).
2. Select the ‘Create Behavior Definition’ option.

This action will automatically generate the behavior definition code for you.

**YHRM\_SD\_M\_EMPLOYEE\_APP** (Business Services / Service Definition)

|  |
| --- |
| @EndUserText.label: 'SD for Employee App'  **define** **service** YHRM\_SD\_M\_EMPLOYEE\_APP **{**  **expose** YHRM\_C\_EMPLOYEE\_APP **as** Employee**;**  **}** |

NOTE:

* **expose YHRM\_C\_EMPLOYEE\_APP as Employee:** Exposes the ‘YHRM\_C\_EMPLOYEE\_APP’ CDS view as ‘Employee’ in the service, making it accessible for consumption.
* To create a service definition, locate the CDS view you wish to expose.

1. Right-click on the selected CDS view.
2. Choose the ‘New Service Definition’ option.

* This action will automatically generate a basic code template for your service definition.

**YHRM\_SB\_M\_EMPLOYEE\_APP** (Business Services / Service Binding)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**YHRM\_SB\_M\_EMPLOYEE\_APP2** (Business Services / Service Binding)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

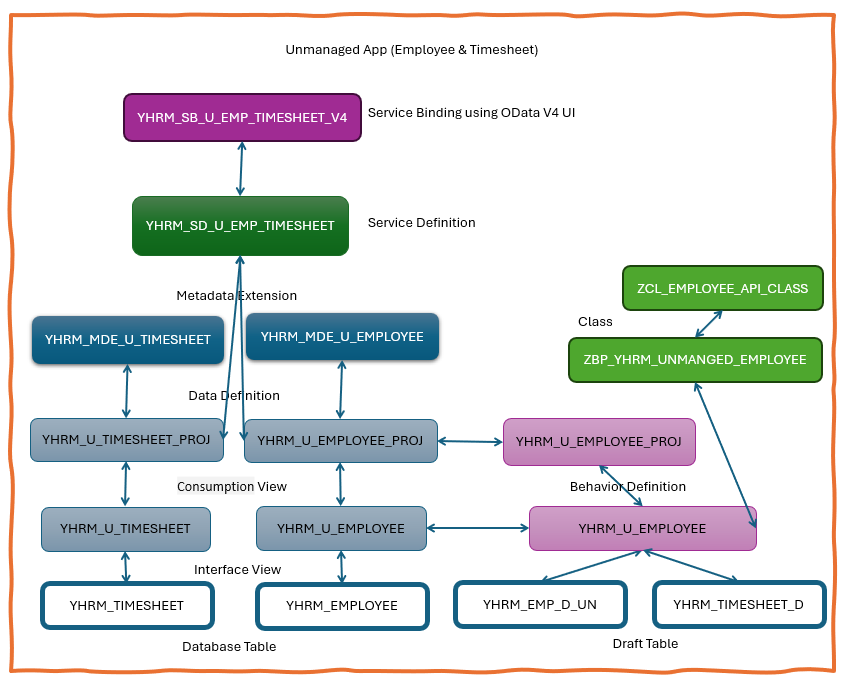
Description automatically generated

NOTE:

* In a managed scenario with OData V2 UI, the ‘Create’ operation is not supported due to UI limitations. However, ‘Edit’ and ‘Delete’ operations are functional.
* In OData V4 UI, the ‘Create’ operation works for both managed and unmanaged scenarios.
* To create a service binding, first locate the service definition you wish to bind.

1. Right-click on the selected service definition.
2. Choose the ‘New Service Binding’ option.
3. Then, select the appropriate binding type for your service.

**Unmanaged Example (Employee & Timesheet)**



**YHRM\_EMPLOYEE** (Dictionary / Database Tables) (see at Employee (YHRM\_EMPOYEE)

**YHRM\_EMP\_D\_UN** (Dictionary / Database Tables) (see at (YHEM\_EMP\_D\_UN (Used in unmanaged app)

))

**YHRM\_TIMESHEET** (Dictionary / Database Tables) (see at Employee (YHRM\_EMPOYEE)

**YHRM\_TIMESHEET\_D** (Dictionary / Database Tables) (see at Employee (YHRM\_EMPOYEE)

**YHRM\_U\_EMPLOYEE** (CDS / Data Definition) Interface View

|  |
| --- |
| @AbapCatalog.viewEnhancementCategory: [#NONE]  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Employee Master Data'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #X,  sizeCategory: #S,  dataClass: #MIXED  }  @Metadata.allowExtensions: true  **define** **root** **view** **entity** yhrm\_U\_EMPLOYEE **as** **select** **from** yhrm\_employee **as** Employee  **composition[**0**..\*]** **of** YHRM\_U\_TIMESHEET **as** \_Timesheet  **association[**1**]** **to** YHRM\_U\_ADDRESS **as** \_Address **on**  **$projection.**AddressId **=** \_Address**.**AddressId  **association[**1**]** **to** YHRM\_U\_JOB **as** \_Job **on**  **$projection.**JobId **=** \_Job**.**JobId  **association[**1**]** **to** yhrm\_employee **as** \_Supervisor **on**  **$projection.**SupervisorId **=** \_Supervisor**.**emp\_id  **association[**1**]** **to** yhrm\_department **as** \_Department **on**  **$projection.**DepartmentId **=** \_Department**.**department\_id  **association[**0**..\*]** **to** YHRM\_F4\_GENDER **as** \_Gender **on** **$projection.**Gender **=** \_Gender**.**Value  **{**  @ObjectModel.text.element: [ 'FirstName' ]  **key** emp\_id **as** EmpId**,**  // concat(concat(first\_name, ' '), last\_name) as EmployeeName,  first\_name **as** FirstName**,**  last\_name **as** LastName**,**  email **as** Email**,**  phone\_no **as** PhoneNo**,**  dob **as** Dob**,**  @ObjectModel.text.element: ['\_Gender.Description']  gender **as** Gender**,**  salary **as** Salary**,**  hire\_date **as** HireDate**,**  active **as** Active**,**  resign\_date **as** ResignDate**,**  address\_id **as** AddressId**,**  @ObjectModel.text.element: ['JobTitle']  @Search.defaultSearchElement: true  job\_id **as** JobId**,**  // \_Job.JobTitle as JobTitle,  @ObjectModel.text.element: ['DepartmentName']  @Search.defaultSearchElement: true  department\_id **as** DepartmentId**,**  // \_Department.department\_name as DepartmentName,  @ObjectModel.text.element: [ 'SupervisorName' ]  supervisor\_id **as** SupervisorId**,**  // concat(concat(\_Supervisor.first\_name, ' '), \_Supervisor.last\_name) as SupervisorName,  @Semantics.user.createdBy: true  created\_by**,**  @Semantics.systemDateTime.createdAt: true  created\_at**,**  @Semantics.user.lastChangedBy: true  last\_changed\_by**,**  @Semantics.user.localInstanceLastChangedBy: true  local\_last\_changed\_by**,**  @Semantics.systemDateTime.localInstanceLastChangedAt: true  local\_last\_changed\_at **as** LocalLastChangedAt**,**  @Semantics.systemDateTime.lastChangedAt: true  last\_changed\_at **as** LastChangedAt**,**  \_Job**.**JobTitle **as** JobTitle**,**  \_Department**.**department\_name **as** DepartmentName**,**  \_Supervisor**.**first\_name **as** SupervisorName**,**  // \_Gender.Description as GenderDescription,  \_Address**,**  \_Department**,**  \_Job**,**  \_Supervisor**,**  \_Timesheet**,**  \_Gender  **}** |

**YHRM\_U\_TIMESHEET** (CDS / Data Definition) Interface View

|  |
| --- |
| @AbapCatalog.viewEnhancementCategory: [#NONE]  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'Timesheet Master Data'  @Metadata.ignorePropagatedAnnotations: true  @ObjectModel.usageType:{  serviceQuality: #X,  sizeCategory: #S,  dataClass: #MIXED  }  @Metadata.allowExtensions: true  **define** **view** **entity** YHRM\_U\_TIMESHEET **as** **select** **from** yhrm\_timesheet  **association** **to** **parent** yhrm\_U\_EMPLOYEE **as** \_Employee **on**  **$projection.**Empid **=** \_Employee**.**EmpId  **association[**0**..\*]** **to** YHRM\_F4\_LEAVETYPE **as** \_LeaveType **on** **$projection.**Leavetype **=** \_LeaveType**.**Value  **{**  @ObjectModel.text.element: ['\_Employee.FirstName']  **key** empid **as** Empid**,**  **key** ydate **as** Ydate**,**  available **as** Available**,**  workinghours **as** Workinghours**,**  @ObjectModel.text.element: ['\_Leavetype.Description']  leavetype **as** Leavetype**,**  overtime\_hrs **as** OvertimeHrs**,**  approved\_by **as** ApprovedBy**,**  // concat(concat(\_Employee.FirstName, ' '), \_Employee.LastName) as EmployeeName,  // \_Employee.FirstName as FirstName,  \_Employee**,**  @Semantics.user.createdBy: true  created\_by**,**  @Semantics.systemDateTime.createdAt: true  created\_at**,**  @Semantics.user.lastChangedBy: true  last\_changed\_by**,**  @Semantics.user.localInstanceLastChangedBy: true  local\_last\_changed\_by**,**  @Semantics.systemDateTime.localInstanceLastChangedAt: true  local\_last\_changed\_at **as** LocalLastChangedAt**,**  @Semantics.systemDateTime.lastChangedAt: true  last\_changed\_at **as** LastChangedAt**,**  \_LeaveType  **}** |

NOTE:

* **Root:** The root keyword is used to define the main entity of the view. It’s the starting point of the view and all other entities are connected to it. In this code, yhrm\_U\_EMPLOYEE is defined as the root entity.
* **Composition:** The composition keyword is used to define a composition relationship between two entities. It implies that the child entities are part of the parent entity and cannot exist without it. In this code, YHRM\_U\_TIMESHEET is a composition of yhrm\_U\_EMPLOYEE, meaning each employee can have multiple timesheets, but a timesheet cannot exist without an associated employee.
* **Association:** The association keyword is used to define a relationship between two entities. It doesn’t imply ownership like composition. In this code, there are associations defined between yhrm\_U\_EMPLOYEE and YHRM\_U\_ADDRESS, YHRM\_U\_JOB, yhrm\_employee (for supervisor), yhrm\_department, and YHRM\_F4\_GENDER.
* **Association to Parent:** It refers to an association from a child entity back to its parent entity. In this code, $projection.Empid = \_Employee.EmpId is an example of an association to parent, where \_Employee is associated back to yhrm\_U\_EMPLOYEE via emp\_id.

**YHRM\_U\_EMPLOYEE\_PROJ** (CDS / Data Definition) Consumption View

|  |
| --- |
| @EndUserText.label: 'Projection view of Employee CDS'  @AccessControl.authorizationCheck: #NOT\_REQUIRED  **define** **root** **view** **entity** YHRM\_U\_EMPLOYEE\_PROJ **provider** **contract** **transactional\_query** **as** **projection** **on** yhrm\_U\_EMPLOYEE  **{**  @ObjectModel.text.element: [ 'FirstName' ]  **key** EmpId**,**  FirstName**,**  LastName**,**  Email**,**  PhoneNo**,**  Dob**,**  Gender**,**  Salary**,**  HireDate**,**  Active**,**  ResignDate**,**  AddressId**,**  JobId**,**  DepartmentId**,**  SupervisorId**,**  created\_by**,**  created\_at**,**  last\_changed\_by**,**  local\_last\_changed\_by**,**  LocalLastChangedAt**,**  LastChangedAt**,**  @UI.hidden: true  DepartmentName**,**  @UI.hidden: true  SupervisorName**,**  @UI.hidden: true  JobTitle**,**  /\* Associations \*/  \_Address**,**  \_Department**,**  \_Job**,**  \_Supervisor**,**  \_Timesheet **:** **redirected** **to** **composition** **child** YHRM\_U\_TIMESHEET\_PROJ  **}** |

**YHRM\_U\_TIMESHEET\_PROJ** (CDS / Data Definition) Consumption View

|  |
| --- |
| @EndUserText.label: 'Projection view of Timesheet CDS'  @AccessControl.authorizationCheck: #NOT\_REQUIRED  **define** **view** **entity** YHRM\_U\_TIMESHEET\_PROJ **as** **projection** **on** YHRM\_U\_TIMESHEET  **{**  **key** Empid**,**  **key** Ydate**,**  Available**,**  Workinghours**,**  Leavetype**,**  OvertimeHrs**,**  ApprovedBy**,**  created\_by**,**  created\_at**,**  last\_changed\_by**,**  local\_last\_changed\_by**,**  LocalLastChangedAt**,**  LastChangedAt**,**  /\* Associations \*/  \_Employee **:** **redirected** **to** **parent** YHRM\_U\_EMPLOYEE\_PROJ**,**  \_LeaveType  **}** |

NOTE:

* **Provider Contract Transactional Query:** The provider contract transactional\_query keyword is used to define the type of the view. It indicates that the view is a projection view and it supports transactional operations. This means that the view can be used for read, insert, update, and delete operations.
* **Projection On:** The projection on keyword is used to define the base entity of the projection view. It indicates that the view is a projection of the specified entity. In your code, YHRM\_U\_EMPLOYEE\_PROJ is a projection of yhrm\_U\_EMPLOYEE and YHRM\_U\_TIMESHEET\_PROJ is a projection of YHRM\_U\_TIMESHEET.
* **Redirected to Composition Child:** The redirected to composition child keyword is used to redirect the composition association to a child projection view. It indicates that the composition association \_Timesheet in YHRM\_U\_EMPLOYEE\_PROJ is redirected to the child projection view YHRM\_U\_TIMESHEET\_PROJ.
* **Redirected to Parent:** The redirected to parent keyword is used to redirect the association to a parent projection view. It indicates that the association \_Employee in YHRM\_U\_TIMESHEET\_PROJ is redirected to the parent projection view YHRM\_U\_EMPLOYEE\_PROJ.

