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)

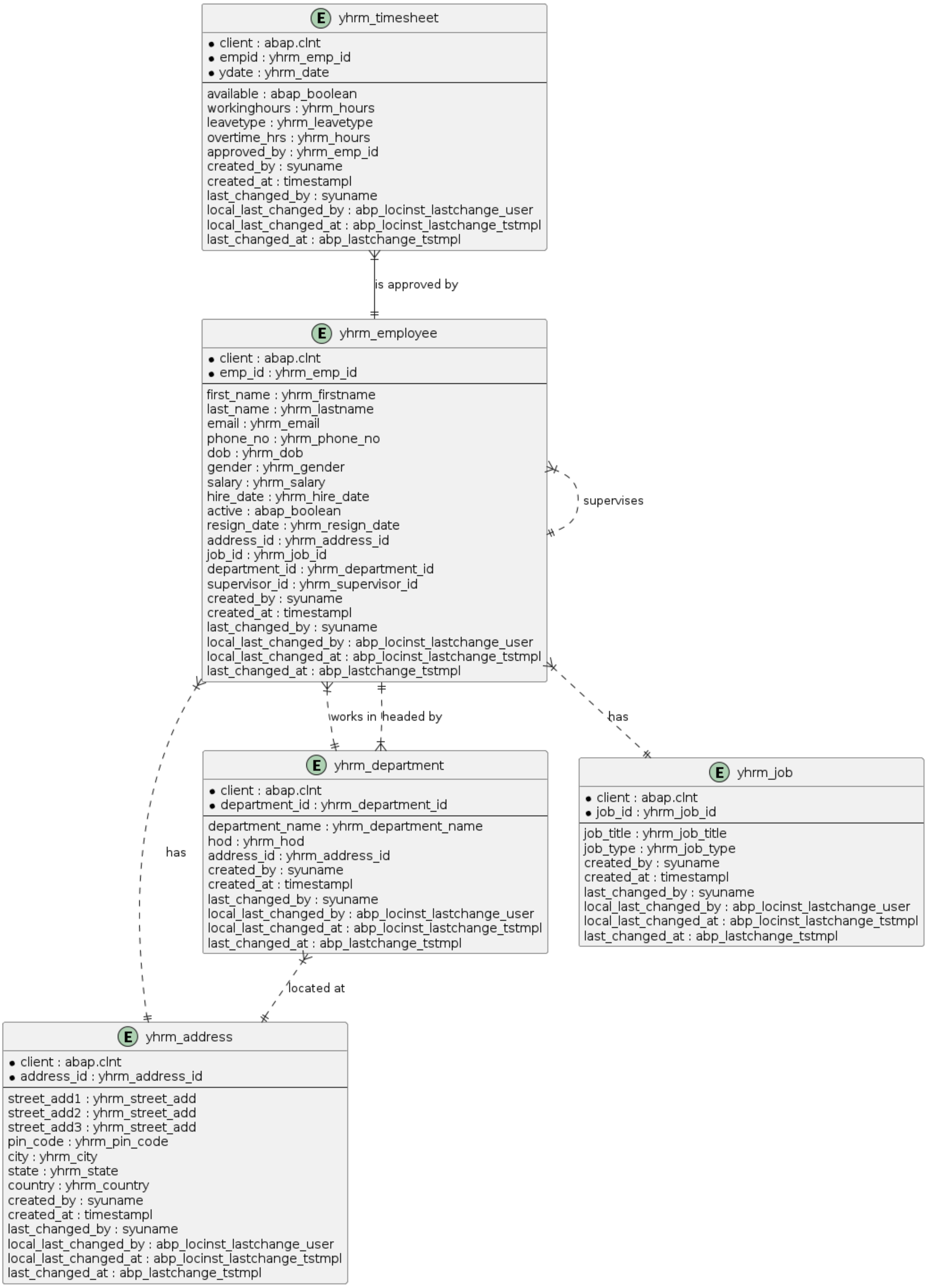
```
@EndUserText.label : 'Job Table'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #ALLOWED
define table yhrm_job {

  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;
}
```


E yhrm_timesheet
• client : abap.clnt
• empid : yhrm_emp_id
• ydate : yhrm_date
available : abap_boolean
workinghours : yhrm_hours
leavetype : yhrm_leavetype
overtime_hrs : yhrm_hours
approved_by : yhrm_emp_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
is approved by
E yhrm_employee
• client : abap.clnt
• emp_id : yhrm_emp_id
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
address_id : yhrm_address_id
job_id : yhrm_job_id
department_id : yhrm_department_id
supervisor_id : 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
supervises
works in
headed by
has
has
E yhrm_department
• client : abap.clnt
• department_id : yhrm_department_id
department_name : yhrm_department_name
hod : yhrm_hod
address_id : 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
E yhrm_job
• client : abap.clnt
• job_id : yhrm_job_id
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
located at
E yhrm_address
• client : abap.clnt
• address_id : yhrm_address_id
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

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)

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