**YHRM\_MDE\_U\_EMPLOYEE** (CDS / Metadata Extension)

|  |
| --- |
| @Metadata.layer: #CORE  @UI.headerInfo: {  typeName: 'Employee',  typeNamePlural: 'Employees',  title: { value: 'EmpId'},  description: { value: 'FirstName' }  ,  typeImageUrl: 'sap-icon://employee'  }  @UI.presentationVariant: [{  sortOrder: [ { by: 'EmpId', direction: #DESC } ],  visualizations: [{ type: #AS\_LINEITEM}]  }]  @Search.searchable: true  **annotate** **entity** yhrm\_U\_EMPLOYEE  **with**  **{**  @UI.facet: [  {  id: 'Employee',  purpose: #HEADER,  type: #DATAPOINT\_REFERENCE,  position: 10,  targetQualifier: 'Emp'  }  ,{  id: 'Timesheet',  purpose: #STANDARD,  position: 30,  label: 'Timesheet',  type: #LINEITEM\_REFERENCE,  targetElement: '\_Timesheet'    }  ,  {  id: 'EmployeeEmail',  purpose: #HEADER,  type: #DATAPOINT\_REFERENCE,  position: 11,  targetQualifier: 'Email'  }  ,  {  id: 'EmployeePhoneNo',  purpose: #HEADER,  type: #DATAPOINT\_REFERENCE,  position: 20,  targetQualifier: 'PhoneNo'  },  {  id: 'Employee',  type: #IDENTIFICATION\_REFERENCE,  purpose: #STANDARD,  label: 'Employee Professional Info',  position: 10  },  {  id: 'EmployeePersonaldata',  type: #FIELDGROUP\_REFERENCE,  purpose: #STANDARD,  label: 'Employee Personal Info',  position: 20,  targetQualifier: 'Info'  // ,  // hidden: true  }  ]    @UI: { lineItem: [  { position: 10 ,label: 'Employee Id' , cssDefault.width: '10rem'} ,  { type: #FOR\_ACTION , dataAction: 'updateEmployeeStatus', label: 'Change Employee Status', position: 10 }  ]}    @UI.selectionField: [{ position: 10}]  @UI.identification: [{ position: 1 , label: 'Employee Id'}]  @Search.defaultSearchElement: true  @EndUserText.label: 'Employee Id'  @Consumption.valueHelpDefinition: [{ entity: { element: 'EmpId' , name: 'YHRM\_F4\_EMPLOYEE'} }]  EmpId**;**  // @UI.hidden: true  // @EndUserText.label: 'EmployeeName'  // employeename;  @UI : { lineItem: [ { position: 20 , label: 'First Name' , cssDefault.width: '6rem'} ] }  @UI.fieldGroup: [{ position: 1, qualifier: 'Info', label: 'First Name' }]  @UI.dataPoint: { qualifier: 'Emp' , title: 'FirstName' }  FirstName**;**  @UI : { lineItem: [ { position: 21 , label: 'Last Name' , cssDefault.width: '6rem'} ] }  @UI.fieldGroup: [{ position: 2, qualifier: 'Info', label: 'Last Name' }]  LastName**;**  @UI : { selectionField: [{ position: 30}] ,  lineItem: [ { position: 30 , label: 'Department Id'} ],  identification: [{ position: 10 , label: 'Department Id'}]  }  @Consumption.valueHelpDefinition: [{ entity: { element: 'DepartmentId' , name: 'YHRM\_F4\_DEPARTMENT'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Department Id'  DepartmentId**;**  @UI : { selectionField: [{ position: 40}] ,  lineItem: [ { position: 40 , label: 'Job Id'} ],  identification: [{ position: 20 , label: 'Job Id'}] }  @Consumption.valueHelpDefinition: [{ entity: { element: 'JobId' , name: 'YHRM\_F4\_JOB'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Job Id'  JobId**;**  @UI : { identification: [{ position: 50 , label: 'Supervisor Id'}] ,  lineItem: [ { position: 50 , label: 'Supervisor Id'} ]}  @Consumption.valueHelpDefinition: [{ entity: { element: 'EmpId' , name: 'YHRM\_F4\_EMPLOYEE'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Supervisor Id'  SupervisorId**;**  @UI.lineItem: [{ position: 22 , label: 'Email', cssDefault.width: '12rem'}]  @UI.dataPoint: { qualifier: 'Email' , title: 'Email' }  @UI.fieldGroup: [{ position: 4, qualifier: 'Info' }]  Email**;**  @UI.lineItem: [{ position: 23 , label: 'Phone Number', cssDefault.width: '8rem'}]  @UI.dataPoint: { qualifier: 'PhoneNo' , title: 'Phone Number'}  @UI.fieldGroup: [{ position: 5, qualifier: 'Info' , label: 'Phone Number'}]  PhoneNo**;**  @UI.lineItem: [{ position: 24 , cssDefault.width: '8rem'}]  @UI.fieldGroup: [{ position: 10, qualifier: 'Info' }]  Dob**;**  @UI.lineItem: [{ position: 25 ,label: 'Gender', cssDefault.width: '4rem'}]  @UI.fieldGroup: [{ position: 20, qualifier: 'Info' ,label: 'Gender'}]  @Consumption.valueHelpDefinition: [{ entity: { element: 'Value' , name: 'YHRM\_F4\_GENDER'} }]  Gender**;**  @UI : {  // selectionField: [{ position: 50}] ,  lineItem: [ { position: 29 , label: 'Address Id', cssDefault.width: '5rem'} ],  identification: [{ position: 30, label: 'Address Id' }] }  @Consumption.valueHelpDefinition: [{ entity: { element: 'AddressId' , name: 'YHRM\_F4\_ADDRESS'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Address Id'  AddressId**;**  @UI.lineItem: [{ position: 27 , label: 'Hire Date', cssDefault.width: '8rem'}]  @UI.fieldGroup: [{ position: 30, qualifier: 'Info' , label: 'Hire Date' }]  HireDate**;**  @UI.lineItem: [{ position: 28 ,label: 'Active', cssDefault.width: '5rem'}]  @UI.fieldGroup: [{ position: 40, qualifier: 'Info' ,label: 'Active'}]  Active**;**  @UI.hidden: true  created\_by**;**  @UI.hidden: true  created\_at**;**  @UI.hidden: true  last\_changed\_by**;**  @UI.hidden: true  local\_last\_changed\_by**;**  @UI.hidden: true  local\_last\_changed\_at**;**  @UI.hidden: true  last\_changed\_at**;**  /\* Associations \*/  @UI.hidden: true  \_Address**;**  @UI.hidden: true  \_Department**;**  @UI.hidden: true  \_Job**;**  @UI.hidden: true  \_Supervisor**;**  @UI.hidden: true  \_Timesheet**;**    **}** |

**YHRM\_MDE\_U\_TIMESHEET** (CDS / Metadata Extension)

|  |
| --- |
| @Metadata.layer: #CORE  @UI.headerInfo: {  typeName: 'Timesheet',  typeNamePlural: 'Timesheets',  title: { value: 'Empid'},  description: { value: '\_Employee.FirstName' }  ,  typeImageUrl: 'sap-icon://timesheet'  }  @UI.presentationVariant: [{  sortOrder: [ { by: 'Ydate', direction: #DESC } ],  visualizations: [{ type: #AS\_LINEITEM}]  }]  **annotate** **entity** YHRM\_U\_TIMESHEET  **with**  **{**  @UI.facet: [{  id: 'Date',  purpose: #HEADER,  type: #DATAPOINT\_REFERENCE,  position: 10,  targetQualifier: 'Date'  }  ,  {  id: 'Available',  purpose: #HEADER,  type: #DATAPOINT\_REFERENCE,  position: 20,  targetQualifier: 'Available'  },  {  id: 'EmployeeInfo',  type: #COLLECTION,  label: 'Employee Info',  position: 10  },  {  id: 'EmployeeTimesheetdata',  type: #FIELDGROUP\_REFERENCE,  purpose: #STANDARD,  label: 'Employee Timesheet data',  parentId: 'EmployeeInfo',  position: 10,  targetQualifier: 'Timesheet'  }  ]    @UI.selectionField: [{ position: 10}]  @UI.lineItem: [{ position: 10 }]  @EndUserText.label: 'Employee Id'  Empid**;**  @UI.selectionField: [{ position: 20}]  @UI.lineItem: [{ position: 20 }]  @UI.dataPoint: { qualifier: 'Date' , title: 'Date'}  Ydate**;**  @UI.lineItem: [{ position: 30 }]  @UI.dataPoint: { qualifier: 'Available' , title: 'Available'}  @UI.fieldGroup: [{ position: 9, qualifier: 'Timesheet'}]  @EndUserText.label: 'Available'  Available**;**  @UI.selectionField: [{ position: 40 }]  @UI.lineItem: [{ position: 40 , label: 'Working Time Hours'}]  @UI.fieldGroup: [{ position: 10, qualifier: 'Timesheet' , label: 'Working Time Hours'}]  Workinghours**;**  @UI.selectionField: [{ position: 50}]  @UI.lineItem: [{ position: 50 }]  @UI.fieldGroup: [{ position: 20, qualifier: 'Timesheet' }]  @Consumption.valueHelpDefinition: [{ entity: { element: 'Value' , name: 'YHRM\_F4\_LEAVETYPE'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Leave Type'  Leavetype**;**  @UI.lineItem: [{ position: 60 , label: 'Over Time Hours'}]  @UI.fieldGroup: [{ position: 30, qualifier: 'Timesheet' , label: 'Over Time Hours'}]  OvertimeHrs**;**  @UI.lineItem: [{ position: 70 , label: 'Approved By' }]  @UI.fieldGroup: [{ position: 40, qualifier: 'Timesheet' , label: 'Approved By'}]  @Consumption.valueHelpDefinition: [{ entity: { element: 'EmpId' , name: 'YHRM\_F4\_EMPLOYEE'} }]  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold: 0.8  @EndUserText.label: 'Department Id'  ApprovedBy**;**  /\* Associations \*/  // \_Employee;    **}** |

NOTE:

* **@UI.headerInfo:** Defines the header information for the entity.
* **@UI.presentationVariant:** Specifies the sort order and visualization type for the entity.
* **@UI.facet:** Outlines the structure and layout of the entity’s user interface.
* **@UI.selectionField:** Determines the fields available for selection in the entity.
* **@UI.lineItem:** Defines the line item ‘Employee Id’ in the ‘Employee’ entity.
* **@UI.identification:** Specifies the identification field ‘Employee Id’ for the ‘Employee’ entity.
* **@Search.defaultSearchElement:** Sets ‘Employee Id’ as the default search element in the ‘Employee’ entity.
* **@EndUserText.label:** Provides a user-friendly label ‘Employee Id’ for the ‘Employee’ entity.
* **@Consumption.valueHelpDefinition:** Defines the value help for the ‘EmpId’ element in the ‘Employee’ entity.
* **@UI.fieldGroup:** Groups related fields under the label ‘First Name’ in the ‘Employee’ entity.
* **@UI.dataPoint:** Defines a data point ‘FirstName’ for the ‘Employee’ entity.
* **cssDefault.width: 5rem:** Sets the default width of the UI element to 5rem, controlling its size for better layout and readability.
* **position: 40:** Determines the position of the UI element in the layout, helping to organize the interface.
* **qualifier:** **'Info':** Provides a unique identifier ‘Info’ for the UI element, aiding in its referencing and customization.
* **type: #DATAPOINT\_REFERENCE:** Refers to a specific data point in the ‘Employee’ entity, allowing for targeted data manipulation and display.
* **type: #IDENTIFICATION\_REFERENCE:** Identifies a specific element in the ‘Employee’ entity, aiding in entity navigation and data retrieval.
* **type: #FIELDGROUP\_REFERENCE:** Groups related fields together in the ‘Employee’ entity, enhancing data organization and user interface structure.
* **TargetElement:** The targetElement keyword is used in the context of UI annotations in SAP Fiori. It is used to specify the target element of a UI annotation. In your code, targetElement: '\_Timesheet' means that the UI facet with id ‘Timesheet’ is targeting the ‘\_Timesheet’ association in the ‘yhrm\_U\_EMPLOYEE’ entity. This allows the UI to display data from the ‘\_Timesheet’ association when the ‘Timesheet’ facet is selected.

**YHRM\_U\_EMPLOYEE** (CDS / Behavior Definition)

|  |
| --- |
| **unmanaged** **implementation** **in** **class** zbp\_yhrm\_unmanged\_employee **unique;**  **strict** **(** 2 **);**  **with** **draft;**  **define** **behavior** **for** yhrm\_U\_EMPLOYEE **alias** Employee  **draft** **table** yhrm\_emp\_d\_un  //late numbering  **early** **numbering**  **lock** **master**  **total** **etag** LocalLastChangedAt  **authorization** **master** **(** **instance** **)**  **etag** **master** LocalLastChangedAt  **{**  **create;**  **update** **(features** **:** **instance);**  **delete** **(features** **:** **instance);**  **association** \_Timesheet **{** **create** **(features** **:** **instance);** **with** **draft;}**  **draft** **action** Edit**;**  **draft** **action** Activate **optimized;**  **draft** **action** Discard**;**  **draft** **action** Resume**;**  **field** **(readonly)** EmpId**;**  **field** **(** **mandatory** **:** **create** **)** FirstName**,** Email**,** Gender**,** Dob**;**  **validation** validate\_fields **on** **save** **{create;** **update;}**  **determination** updateemployeename **on** **modify** **{** **field** Gender**;** **}**  **side** **effects** **{** **field** Gender **affects** **field** FirstName **;}**  **draft** **determine** **action** Prepare**{**  **validation** validate\_fields**;**  **}**  **field** **(** **readonly** **)** LastChangedAt**,** last\_changed\_by**,** LocalLastChangedAt**,** local\_last\_changed\_by**,** created\_at**,** created\_by**;**  **action** updateEmployeeStatus **parameter** YHRM\_EMP\_ACTIVE **result** **[1]** **$self;**  **mapping** **for** yhrm\_employee **control** yhrm\_emp\_u\_structure**{**  Active **=** active**;**  AddressId **=** address\_id**;**  DepartmentId **=** department\_id**;**  Dob **=** dob**;**  Email **=** email**;**  EmpId **=** emp\_id**;**  FirstName **=** first\_name**;**  Gender **=** gender**;**  HireDate **=** hire\_date**;**  JobId **=** job\_id**;**  LastName **=** last\_name**;**  PhoneNo **=** phone\_no**;**  ResignDate **=** resign\_date**;**  Salary **=** salary**;**  SupervisorId **=** supervisor\_id**;**  LastChangedAt **=** last\_changed\_at**;**  LocalLastChangedAt **=** local\_last\_changed\_at**;**  **}**  **}**  **define** **behavior** **for** YHRM\_U\_TIMESHEET **alias** Timesheet  **draft** **table** yhrm\_timesheet\_d  //late numbering  **early** **numbering**  **lock** **dependent** **by** \_Employee  **authorization** **dependent** **by** \_Employee  **etag** **master** LocalLastChangedAt  **{**  **update;**  **delete;**  **field** **(** **readonly** **)** Empid**;**  **association** \_Employee **{with** **draft;}**  **field** **(mandatory** **:** **create,** **readonly** **:** **update)** Ydate**;**  **field** **(** **readonly** **)** created\_at**,**created\_by**,** LastChangedAt**,**last\_changed\_by**,**LocalLastChangedAt**,**local\_last\_changed\_by**;**  **determination** updateHours **on** **modify** **{** **field** Available**;** **}**  **side** **effects** **{** **field** Available **affects** **field** Leavetype **;}**  **mapping** **for** yhrm\_timesheet **corresponding{**  ApprovedBy **=** approved\_by**;**  Available **=** available**;**  Empid **=** empid**;**  Leavetype **=** leavetype**;**  OvertimeHrs **=** overtime\_hrs**;**  Ydate **=** ydate**;**  Workinghours **=** workinghours**;**  LastChangedAt **=** last\_changed\_at**;**  **}**  **}** |

NOTE:

* **With Draft:** The with draft keyword is used to indicate that the behavior definition supports draft handling. This means that changes to the data can be saved as a draft before being officially saved.
* **Early Numbering:** The early numbering keyword is used to indicate that keys for new instances are determined as soon as the instance is created, not when it is saved.
* **Total Etag:** The total etag keyword is used to define the field that is used for concurrency control. The field defined after this keyword is used to check if the data has been changed by another user.
* **Etag Master:** The etag master keyword is used to specify the field that is used to calculate the ETag for concurrency control.
* **Lock Master:** The lock master keyword is used to indicate that the entity defined in the behavior definition is the leading entity in a lock object.
* **Validation:** The validation keyword is used to define a validation rule that is checked when the specified operations (create, update, etc.) are performed.
* **Determination:** The determination keyword is used to define a determination rule that is executed when the specified operations (modify, etc.) are performed.
* **Action:** The action keyword is used to define an action that can be performed on the entity.
* **Side Effects:** The side effects keyword is used to define a side effect rule that specifies how changes to one field affect other fields.
* **Draft Determine Action Prepare:** The draft determine action prepare keyword is used to define a determination rule that is executed when the ‘Prepare’ draft action is performed.

**YHRM\_U\_EMPLOYEE\_PROJ** (CDS / Behavior Definition)

|  |
| --- |
| **projection;**  **strict** **(** 2 **);**  **use** **draft;**  **use** **side** **effects;**  **define** **behavior** **for** YHRM\_U\_EMPLOYEE\_PROJ **alias** Employee  **use** **etag**  **{**  **use** **create;**  **use** **update;**  **use** **delete;**  **use** **action** Edit**;**  **use** **action** Activate**;**  **use** **action** Discard**;**  **use** **action** Resume**;**  **use** **action** Prepare**;**  **use** **action** updateEmployeeStatus**;**  **use** **association** \_Timesheet **{** **create;** **with** **draft;** **}**  **}**  **define** **behavior** **for** YHRM\_U\_TIMESHEET\_PROJ **alias** Timesheet  **{**  **use** **update;**  **use** **delete;**  **use** **association** \_Employee **{** **with** **draft;** **}**  **}** |

**ZBP\_YHRM\_UNMANAGED\_EMPLOYEE** (Source Code Library / Classes)

|  |
| --- |
| \* lhc\_employee stands for Local Handler Class for Employee Entity  CLASS lhc\_employee DEFINITION INHERITING FROM cl\_abap\_behavior\_handler.  PRIVATE SECTION.  METHODS get\_instance\_features FOR INSTANCE FEATURES  IMPORTING keys REQUEST requested\_features FOR employee RESULT result.  METHODS get\_instance\_authorizations FOR INSTANCE AUTHORIZATION  IMPORTING keys REQUEST requested\_authorizations FOR employee RESULT result.  METHODS create FOR MODIFY  IMPORTING entities FOR CREATE employee.  METHODS earlynumbering\_create FOR NUMBERING  IMPORTING entities FOR CREATE employee.  METHODS update FOR MODIFY  IMPORTING entities FOR UPDATE employee.  METHODS delete FOR MODIFY  IMPORTING keys FOR DELETE employee.  METHODS read FOR READ  IMPORTING keys FOR READ employee RESULT result.  METHODS lock FOR LOCK  IMPORTING keys FOR LOCK employee.  \* rba stands for Read by association  METHODS rba\_timesheet FOR READ  IMPORTING keys\_rba FOR READ employee\\_timesheet FULL result\_requested RESULT result LINK association\_links.  \* cba stands for Create by association  METHODS cba\_timesheet FOR MODIFY  IMPORTING entities\_cba FOR CREATE employee\\_timesheet.  METHODS validate\_fields FOR VALIDATE ON SAVE  IMPORTING keys FOR employee~validate\_fields.  METHODS updateemployeename FOR DETERMINE ON MODIFY  IMPORTING keys FOR employee~updateemployeename.  METHODS updateemployeestatus FOR MODIFY  IMPORTING keys FOR ACTION employee~updateemployeestatus RESULT result.  METHODS earlynumbering\_cba\_timesheet FOR NUMBERING  IMPORTING entities FOR CREATE employee\\_timesheet.  ENDCLASS.  CLASS lhc\_employee IMPLEMENTATION.  METHOD get\_instance\_features.  ENDMETHOD.  METHOD get\_instance\_authorizations.  ENDMETHOD.  METHOD create.  zcl\_employee\_api\_class=>get\_instance( )->create\_employee(  EXPORTING  entities = entities  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD.  METHOD earlynumbering\_create.  zcl\_employee\_api\_class=>get\_instance( )->earlynumbering\_create\_employee(  EXPORTING  entities = entities  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD.  METHOD update.  zcl\_employee\_api\_class=>get\_instance( )->update\_employee(  EXPORTING  entities = entities  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD.  METHOD delete.  zcl\_employee\_api\_class=>get\_instance( )->delete\_employee(  EXPORTING  keys = keys  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD.  METHOD read.  zcl\_employee\_api\_class=>get\_instance( )->read\_employee(  EXPORTING  keys = keys  CHANGING  result = result  failed = failed  reported = reported  ).  ENDMETHOD.  METHOD lock.  Try.  DATA(lock) = cl\_abap\_lock\_object\_factory=>get\_instance( iv\_name = 'EYHRM\_U\_LOCKEMP' ).  catch cx\_abap\_lock\_failure into DATA(exception).  RAISE SHORTDUMP exception.  ENDTRY.  LOOP AT keys ASSIGNING FIELD-SYMBOL(<lfs\_employee>).  try.  lock->enqueue(  it\_parameter = VALUE #( ( name = 'EmpId' value = ref #( <lfs\_employee>-EmpId ) ) )  ).  catch cx\_abap\_foreign\_lock into DATA(foreign\_lock).  APPEND VALUE #(  EmpId = keys[ 1 ]-EmpId  %msg = new\_message\_With\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Record is locked by : ' && foreign\_lock->user\_name  )  ) TO reported-employee.  APPEND VALUE #(  EmpId = keys[ 1 ]-EmpId  ) TO failed-employee.  catch cx\_abap\_lock\_failure into exception.  RAISE SHORTDUMP exception.  ENDTRY.  ENDLOOP.  ENDMETHOD.  METHOD rba\_timesheet.  ENDMETHOD.  METHOD cba\_timesheet.  zcl\_employee\_api\_class=>get\_instance( )->cba\_timesheet(  EXPORTING  entities\_cba = entities\_cba  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD.  METHOD earlynumbering\_cba\_timesheet.  zcl\_employee\_api\_class=>get\_instance( )->earlynumbering\_cba\_timesheet(  EXPORTING  entities = entities  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD.  METHOD validate\_fields.  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY employee  ALL FIELDS WITH CORRESPONDING #( keys )  RESULT DATA(lt\_employee\_tmp)  REPORTED DATA(lt\_reported)  FAILED DATA(lt\_failed).  IF NOT lt\_employee\_tmp[] is INITIAL.  READ TABLE lt\_employee\_tmp ASSIGNING FIELD-SYMBOL(<lfs\_employee\_tmp>) index 1.  if <lfs\_employee\_tmp> is ASSIGNED.  reported-employee = VALUE #(  ( %tky = <lfs\_employee\_tmp>-%tky %state\_area = 'VALIDATE\_FNM' )  ( %tky = <lfs\_employee\_tmp>-%tky %state\_area = 'VALIDATE\_GENDER' )  ( %tky = <lfs\_employee\_tmp>-%tky %state\_area = 'VALIDATE\_DOB' )  ( %tky = <lfs\_employee\_tmp>-%tky %state\_area = 'VALIDATE\_EMAIL' )  ).  if <lfs\_employee\_tmp>-FirstName is INITIAL or  <lfs\_employee\_tmp>-Email is INITIAL or  <lfs\_employee\_tmp>-Dob is INITIAL or  <lfs\_employee\_tmp>-Gender is INITIAL.  failed-employee = VALUE #( ( %tky = <lfs\_employee\_tmp>-%tky ) ).  IF <lfs\_employee\_tmp>-FirstName is INITIAL.  reported-employee = VALUE #( (  %tky = <lfs\_employee\_tmp>-%tky  %state\_area = 'VALIDATE\_FNM'  %element-firstname = if\_abap\_behv=>mk-on  %msg = new\_message(  id = 'SY'  number = '002'  severity = if\_abap\_behv\_message=>severity-error  v1 = 'FirstName is Required!'  )  ) ).  ENDIF.  IF <lfs\_employee\_tmp>-Gender is INITIAL.  reported-employee = VALUE #( BASE reported-employee (  %tky = <lfs\_employee\_tmp>-%tky  %state\_area = 'VALIDATE\_GENDER'  %element-gender = if\_abap\_behv=>mk-on  %msg = new\_message(  id = 'SY'  number = '002'  severity = if\_abap\_behv\_message=>severity-error  v1 = 'Gender is Required!'  )  ) ).  ENDIF.  IF <lfs\_employee\_tmp>-Dob is INITIAL.  reported-employee = VALUE #( BASE reported-employee (  %tky = <lfs\_employee\_tmp>-%tky  %state\_area = 'VALIDATE\_DOB'  %element-dob = if\_abap\_behv=>mk-on  %msg = new\_message(  id = 'SY'  number = '002'  severity = if\_abap\_behv\_message=>severity-error  v1 = 'Date of Birth is Required!'  )  ) ).  ENDIF.  IF <lfs\_employee\_tmp>-Email is INITIAL.  reported-employee = VALUE #( BASE reported-employee (  %tky = <lfs\_employee\_tmp>-%tky  %state\_area = 'VALIDATE\_EMAIL'  %element-email = if\_abap\_behv=>mk-on  %msg = new\_message(  id = 'SY'  number = '002'  severity = if\_abap\_behv\_message=>severity-error  v1 = 'Email is Required!'  )  ) ).  ENDIF.  ENDIF.  ENDIF.  ENDIF.  ENDMETHOD.  METHOD updateemployeename.  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY employee  FIELDS ( Gender ) WITH CORRESPONDING #( keys )  RESULT DATA(lt\_employee).  LOOP at lt\_employee ASSIGNING FIELD-SYMBOL(<lfs\_employee>).  DATA: lv\_firstname TYPE string.  lv\_firstname = <lfs\_employee>-FirstName.  IF lv\_firstname CP 'Mr.\*' OR lv\_firstname CP 'Mrs.\*'.  SPLIT lv\_firstname AT ' ' INTO TABLE DATA(lt\_name\_parts).  DELETE lt\_name\_parts INDEX 1.  CONCATENATE LINES OF lt\_name\_parts INTO lv\_firstname SEPARATED BY SPACE.  ENDIF.  IF <lfs\_employee>-Gender EQ 'M'.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Employee  UPDATE FIELDS ( FirstName )  WITH VALUE #( (  %tky = <lfs\_employee>-%tky  FirstName = |Mr. { lv\_firstname }|  ) ).  ELSEIF <lfs\_employee>-Gender = 'F'.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Employee  UPDATE FIELDS ( FirstName )  WITH VALUE #( (  %tky = <lfs\_employee>-%tky  FirstName = |Mrs. { lv\_firstname }|  ) ).  ENDIF.  ENDLOOP.  ENDMETHOD.  METHOD updateEmployeeStatus.  DATA(lt\_keys) = keys.  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Employee  FIELDS ( Active ) WITH CORRESPONDING #( keys )  RESULT DATA(lt\_employee\_status).  DATA(lv\_new\_status) = lt\_keys[ 1 ]-%param-active.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  in LOCAL MODE  ENTITY Employee  UPDATE FIELDS ( Active )  WITH VALUE #( (  %tky = lt\_employee\_status[ 1 ]-%tky Active = lv\_new\_status  ) ).  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Employee  ALL FIELDS WITH CORRESPONDING #( keys )  RESULT DATA(lt\_employee).  result = VALUE #( FOR <lfs\_employee> in lt\_employee (  %tky = <lfs\_employee>-%tky  %param = <lfs\_employee>  ) ).  ENDMETHOD.  ENDCLASS.  CLASS lhc\_timesheet DEFINITION INHERITING FROM cl\_abap\_behavior\_handler.  PRIVATE SECTION.  METHODS update FOR MODIFY  IMPORTING entities FOR UPDATE timesheet.  METHODS delete FOR MODIFY  IMPORTING keys FOR DELETE timesheet.  METHODS read FOR READ  IMPORTING keys FOR READ timesheet RESULT result.  METHODS rba\_employee FOR READ  IMPORTING keys\_rba FOR READ timesheet\\_employee FULL result\_requested RESULT result LINK association\_links.  METHODS updatehours FOR DETERMINE ON MODIFY  IMPORTING keys FOR timesheet~updatehours.  ENDCLASS.  CLASS lhc\_timesheet IMPLEMENTATION.  METHOD update.  ENDMETHOD.  METHOD delete.  ENDMETHOD.  METHOD read.  ENDMETHOD.  METHOD rba\_employee.  ENDMETHOD.  METHOD updateHours.  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Timesheet  FIELDS ( Available ) WITH CORRESPONDING #( keys )  RESULT DATA(lt\_timesheet).  LOOP at lt\_timesheet ASSIGNING FIELD-SYMBOL(<lfs\_timesheet>).  IF <lfs\_timesheet>-Available EQ 'X'.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Timesheet  UPDATE FIELDS ( Leavetype )  WITH VALUE #( (  %tky = <lfs\_timesheet>-%tky  Leavetype = ''  ) ).  ELSE.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Timesheet  UPDATE FIELDS ( Workinghours OvertimeHrs )  WITH VALUE #( (  %tky = <lfs\_timesheet>-%tky  Workinghours = 0  OvertimeHrs = 0  ) ).  ENDIF.  ENDLOOP.  ENDMETHOD.  ENDCLASS.  \* lsc stands for Local Saver Class  CLASS lsc\_yhrm\_u\_employee DEFINITION INHERITING FROM cl\_abap\_behavior\_saver.  PROTECTED SECTION.  METHODS finalize REDEFINITION.  METHODS check\_before\_save REDEFINITION.  METHODS save REDEFINITION.  METHODS cleanup REDEFINITION.  METHODS cleanup\_finalize REDEFINITION.  ENDCLASS.  CLASS lsc\_yhrm\_u\_employee IMPLEMENTATION.  METHOD finalize.  ENDMETHOD.  METHOD check\_before\_save.  DATA: gt\_employee\_tmp TYPE STANDARD TABLE OF yhrm\_employee,  gt\_timesheet\_tmp TYPE STANDARD TABLE OF yhrm\_timesheet,  lv\_age TYPE i,  lv\_hire\_date TYPE d,  lv\_dob TYPE d,  lv\_phone\_no\_string TYPE string.  gt\_employee\_tmp = zcl\_employee\_api\_class=>get\_instance( )->gt\_employee.  gt\_timesheet\_tmp = zcl\_employee\_api\_class=>get\_instance( )->gt\_timesheet.  IF NOT gt\_employee\_tmp[] IS INITIAL.  READ TABLE gt\_employee\_tmp ASSIGNING FIELD-SYMBOL(<lfs\_employee\_tmp>) INDEX 1.  IF <lfs\_employee\_tmp> IS ASSIGNED.  IF <lfs\_employee\_tmp>-dob IS NOT INITIAL.  lv\_age = trunc( ( sy-datum - <lfs\_employee\_tmp>-dob ) / 365 ).  lv\_hire\_date = <lfs\_employee\_tmp>-dob + 16 \* 365.  ELSEIF <lfs\_employee\_tmp>-hire\_date IS NOT INITIAL.  lv\_age = trunc( ( sy-datum - <lfs\_employee\_tmp>-hire\_date ) / 365 ).  ENDIF.  IF lv\_age < 16.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Employee age should be 16 or more.'  )  ) TO reported-employee.  ENDIF.  IF <lfs\_employee\_tmp>-hire\_date IS NOT INITIAL and <lfs\_employee\_tmp>-hire\_date <= lv\_hire\_date.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Hire date should be after 16 years from DOB.'  )  ) TO reported-employee.  ENDIF.  IF <lfs\_employee\_tmp>-email IS NOT INITIAL AND <lfs\_employee\_tmp>-email CS '@. ' .  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id ) to failed-employee.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Please enter a valid email address.'  )  ) to reported-employee.  ENDIF.  IF <lfs\_employee\_tmp>-phone\_no IS NOT INITIAL.  SHIFT <lfs\_employee\_tmp>-phone\_no LEFT DELETING LEADING '0'.  lv\_phone\_no\_string = <lfs\_employee\_tmp>-phone\_no.  IF strlen( lv\_phone\_no\_string ) <> 10.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id ) to failed-employee.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Phone number should be exactly 10 digits.'  )  ) to reported-employee.  ENDIF.  ENDIF.  ENDIF.  ENDIF.  IF NOT gt\_timesheet\_tmp[] IS INITIAL.  READ TABLE gt\_timesheet\_tmp ASSIGNING FIELD-SYMBOL(<lfs\_timesheet\_tmp>) INDEX 1.  IF <lfs\_timesheet\_tmp> IS ASSIGNED.  IF <lfs\_timesheet\_tmp>-ydate IS NOT INITIAL AND <lfs\_timesheet\_tmp>-ydate <= <lfs\_employee\_tmp>-hire\_date.  APPEND VALUE #( empid = <lfs\_timesheet\_tmp>-empid  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Timesheet date should be after Hiredate.'  )  ) TO reported-employee.  ENDIF.  IF <lfs\_timesheet\_tmp>-available IS NOT INITIAL AND NOT <lfs\_timesheet\_tmp>-leavetype IS INITIAL AND <lfs\_timesheet\_tmp>-overtime\_hrs > 0.  APPEND VALUE #( empid = <lfs\_timesheet\_tmp>-empid  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'If employee not available, Leave type should be mandatory, and overtime hours must be 0.'  )  ) TO reported-employee.  ENDIF.  IF <lfs\_timesheet\_tmp>-workinghours IS NOT INITIAL AND <lfs\_timesheet\_tmp>-overtime\_hrs IS NOT INITIAL.  IF <lfs\_timesheet\_tmp>-workinghours > 8 OR <lfs\_timesheet\_tmp>-overtime\_hrs > 6.  APPEND VALUE #( empid = <lfs\_timesheet\_tmp>-empid  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Working hours should not exceed 8 and overtime hours should not exceed 6.'  )  ) TO reported-employee.  ENDIF.  ENDIF.  ENDIF.  ENDIF.  ENDMETHOD.  METHOD save.  zcl\_employee\_api\_class=>get\_instance( )->savedata(  CHANGING  reported = reported  ).  ENDMETHOD.  METHOD cleanup.  ENDMETHOD.  METHOD cleanup\_finalize.  ENDMETHOD.  ENDCLASS. |

NOTE:

* **create:** This method uses the create\_employee function of the zcl\_employee\_api\_class class to create a new employee. It exports the entities and changes the mapped, failed, and reported parameters.

|  |
| --- |
| METHOD create.  zcl\_employee\_api\_class=>get\_instance( )->create\_employee(  EXPORTING  entities = entities  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD. |

* **earlynumbering\_create:** Similar to the create method, but it uses the earlynumbering\_create\_employee function of the zcl\_employee\_api\_class class.

|  |
| --- |
| METHOD earlynumbering\_create.  zcl\_employee\_api\_class=>get\_instance( )->earlynumbering\_create\_employee(  EXPORTING  entities = entities  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD. |

* **update:** This method uses the update\_employee function of the zcl\_employee\_api\_class class to update an existing employee’s details. It exports the entities and changes the mapped, failed, and reported parameters.

|  |
| --- |
| METHOD update.  zcl\_employee\_api\_class=>get\_instance( )->update\_employee(  EXPORTING  entities = entities  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD. |

* **delete:** This method uses the delete\_employee function of the zcl\_employee\_api\_class class to delete an employee. It exports the keys and changes the mapped, failed, and reported parameters.

|  |
| --- |
| METHOD delete.  zcl\_employee\_api\_class=>get\_instance( )->delete\_employee(  EXPORTING  keys = keys  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD. |

* **read:** This method uses the read\_employee function of the zcl\_employee\_api\_class class to read an employee’s details. It exports the keys and changes the result, failed, and reported parameters.

|  |
| --- |
| METHOD read.  zcl\_employee\_api\_class=>get\_instance( )->read\_employee(  EXPORTING  keys = keys  CHANGING  result = result  failed = failed  reported = reported  ).  ENDMETHOD. |

* **lock:** This method is used to lock an employee record to prevent concurrent modifications. It uses the cl\_abap\_lock\_object\_factory class to get a lock object and then enqueues the lock. If the record is already locked by another user, it reports the lock and fails the operation.

|  |
| --- |
| METHOD lock.  Try.  DATA(lock) = cl\_abap\_lock\_object\_factory=>get\_instance( iv\_name = 'EYHRM\_U\_LOCKEMP' ).  catch cx\_abap\_lock\_failure into DATA(exception).  RAISE SHORTDUMP exception.  ENDTRY.  LOOP AT keys ASSIGNING FIELD-SYMBOL(<lfs\_employee>).  try.  lock->enqueue(  it\_parameter = VALUE #( ( name = 'EmpId' value = ref #( <lfs\_employee>-EmpId ) ) )  ).  catch cx\_abap\_foreign\_lock into DATA(foreign\_lock).  APPEND VALUE #(  EmpId = keys[ 1 ]-EmpId  %msg = new\_message\_With\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Record is locked by : ' && foreign\_lock->user\_name  )  ) TO reported-employee.  APPEND VALUE #(  EmpId = keys[ 1 ]-EmpId  ) TO failed-employee.  catch cx\_abap\_lock\_failure into exception.  RAISE SHORTDUMP exception.  ENDTRY.  ENDLOOP.  ENDMETHOD. |

* **cba\_timesheet:** This method uses the cba\_timesheet function of the zcl\_employee\_api\_class class. It exports the entities\_cba and changes the mapped, failed, and reported parameters.

|  |
| --- |
| METHOD cba\_timesheet.  zcl\_employee\_api\_class=>get\_instance( )->cba\_timesheet(  EXPORTING  entities\_cba = entities\_cba  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD. |

* **earlynumbering\_cba\_timesheet:** Similar to the cba\_timesheet method, but it uses the earlynumbering\_cba\_timesheet function of the zcl\_employee\_api\_class class.

|  |
| --- |
| METHOD earlynumbering\_cba\_timesheet.  zcl\_employee\_api\_class=>get\_instance( )->earlynumbering\_cba\_timesheet(  EXPORTING  entities = entities  CHANGING  mapped = mapped  failed = failed  reported = reported  ).  ENDMETHOD. |

* **validate\_fields:** This method validates the fields of an employee record. It checks if the FirstName, Gender, Dob, and Email fields are not initial (i.e., they have been assigned values). If any of these fields are initial, it reports the error and fails the operation.

|  |
| --- |
| METHOD validate\_fields.  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY employee  ALL FIELDS WITH CORRESPONDING #( keys )  RESULT DATA(lt\_employee\_tmp)  REPORTED DATA(lt\_reported)  FAILED DATA(lt\_failed).  IF NOT lt\_employee\_tmp[] is INITIAL.  READ TABLE lt\_employee\_tmp ASSIGNING FIELD-SYMBOL(<lfs\_employee\_tmp>) index 1.  if <lfs\_employee\_tmp> is ASSIGNED.  reported-employee = VALUE #(  ( %tky = <lfs\_employee\_tmp>-%tky %state\_area = 'VALIDATE\_FNM' )  ( %tky = <lfs\_employee\_tmp>-%tky %state\_area = 'VALIDATE\_GENDER' )  ( %tky = <lfs\_employee\_tmp>-%tky %state\_area = 'VALIDATE\_DOB' )  ( %tky = <lfs\_employee\_tmp>-%tky %state\_area = 'VALIDATE\_EMAIL' )  ).  if <lfs\_employee\_tmp>-FirstName is INITIAL or  <lfs\_employee\_tmp>-Email is INITIAL or  <lfs\_employee\_tmp>-Dob is INITIAL or  <lfs\_employee\_tmp>-Gender is INITIAL.  failed-employee = VALUE #( ( %tky = <lfs\_employee\_tmp>-%tky ) ).  IF <lfs\_employee\_tmp>-FirstName is INITIAL.  reported-employee = VALUE #( (  %tky = <lfs\_employee\_tmp>-%tky  %state\_area = 'VALIDATE\_FNM'  %element-firstname = if\_abap\_behv=>mk-on  %msg = new\_message(  id = 'SY'  number = '002'  severity = if\_abap\_behv\_message=>severity-error  v1 = 'FirstName is Required!'  )  ) ).  ENDIF.  IF <lfs\_employee\_tmp>-Gender is INITIAL.  reported-employee = VALUE #( BASE reported-employee (  %tky = <lfs\_employee\_tmp>-%tky  %state\_area = 'VALIDATE\_GENDER'  %element-gender = if\_abap\_behv=>mk-on  %msg = new\_message(  id = 'SY'  number = '002'  severity = if\_abap\_behv\_message=>severity-error  v1 = 'Gender is Required!'  )  ) ).  ENDIF.  IF <lfs\_employee\_tmp>-Dob is INITIAL.  reported-employee = VALUE #( BASE reported-employee (  %tky = <lfs\_employee\_tmp>-%tky  %state\_area = 'VALIDATE\_DOB'  %element-dob = if\_abap\_behv=>mk-on  %msg = new\_message(  id = 'SY'  number = '002'  severity = if\_abap\_behv\_message=>severity-error  v1 = 'Date of Birth is Required!'  )  ) ).  ENDIF.  IF <lfs\_employee\_tmp>-Email is INITIAL.  reported-employee = VALUE #( BASE reported-employee (  %tky = <lfs\_employee\_tmp>-%tky  %state\_area = 'VALIDATE\_EMAIL'  %element-email = if\_abap\_behv=>mk-on  %msg = new\_message(  id = 'SY'  number = '002'  severity = if\_abap\_behv\_message=>severity-error  v1 = 'Email is Required!'  )  ) ).  ENDIF.  ENDIF.  ENDIF.  ENDIF.  ENDMETHOD. |

* **updateemployeename:** This method updates the first name of an employee based on their gender. If the gender is ‘M’, it prefixes the first name with ‘Mr.’. If the gender is ‘F’, it prefixes the first name with ‘Mrs.’.

|  |
| --- |
| METHOD updateemployeename.  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY employee  FIELDS ( Gender ) WITH CORRESPONDING #( keys )  RESULT DATA(lt\_employee).  LOOP at lt\_employee ASSIGNING FIELD-SYMBOL(<lfs\_employee>).  DATA: lv\_firstname TYPE string.  lv\_firstname = <lfs\_employee>-FirstName.  IF lv\_firstname CP 'Mr.\*' OR lv\_firstname CP 'Mrs.\*'.  SPLIT lv\_firstname AT ' ' INTO TABLE DATA(lt\_name\_parts).  DELETE lt\_name\_parts INDEX 1.  CONCATENATE LINES OF lt\_name\_parts INTO lv\_firstname SEPARATED BY SPACE.  ENDIF.  IF <lfs\_employee>-Gender EQ 'M'.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Employee  UPDATE FIELDS ( FirstName )  WITH VALUE #( (  %tky = <lfs\_employee>-%tky  FirstName = |Mr. { lv\_firstname }|  ) ).  ELSEIF <lfs\_employee>-Gender = 'F'.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Employee  UPDATE FIELDS ( FirstName )  WITH VALUE #( (  %tky = <lfs\_employee>-%tky  FirstName = |Mrs. { lv\_firstname }|  ) ).  ENDIF.  ENDLOOP.  ENDMETHOD. |

* **updateEmployeeStatus:** This method updates the active status of an employee. It reads the current status of the employee and updates it with the new status provided in the keys.

|  |
| --- |
| METHOD updateEmployeeStatus.  DATA(lt\_keys) = keys.  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Employee  FIELDS ( Active ) WITH CORRESPONDING #( keys )  RESULT DATA(lt\_employee\_status).  DATA(lv\_new\_status) = lt\_keys[ 1 ]-%param-active.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  in LOCAL MODE  ENTITY Employee  UPDATE FIELDS ( Active )  WITH VALUE #( (  %tky = lt\_employee\_status[ 1 ]-%tky Active = lv\_new\_status  ) ).  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Employee  ALL FIELDS WITH CORRESPONDING #( keys )  RESULT DATA(lt\_employee).  result = VALUE #( FOR <lfs\_employee> in lt\_employee (  %tky = <lfs\_employee>-%tky  %param = <lfs\_employee>  ) ).  ENDMETHOD. |

* **updateHours:** This method updates the timesheet of an employee. If the employee is available (‘X’), it clears the leave type. If the employee is not available, it sets the working hours and overtime hours to 0.

|  |
| --- |
| METHOD updateHours.  READ ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Timesheet  FIELDS ( Available ) WITH CORRESPONDING #( keys )  RESULT DATA(lt\_timesheet).  LOOP at lt\_timesheet ASSIGNING FIELD-SYMBOL(<lfs\_timesheet>).  IF <lfs\_timesheet>-Available EQ 'X'.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Timesheet  UPDATE FIELDS ( Leavetype )  WITH VALUE #( (  %tky = <lfs\_timesheet>-%tky  Leavetype = ''  ) ).  ELSE.  MODIFY ENTITIES OF yhrm\_U\_EMPLOYEE  IN LOCAL MODE  ENTITY Timesheet  UPDATE FIELDS ( Workinghours OvertimeHrs )  WITH VALUE #( (  %tky = <lfs\_timesheet>-%tky  Workinghours = 0  OvertimeHrs = 0  ) ).  ENDIF.  ENDLOOP.  ENDMETHOD. |

* **check\_before\_save:** This method performs several checks before saving an employee’s data. It checks if the employee’s age is at least 16, if the hire date is after the employee’s 16th birthday, if the email address is valid, and if the phone number is exactly 10 digits. For the timesheet data, it checks if the timesheet date is after the hire date, if the leave type is mandatory when the employee is not available, and if the working hours do not exceed 8 and overtime hours do not exceed 6.

|  |
| --- |
| METHOD check\_before\_save.  DATA: gt\_employee\_tmp TYPE STANDARD TABLE OF yhrm\_employee,  gt\_timesheet\_tmp TYPE STANDARD TABLE OF yhrm\_timesheet,  lv\_age TYPE i,  lv\_hire\_date TYPE d,  lv\_dob TYPE d,  lv\_phone\_no\_string TYPE string.  gt\_employee\_tmp = zcl\_employee\_api\_class=>get\_instance( )->gt\_employee.  gt\_timesheet\_tmp = zcl\_employee\_api\_class=>get\_instance( )->gt\_timesheet.  IF NOT gt\_employee\_tmp[] IS INITIAL.  READ TABLE gt\_employee\_tmp ASSIGNING FIELD-SYMBOL(<lfs\_employee\_tmp>) INDEX 1.  IF <lfs\_employee\_tmp> IS ASSIGNED.  IF <lfs\_employee\_tmp>-dob IS NOT INITIAL.  lv\_age = trunc( ( sy-datum - <lfs\_employee\_tmp>-dob ) / 365 ).  lv\_hire\_date = <lfs\_employee\_tmp>-dob + 16 \* 365.  ELSEIF <lfs\_employee\_tmp>-hire\_date IS NOT INITIAL.  lv\_age = trunc( ( sy-datum - <lfs\_employee\_tmp>-hire\_date ) / 365 ).  ENDIF.  IF lv\_age < 16.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Employee age should be 16 or more.'  )  ) TO reported-employee.  ENDIF.  IF <lfs\_employee\_tmp>-hire\_date IS NOT INITIAL and <lfs\_employee\_tmp>-hire\_date <= lv\_hire\_date.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Hire date should be after 16 years from DOB.'  )  ) TO reported-employee.  ENDIF.  IF <lfs\_employee\_tmp>-email IS NOT INITIAL AND <lfs\_employee\_tmp>-email CS '@. ' .  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id ) to failed-employee.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Please enter a valid email address.'  )  ) to reported-employee.  ENDIF.  IF <lfs\_employee\_tmp>-phone\_no IS NOT INITIAL.  SHIFT <lfs\_employee\_tmp>-phone\_no LEFT DELETING LEADING '0'.  lv\_phone\_no\_string = <lfs\_employee\_tmp>-phone\_no.  IF strlen( lv\_phone\_no\_string ) <> 10.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id ) to failed-employee.  APPEND VALUE #( empid = <lfs\_employee\_tmp>-emp\_id  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Phone number should be exactly 10 digits.'  )  ) to reported-employee.  ENDIF.  ENDIF.  ENDIF.  ENDIF.  IF NOT gt\_timesheet\_tmp[] IS INITIAL.  READ TABLE gt\_timesheet\_tmp ASSIGNING FIELD-SYMBOL(<lfs\_timesheet\_tmp>) INDEX 1.  IF <lfs\_timesheet\_tmp> IS ASSIGNED.  IF <lfs\_timesheet\_tmp>-ydate IS NOT INITIAL AND <lfs\_timesheet\_tmp>-ydate <= <lfs\_employee\_tmp>-hire\_date.  APPEND VALUE #( empid = <lfs\_timesheet\_tmp>-empid  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Timesheet date should be after Hiredate.'  )  ) TO reported-employee.  ENDIF.  IF <lfs\_timesheet\_tmp>-available IS NOT INITIAL AND NOT <lfs\_timesheet\_tmp>-leavetype IS INITIAL AND <lfs\_timesheet\_tmp>-overtime\_hrs > 0.  APPEND VALUE #( empid = <lfs\_timesheet\_tmp>-empid  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'If employee not available, Leave type should be mandatory, and overtime hours must be 0.'  )  ) TO reported-employee.  ENDIF.  IF <lfs\_timesheet\_tmp>-workinghours IS NOT INITIAL AND <lfs\_timesheet\_tmp>-overtime\_hrs IS NOT INITIAL.  IF <lfs\_timesheet\_tmp>-workinghours > 8 OR <lfs\_timesheet\_tmp>-overtime\_hrs > 6.  APPEND VALUE #( empid = <lfs\_timesheet\_tmp>-empid  %msg = new\_message\_with\_text(  severity = if\_abap\_behv\_message=>severity-error  text = 'Working hours should not exceed 8 and overtime hours should not exceed 6.'  )  ) TO reported-employee.  ENDIF.  ENDIF.  ENDIF.  ENDIF.  ENDMETHOD. |

* **save:** This method saves the employee’s data by calling the savedata function of the zcl\_employee\_api\_class class. It changes the reported parameter.

|  |
| --- |
| METHOD save.  zcl\_employee\_api\_class=>get\_instance( )->savedata(  CHANGING  reported = reported  ).  ENDMETHOD. |

**ZCL\_EMPLOYEE\_API\_CLASS** (Source Code Library / Classes)

|  |
| --- |
| CLASS zcl\_employee\_api\_class DEFINITION  PUBLIC  FINAL  CREATE PUBLIC .  PUBLIC SECTION.  DATA:  gt\_employee TYPE STANDARD TABLE OF yhrm\_employee,  gt\_timesheet TYPE STANDARD TABLE OF yhrm\_timesheet,  gt\_department TYPE STANDARD TABLE OF yhrm\_department,  gt\_address TYPE STANDARD TABLE OF yhrm\_address,  gt\_job TYPE STANDARD TABLE OF yhrm\_job.  TYPES:  tt\_create\_employee TYPE TABLE FOR create yhrm\_u\_employee,  tt\_mapped\_early TYPE RESPONSE FOR MAPPED EARLY yhrm\_u\_employee,  tt\_failed\_early TYPE RESPONSE FOR FAILED EARLY yhrm\_u\_employee,  tt\_reported\_early TYPE RESPONSE FOR REPORTED EARLY yhrm\_u\_employee,  tt\_reported\_late TYPE RESPONSE FOR REPORTED LATE yhrm\_u\_employee,  tt\_employee\_keys TYPE TABLE FOR READ IMPORT yhrm\_u\_employee\\employee,  tt\_employee\_result TYPE TABLE FOR READ RESULT yhrm\_u\_employee\\employee,  tt\_employee\_update TYPE TABLE FOR UPDATE yhrm\_u\_employee\\employee ,  tt\_employee\_delete TYPE TABLE FOR delete yhrm\_u\_employee\\Employee,  tt\_cba\_timesheet TYPE table for create yhrm\_u\_employee\\employee\\_timesheet  .  CLASS-METHODS: get\_Instance RETURNING VALUE(ro\_instance) TYPE REF TO zcl\_employee\_api\_class.  METHODS:  earlynumbering\_create\_employee  importing entities type tt\_create\_employee "table for create yhrm\_u\_employee\\employee  changing mapped type tt\_mapped\_early "response for mapped early yhrm\_u\_employee  failed type tt\_failed\_early "response for failed early yhrm\_u\_employee  reported type tt\_reported\_early, "response for reported early yhrm\_u\_employee  create\_employee  importing entities type tt\_create\_employee "table for create yhrm\_u\_employee\\employee  changing mapped type tt\_mapped\_early "response for mapped early yhrm\_u\_employee  failed type tt\_failed\_early "response for failed early yhrm\_u\_employee  reported type tt\_reported\_early, "response for reported early yhrm\_u\_employee  update\_employee  importing entities type tt\_employee\_update"table for update yhrm\_u\_employee\\employee  changing mapped type tt\_mapped\_early"response for mapped early yhrm\_u\_employee  failed type tt\_failed\_early"response for failed early yhrm\_u\_employee  reported type tt\_reported\_early, "response for reported early yhrm\_u\_employee  savedata  changing reported type tt\_reported\_late, "response for reported late yhrm\_u\_employee  read\_employee  importing keys type tt\_employee\_keys "table for read import yhrm\_u\_employee\\employee  changing result type tt\_employee\_result "table for read result yhrm\_u\_employee\\employee  failed type tt\_failed\_early "response for failed early yhrm\_u\_employee  reported type tt\_reported\_early, "response for reported early yhrm\_u\_employee  delete\_employee  importing keys type tt\_employee\_delete "table for delete yhrm\_u\_employee\\employee  changing mapped type tt\_mapped\_early "response for mapped early yhrm\_u\_employee  failed type tt\_failed\_early "response for failed early yhrm\_u\_employee  reported type tt\_reported\_early, "response for reported early yhrm\_u\_employee  earlynumbering\_cba\_timesheet  importing entities type tt\_cba\_timesheet "table for create yhrm\_u\_employee\\employee\\_timesheet  changing mapped type tt\_mapped\_early"response for mapped early yhrm\_u\_employee  failed type tt\_failed\_early"response for failed early yhrm\_u\_employee  reported type tt\_reported\_early, "response for reported early yhrm\_u\_employee  cba\_timesheet  importing entities\_cba type tt\_cba\_timesheet "table for create yhrm\_u\_employee\\employee\\_timesheet  changing mapped type tt\_mapped\_early"response for mapped early yhrm\_u\_employee  failed type tt\_failed\_early"response for failed early yhrm\_u\_employee  reported type tt\_reported\_early"response for reported early yhrm\_u\_employee  .  METHODS get\_next\_id  EXPORTING rv\_id type sysuuid\_x16.  METHODS get\_next\_employee\_id  EXPORTING rv\_empid type yhrm\_emp\_id.  PROTECTED SECTION.  PRIVATE SECTION.  CLASS-DATA: mo\_instance TYPE REF TO zcl\_employee\_api\_class,  gr\_employee\_d TYPE RANGE OF yhrm\_employee-emp\_id,  lv\_timestampl TYPE timestampl,  gs\_mapped TYPE tt\_mapped\_early.  ENDCLASS.  CLASS zcl\_employee\_api\_class IMPLEMENTATION.  METHOD get\_instance.  mo\_instance = ro\_instance = COND #( When mo\_instance IS BOUND  THEN mo\_instance  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 get\_next\_employee\_id.  DATA: lv\_max\_employeeid TYPE yhrm\_employee-emp\_id,  lv\_emp\_number TYPE i,  lv\_new\_empid TYPE yhrm\_employee-emp\_id,  lv\_emp\_number\_char TYPE c LENGTH 4.  SELECT emp\_id FROM yhrm\_employee ORDER BY emp\_id DESCENDING INTO @lv\_max\_employeeid UP TO 1 ROWS.  ENDSELECT.  lv\_emp\_number = CONV i( lv\_max\_employeeid+3(4) ).  lv\_emp\_number = lv\_emp\_number + 1.  lv\_emp\_number\_char = CONV #( lv\_emp\_number ).  IF strlen( lv\_emp\_number\_char ) = 1.  CONCATENATE 'EMP000' lv\_emp\_number\_char INTO lv\_new\_empid.  ELSEIF strlen( lv\_emp\_number\_char ) = 2.  CONCATENATE 'EMP00' lv\_emp\_number\_char INTO lv\_new\_empid.  ELSEIF strlen( lv\_emp\_number\_char ) = 3.  CONCATENATE 'EMP0' lv\_emp\_number\_char INTO lv\_new\_empid.  ELSEIF strlen( lv\_emp\_number\_char ) = 4.  CONCATENATE 'EMP' lv\_emp\_number\_char INTO lv\_new\_empid.  ELSE.  " Handle the error or warning here  ENDIF.  rv\_empid = lv\_new\_empid.  ENDMETHOD.  METHOD earlynumbering\_create\_employee.  DATA(ls\_mapped) = gs\_mapped.  get\_next\_employee\_id(  IMPORTING rv\_empid = DATA(lv\_new\_empid)  ).  READ TABLE gt\_employee ASSIGNING FIELD-SYMBOL(<lfs\_employee>) INDEX 1.  IF <lfs\_employee> is ASSIGNED.  <lfs\_employee>-emp\_id = lv\_new\_empid.  UNASSIGN <lfs\_employee>.  ENDIF.  mapped-employee = VALUE #(  FOR ls\_entities In entities WHERE ( empid is INITIAL )  (  %cid = ls\_entities-%cid  %is\_draft = ls\_entities-%is\_draft  EmpId = lv\_new\_empid  )  ).  ENDMETHOD.  METHOD create\_employee.  gt\_employee = CORRESPONDING #( entities MAPPING FROM ENTITY ).  get time STAMP FIELD lv\_timestampl.  " Get current user  DATA(lv\_syuname) = sy-uname.  " Assign values to the fields  gt\_employee[ 1 ]-created\_by = lv\_syuname.  gt\_employee[ 1 ]-created\_at = lv\_timestampl.  gt\_employee[ 1 ]-last\_changed\_by = lv\_syuname.  gt\_employee[ 1 ]-local\_last\_changed\_by = lv\_syuname.  gt\_employee[ 1 ]-local\_last\_changed\_at = lv\_timestampl.  gt\_employee[ 1 ]-last\_changed\_at = lv\_timestampl.  gt\_employee[ 1 ]-local\_last\_changed\_at = lv\_timestampl.  gt\_employee[ 1 ]-last\_changed\_at = lv\_timestampl.  mapped = VALUE #(  employee = value #(  FOR ls\_entity IN entities (  %cid = ls\_entity-%cid  %key = ls\_entity-%key  %is\_draft = ls\_entity-%is\_draft  )  )  ).  \* Loop at entities ASSIGNING FIELD-SYMBOL(<lfs\_entities>).  \* IF Not gt\_employee[] is INITIAL.  \* get\_next\_employee\_id(  \* IMPORTING rv\_empid = gt\_employee[ 1 ]-emp\_id  \* ).  \*  \* mapped-employee = VALUE #( (  \*  \* %cid = <lfs\_entities>-%cid  \* %key = <lfs\_entities>-%key  \* %is\_draft = <lfs\_entities>-%is\_draft  \* ) ).  \* ENDIF.  \*  \* ENDLOOP.  ENDMETHOD.  METHOD savedata.  IF NOT gt\_employee[] IS INITIAL.  MODIFY yhrm\_employee FROM TABLE @gt\_employee.  ENDIF.  IF NOT gt\_timesheet[] IS INITIAL.  MODIFY yhrm\_timesheet FROM TABLE @gt\_timesheet.  ENDIF.  IF NOT gr\_employee\_d is INITIAL.  DELETE FROM yhrm\_employee WHERE emp\_id IN @gr\_employee\_d.  ENDIF.  ENDMETHOD.  METHOD read\_employee.  SELECT \* FROM yhrm\_employee FOR ALL ENTRIES IN @keys  WHERE emp\_id = @keys-Empid  into TABLE @DATA(lt\_employee\_data).  result = CORRESPONDING #( lt\_employee\_data MAPPING TO ENTITY ).  ENDMETHOD.  METHOD update\_employee.  DATA: lt\_employee\_update TYPE STANDARD TABLE OF yhrm\_employee,  lt\_employee\_update\_x TYPE STANDARD TABLE OF yhrm\_emp\_u\_structure.  lt\_employee\_update = CORRESPONDING #( entities MAPPING FROM ENTITY ).  lt\_employee\_update\_x = CORRESPONDING #( entities MAPPING FROM ENTITY using CONTROL ).  get time STAMP FIELD lv\_timestampl.  " Get current user  DATA(lv\_syuname) = sy-uname.  if not lt\_employee\_update is INITIAL.  SELECT \* FROM yhrm\_employee  FOR ALL ENTRIES IN @lt\_employee\_update  where emp\_id = @lt\_employee\_update-emp\_id  into TABLE @DATA(lt\_employee\_update\_old).  ENDIF.  gt\_employee = VALUE #(  FOR x = 1 WHILE x <= lines( lt\_employee\_update )  LET  ls\_control\_flag = VALUE #( lt\_employee\_update\_x[ x ] OPTIONAL )  ls\_employee\_new = VALUE #( lt\_employee\_update[ x ] OPTIONAL )  ls\_employee\_old = VALUE #( lt\_employee\_update\_old[ emp\_id = ls\_employee\_new-emp\_id ] OPTIONAL )  IN  (  emp\_id = COND #( When ls\_control\_flag-empid is not INITIAL  then ls\_employee\_new-emp\_id  else ls\_employee\_old-emp\_id )  active = COND #( When ls\_control\_flag-active is not INITIAL  then ls\_employee\_new-active  else ls\_employee\_old-active )  address\_id = COND #( When ls\_control\_flag-addressid is not INITIAL  then ls\_employee\_new-address\_id  else ls\_employee\_old-address\_id )  department\_id = COND #( When ls\_control\_flag-departmentid is not INITIAL  then ls\_employee\_new-department\_id  else ls\_employee\_old-department\_id )  dob = COND #( When ls\_control\_flag-dob is not INITIAL  then ls\_employee\_new-dob  else ls\_employee\_old-dob )  email = COND #( When ls\_control\_flag-email is not INITIAL  then ls\_employee\_new-email  else ls\_employee\_old-email )  first\_name = COND #( When ls\_control\_flag-firstname is not INITIAL  then ls\_employee\_new-first\_name  else ls\_employee\_old-first\_name )  gender = COND #( When ls\_control\_flag-gender is not INITIAL  then ls\_employee\_new-gender  else ls\_employee\_old-gender )  hire\_date = COND #( When ls\_control\_flag-hiredate is not INITIAL  then ls\_employee\_new-hire\_date  else ls\_employee\_old-hire\_date )  job\_id = COND #( When ls\_control\_flag-jobid is not INITIAL  then ls\_employee\_new-job\_id  else ls\_employee\_old-job\_id )  last\_name = COND #( When ls\_control\_flag-lastname is not INITIAL  then ls\_employee\_new-last\_name  else ls\_employee\_old-last\_name )  phone\_no = COND #( When ls\_control\_flag-phoneno is not INITIAL  then ls\_employee\_new-phone\_no  else ls\_employee\_old-phone\_no )  resign\_date = COND #( When ls\_control\_flag-resigndate is not INITIAL  then ls\_employee\_new-resign\_date  else ls\_employee\_old-resign\_date )  salary = COND #( When ls\_control\_flag-salary is not INITIAL  then ls\_employee\_new-salary  else ls\_employee\_old-salary )  supervisor\_id = COND #( When ls\_control\_flag-supervisorid is not INITIAL  then ls\_employee\_new-supervisor\_id  else ls\_employee\_old-supervisor\_id )  last\_changed\_at = lv\_timestampl  last\_changed\_by = lv\_syuname  local\_last\_changed\_at = lv\_timestampl  local\_last\_changed\_by = lv\_syuname  )  ).  ENDMETHOD.  METHOD delete\_employee.  DATA: lt\_employee TYPE STANDARD TABLE OF yhrm\_employee.  lt\_employee = CORRESPONDING #( keys MAPPING FROM ENTITY ).  gr\_employee\_d = VALUE #(  FOR ls\_employee\_d In lt\_employee  sign = 'I'  option = 'EQ'  ( low = ls\_employee\_d-emp\_id )  ).  ENDMETHOD.  METHOD earlynumbering\_cba\_timesheet.  Loop at entities ASSIGNING FIELD-SYMBOL(<lfs\_entities>).  loop at <lfs\_entities>-%target ASSIGNING FIELD-SYMBOL(<lfs\_timesheet\_create>).  mapped-timesheet = VALUE #( (  %cid = <lfs\_timesheet\_create>-%cid  %key = <lfs\_timesheet\_create>-%key  %is\_draft = <lfs\_timesheet\_create>-%is\_draft  ) ).  ENDLOOP.  ENDLOOP.  ENDMETHOD.  METHOD cba\_timesheet.  gt\_timesheet = VALUE #(  FOR ls\_entities\_cba IN entities\_cba  FOR ls\_timesheet\_cba IN ls\_entities\_cba-%target  LET  ls\_rap\_timesheet = CORRESPONDING yhrm\_timesheet(  ls\_timesheet\_cba MAPPING FROM ENTITY  )  IN (  ls\_rap\_timesheet  )  ).  mapped = VALUE #(  timesheet = VALUE #(  FOR i = 1 WHILE i <= lines( entities\_cba )  LET  lt\_timesheets = value #( entities\_cba[ i ]-%target OPTIONAL )  IN  FOR j = 1 WHILE j <= lines( lt\_timesheets )  LET  ls\_curr\_timesheet = VALUE #( lt\_timesheets[ j ] OPTIONAL )  IN (  %cid = ls\_curr\_timesheet-%cid  %key = ls\_curr\_timesheet-%key  Empid = ls\_curr\_timesheet-Empid  )  )  ).  ENDMETHOD.  ENDCLASS. |

NOTE:

* **get\_Instance:** This method is a singleton pattern implementation. It ensures that only one instance of the zcl\_employee\_api\_class class is created throughout the program execution. If an instance already exists, it returns the existing instance; otherwise, it creates a new one.

|  |
| --- |
| METHOD get\_instance.  mo\_instance = ro\_instance = COND #( When mo\_instance IS BOUND  THEN mo\_instance  ELSE NEW #( ) ).  ENDMETHOD. |

* **get\_next\_id:** This method generates a new UUID (Universally Unique Identifier) using the cl\_uuid\_factory class. It’s used when a unique ID is needed, such as when creating a new employee record.

|  |
| --- |
| METHOD get\_next\_id.  TRY.  rv\_id = cl\_uuid\_factory=>create\_system\_uuid( )->create\_uuid\_x16( ).  CATCH cx\_uuid\_error.  ENDTRY.  ENDMETHOD. |

* **get\_next\_employee\_id: This** method generates a new employee ID. It first fetches the highest existing employee ID from the yhrm\_employee table, then increments the numeric part of the ID by 1, and finally prefixes it with ‘EMP’ to form the new employee ID.

|  |
| --- |
| METHOD get\_next\_employee\_id.  DATA: lv\_max\_employeeid TYPE yhrm\_employee-emp\_id,  lv\_emp\_number TYPE i,  lv\_new\_empid TYPE yhrm\_employee-emp\_id,  lv\_emp\_number\_char TYPE c LENGTH 4.  SELECT emp\_id FROM yhrm\_employee ORDER BY emp\_id DESCENDING INTO @lv\_max\_employeeid UP TO 1 ROWS.  ENDSELECT.  lv\_emp\_number = CONV i( lv\_max\_employeeid+3(4) ).  lv\_emp\_number = lv\_emp\_number + 1.  lv\_emp\_number\_char = CONV #( lv\_emp\_number ).  IF strlen( lv\_emp\_number\_char ) = 1.  CONCATENATE 'EMP000' lv\_emp\_number\_char INTO lv\_new\_empid.  ELSEIF strlen( lv\_emp\_number\_char ) = 2.  CONCATENATE 'EMP00' lv\_emp\_number\_char INTO lv\_new\_empid.  ELSEIF strlen( lv\_emp\_number\_char ) = 3.  CONCATENATE 'EMP0' lv\_emp\_number\_char INTO lv\_new\_empid.  ELSEIF strlen( lv\_emp\_number\_char ) = 4.  CONCATENATE 'EMP' lv\_emp\_number\_char INTO lv\_new\_empid.  ELSE.  " Handle the error or warning here  ENDIF.  rv\_empid = lv\_new\_empid.  ENDMETHOD. |

* **earlynumbering\_create\_employee:** This method is used to create a new employee record with a unique employee ID. It first gets a new employee ID using the get\_next\_employee\_id method, then assigns this new ID to the employee record in the gt\_employee table. The new employee record is then added to the mapped-employee table.

|  |
| --- |
| METHOD earlynumbering\_create\_employee.  DATA(ls\_mapped) = gs\_mapped.  get\_next\_employee\_id(  IMPORTING rv\_empid = DATA(lv\_new\_empid)  ).  READ TABLE gt\_employee ASSIGNING FIELD-SYMBOL(<lfs\_employee>) INDEX 1.  IF <lfs\_employee> is ASSIGNED.  <lfs\_employee>-emp\_id = lv\_new\_empid.  UNASSIGN <lfs\_employee>.  ENDIF.  mapped-employee = VALUE #(  FOR ls\_entities In entities WHERE ( empid is INITIAL )  (  %cid = ls\_entities-%cid  %is\_draft = ls\_entities-%is\_draft  EmpId = lv\_new\_empid  )  ).  ENDMETHOD. |

* **create\_employee:** This method is used to create a new employee record. It assigns the current timestamp and username to the relevant fields of the employee record. The new employee record is then added to the mapped table.

|  |
| --- |
| METHOD create\_employee.  gt\_employee = CORRESPONDING #( entities MAPPING FROM ENTITY ).  get time STAMP FIELD lv\_timestampl.  " Get current user  DATA(lv\_syuname) = sy-uname.  " Assign values to the fields  gt\_employee[ 1 ]-created\_by = lv\_syuname.  gt\_employee[ 1 ]-created\_at = lv\_timestampl.  gt\_employee[ 1 ]-last\_changed\_by = lv\_syuname.  gt\_employee[ 1 ]-local\_last\_changed\_by = lv\_syuname.  gt\_employee[ 1 ]-local\_last\_changed\_at = lv\_timestampl.  gt\_employee[ 1 ]-last\_changed\_at = lv\_timestampl.  gt\_employee[ 1 ]-local\_last\_changed\_at = lv\_timestampl.  gt\_employee[ 1 ]-last\_changed\_at = lv\_timestampl.  mapped = VALUE #(  employee = value #(  FOR ls\_entity IN entities (  %cid = ls\_entity-%cid  %key = ls\_entity-%key  %is\_draft = ls\_entity-%is\_draft  )  )  ).  \* Loop at entities ASSIGNING FIELD-SYMBOL(<lfs\_entities>).  \* IF Not gt\_employee[] is INITIAL.  \* get\_next\_employee\_id(  \* IMPORTING rv\_empid = gt\_employee[ 1 ]-emp\_id  \* ).  \*  \* mapped-employee = VALUE #( (  \*  \* %cid = <lfs\_entities>-%cid  \* %key = <lfs\_entities>-%key  \* %is\_draft = <lfs\_entities>-%is\_draft  \* ) ).  \* ENDIF.  \*  \* ENDLOOP.  ENDMETHOD. |

* **savedata:** This method saves the employee and timesheet data to the yhrm\_employee and yhrm\_timesheet tables respectively. If there are any employees to be deleted (indicated by gr\_employee\_d), it deletes those records from the yhrm\_employee table.

|  |
| --- |
| METHOD savedata.  IF NOT gt\_employee[] IS INITIAL.  MODIFY yhrm\_employee FROM TABLE @gt\_employee.  ENDIF.  IF NOT gt\_timesheet[] IS INITIAL.  MODIFY yhrm\_timesheet FROM TABLE @gt\_timesheet.  ENDIF.  IF NOT gr\_employee\_d is INITIAL.  DELETE FROM yhrm\_employee WHERE emp\_id IN @gr\_employee\_d.  ENDIF.  ENDMETHOD. |

* **read\_employee:** This method reads employee data from the yhrm\_employee table for the given keys. The result is mapped to the result table.

|  |
| --- |
| METHOD read\_employee.  SELECT \* FROM yhrm\_employee FOR ALL ENTRIES IN @keys  WHERE emp\_id = @keys-Empid  into TABLE @DATA(lt\_employee\_data).  result = CORRESPONDING #( lt\_employee\_data MAPPING TO ENTITY ).  ENDMETHOD. |

* **update\_employee:** This method updates an employee record. First, a structure of the yhrm\_employee table is created. This structure represents an employee record and contains fields for each attribute of an employee.

|  |
| --- |
| @EndUserText.label : 'structure for update employee unmanaged'  @AbapCatalog.enhancement.category : #NOT\_EXTENSIBLE  **define** **structure** yhrm\_emp\_u\_structure **{**  **key** empid **:** xsdboolean **not** **null;**  firstname **:** xsdboolean**;**  lastname **:** xsdboolean**;**  email **:** xsdboolean**;**  phoneno **:** xsdboolean**;**  dob **:** xsdboolean**;**  gender **:** xsdboolean**;**  salary **:** xsdboolean**;**  hiredate **:** xsdboolean**;**  active **:** xsdboolean**;**  resigndate **:** xsdboolean**;**  addressid **:** xsdboolean**;**  jobid **:** xsdboolean**;**  departmentid **:** xsdboolean**;**  supervisorid **:** xsdboolean**;**  created\_by **:** xsdboolean**;**  created\_at **:** xsdboolean**;**  last\_changed\_by **:** xsdboolean**;**  local\_last\_changed\_by **:** xsdboolean**;**  local\_last\_changed\_at **:** xsdboolean**;**  last\_changed\_at **:** xsdboolean**;**  **}** |

It first reads the old employee data from the yhrm\_employee table. Then, it updates the fields of the employee record based on the control flags. If a control flag for a field is set, it uses the new value; otherwise, it retains the old value. The updated employee record is then added to the gt\_employee table.

|  |
| --- |
| METHOD update\_employee.  DATA: lt\_employee\_update TYPE STANDARD TABLE OF yhrm\_employee,  lt\_employee\_update\_x TYPE STANDARD TABLE OF yhrm\_emp\_u\_structure.  lt\_employee\_update = CORRESPONDING #( entities MAPPING FROM ENTITY ).  lt\_employee\_update\_x = CORRESPONDING #( entities MAPPING FROM ENTITY using CONTROL ).  get time STAMP FIELD lv\_timestampl.  " Get current user  DATA(lv\_syuname) = sy-uname.  \* " Assign values to the fields  \* gt\_employee[ 1 ]-last\_changed\_by = lv\_syuname.  \* gt\_employee[ 1 ]-local\_last\_changed\_by = lv\_syuname.  \* gt\_employee[ 1 ]-local\_last\_changed\_at = lv\_timestampl.  \* gt\_employee[ 1 ]-last\_changed\_at = lv\_timestampl.  if not lt\_employee\_update is INITIAL.  SELECT \* FROM yhrm\_employee  FOR ALL ENTRIES IN @lt\_employee\_update  where emp\_id = @lt\_employee\_update-emp\_id  into TABLE @DATA(lt\_employee\_update\_old).  ENDIF.  gt\_employee = VALUE #(  FOR x = 1 WHILE x <= lines( lt\_employee\_update )  LET  ls\_control\_flag = VALUE #( lt\_employee\_update\_x[ x ] OPTIONAL )  ls\_employee\_new = VALUE #( lt\_employee\_update[ x ] OPTIONAL )  ls\_employee\_old = VALUE #( lt\_employee\_update\_old[ emp\_id = ls\_employee\_new-emp\_id ] OPTIONAL )  IN  (  emp\_id = COND #( When ls\_control\_flag-empid is not INITIAL  then ls\_employee\_new-emp\_id  else ls\_employee\_old-emp\_id )  active = COND #( When ls\_control\_flag-active is not INITIAL  then ls\_employee\_new-active  else ls\_employee\_old-active )  address\_id = COND #( When ls\_control\_flag-addressid is not INITIAL  then ls\_employee\_new-address\_id  else ls\_employee\_old-address\_id )  department\_id = COND #( When ls\_control\_flag-departmentid is not INITIAL  then ls\_employee\_new-department\_id  else ls\_employee\_old-department\_id )  dob = COND #( When ls\_control\_flag-dob is not INITIAL  then ls\_employee\_new-dob  else ls\_employee\_old-dob )  email = COND #( When ls\_control\_flag-email is not INITIAL  then ls\_employee\_new-email  else ls\_employee\_old-email )  first\_name = COND #( When ls\_control\_flag-firstname is not INITIAL  then ls\_employee\_new-first\_name  else ls\_employee\_old-first\_name )  gender = COND #( When ls\_control\_flag-gender is not INITIAL  then ls\_employee\_new-gender  else ls\_employee\_old-gender )  hire\_date = COND #( When ls\_control\_flag-hiredate is not INITIAL  then ls\_employee\_new-hire\_date  else ls\_employee\_old-hire\_date )  job\_id = COND #( When ls\_control\_flag-jobid is not INITIAL  then ls\_employee\_new-job\_id  else ls\_employee\_old-job\_id )  last\_name = COND #( When ls\_control\_flag-lastname is not INITIAL  then ls\_employee\_new-last\_name  else ls\_employee\_old-last\_name )  phone\_no = COND #( When ls\_control\_flag-phoneno is not INITIAL  then ls\_employee\_new-phone\_no  else ls\_employee\_old-phone\_no )  resign\_date = COND #( When ls\_control\_flag-resigndate is not INITIAL  then ls\_employee\_new-resign\_date  else ls\_employee\_old-resign\_date )  salary = COND #( When ls\_control\_flag-salary is not INITIAL  then ls\_employee\_new-salary  else ls\_employee\_old-salary )  supervisor\_id = COND #( When ls\_control\_flag-supervisorid is not INITIAL  then ls\_employee\_new-supervisor\_id  else ls\_employee\_old-supervisor\_id )  last\_changed\_at = lv\_timestampl  last\_changed\_by = lv\_syuname  local\_last\_changed\_at = lv\_timestampl  local\_last\_changed\_by = lv\_syuname  )  ).  ENDMETHOD. |

* **delete\_employee:** This method prepares a range table (gr\_employee\_d) for deleting employee records. The range table contains the employee IDs of the records to be deleted.

|  |
| --- |
| METHOD delete\_employee.  DATA: lt\_employee TYPE STANDARD TABLE OF yhrm\_employee.  lt\_employee = CORRESPONDING #( keys MAPPING FROM ENTITY ).  gr\_employee\_d = VALUE #(  FOR ls\_employee\_d In lt\_employee  sign = 'I'  option = 'EQ'  ( low = ls\_employee\_d-emp\_id )  ).  ENDMETHOD. |

* **earlynumbering\_cba\_timesheet:** This method prepares the mapped table for creating timesheet records. It loops through the entities and their targets, and adds each timesheet record to the mapped table.

|  |
| --- |
| METHOD earlynumbering\_cba\_timesheet.  \* DATA(lv\_new\_timesheet\_id) = get\_next\_id( ).  Loop at entities ASSIGNING FIELD-SYMBOL(<lfs\_entities>).  loop at <lfs\_entities>-%target ASSIGNING FIELD-SYMBOL(<lfs\_timesheet\_create>).  mapped-timesheet = VALUE #( (  %cid = <lfs\_timesheet\_create>-%cid  %key = <lfs\_timesheet\_create>-%key  %is\_draft = <lfs\_timesheet\_create>-%is\_draft  ) ).  ENDLOOP.  ENDLOOP.  ENDMETHOD. |

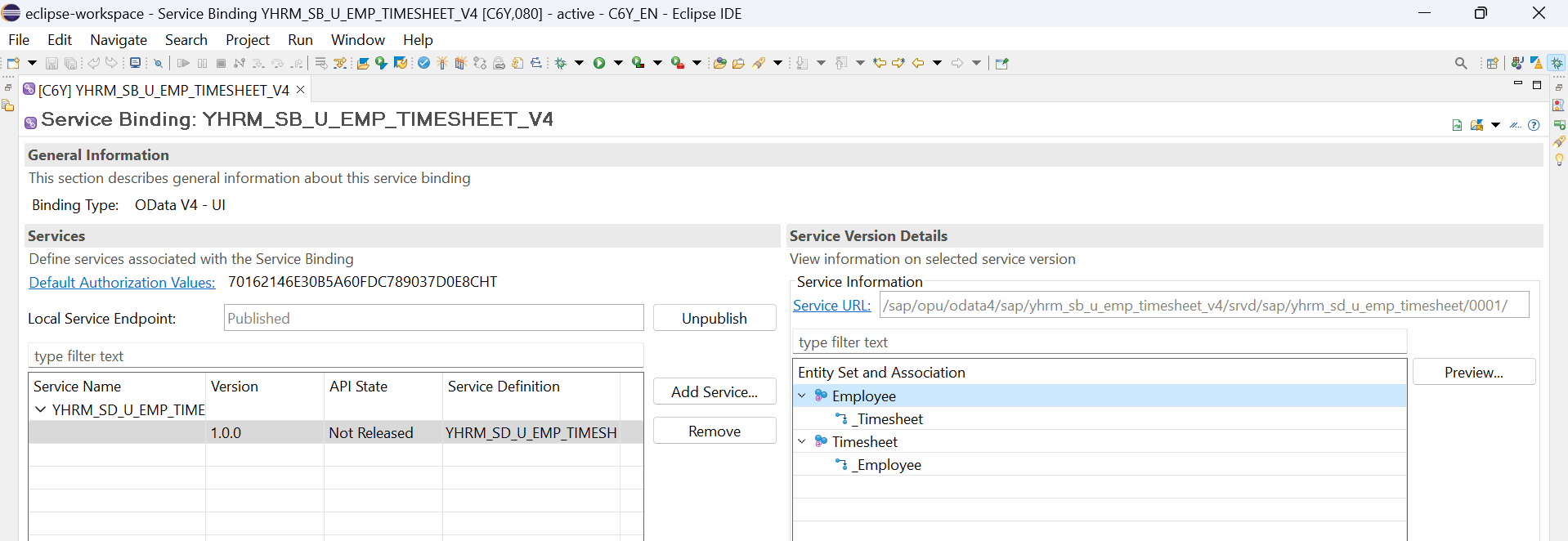
* **cba\_timesheet:** This method is used to create timesheet records for employees. It maps the incoming timesheet data (entities\_cba) to the gt\_timesheet table and prepares the mapped table for further processing. The mapped table contains the %cid, %key, and Empid fields of each timesheet record.

|  |
| --- |
| METHOD cba\_timesheet.  gt\_timesheet = VALUE #(  FOR ls\_entities\_cba IN entities\_cba  FOR ls\_timesheet\_cba IN ls\_entities\_cba-%target  LET  ls\_rap\_timesheet = CORRESPONDING yhrm\_timesheet(  ls\_timesheet\_cba MAPPING FROM ENTITY  )  IN (  ls\_rap\_timesheet  )  ).  mapped = VALUE #(  timesheet = VALUE #(  FOR i = 1 WHILE i <= lines( entities\_cba )  LET  lt\_timesheets = value #( entities\_cba[ i ]-%target OPTIONAL )  IN  FOR j = 1 WHILE j <= lines( lt\_timesheets )  LET  ls\_curr\_timesheet = VALUE #( lt\_timesheets[ j ] OPTIONAL )  IN (  %cid = ls\_curr\_timesheet-%cid  %key = ls\_curr\_timesheet-%key  Empid = ls\_curr\_timesheet-Empid  )  )  ).  ENDMETHOD. |

**YHRM\_SD\_U\_EMP\_TIMESHEET** (Business Services / Service Definition)

|  |
| --- |
| @EndUserText.label: 'SD Employee and Timesheet unmanaged'  **define** **service** YHRM\_SD\_U\_EMP\_TIMESHEET **{**  **expose** YHRM\_U\_EMPLOYEE\_PROJ **as** Employee**;**  **expose** YHRM\_U\_TIMESHEET\_PROJ **as** Timesheet**;**  **}** |

**YHRM\_SB\_U\_EMP\_TIMESHEET\_V4** (Business Services / Service Binding)



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

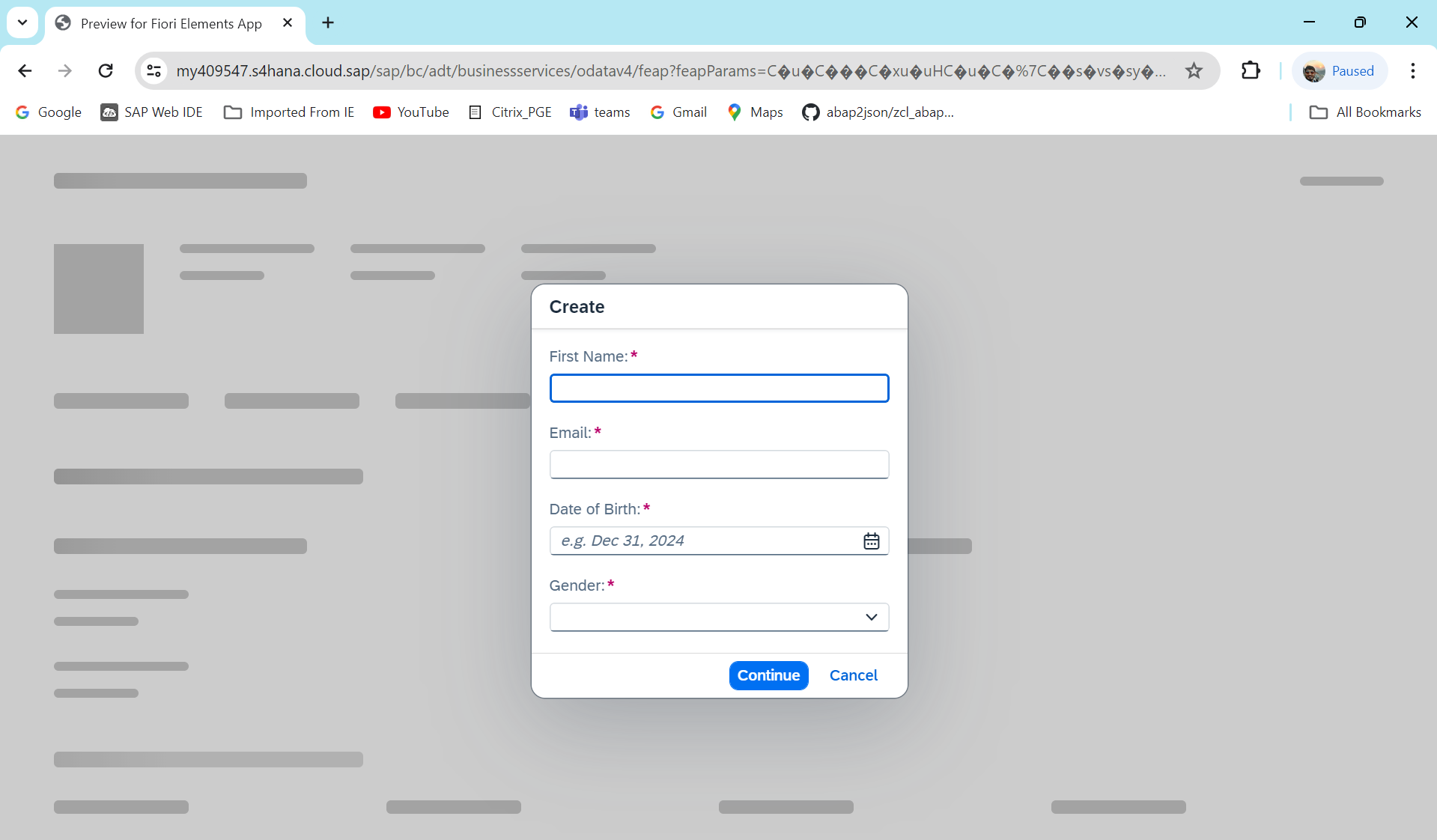
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



Important Links:  
[Unmanaged RAP Model (Student Example)](https://youtube.com/playlist?list=PLU0olhFj5UjUj38zAwoVtt_Hd-TzkLwgt&si=kJ3tw7w8ynYtL_yV)

[RAP Model CodeInMins YouTube](https://youtube.com/playlist?list=PLU0olhFj5UjUSsUz6dwsJMzL5ncdmHCok&si=KBnSo2Pauxf0my-K)

[SAP TECHNOMANIAC YouTube (travel booking example)](https://youtube.com/playlist?list=PLqz8SLrkjv2iwPtPRJ4V6zJRfgNrgZKlQ&si=Uo-sBB8jGoWNh4Cs)

[Managed Scenario Travel Booking Complete Example](https://discoveringabap.com/2021/12/03/abap-restful-application-programming-model-2-working-example/)

[Unmanaged Scenario Travel Booking Example part 1](https://discoveringabap.com/2022/06/07/abap-restful-application-programming-model-4-unmanaged-scenario-part-1/)

[Unmanaged Scenario Travel Booking Example part 2](https://discoveringabap.com/2022/06/08/abap-restful-application-programming-model-5-unmanaged-scenario-part-2/)

[Unmanaged Scenario Travel Booking Example part 3](https://discoveringabap.com/2022/06/11/abap-restful-application-programming-model-6-unmanaged-scenario-part-3/)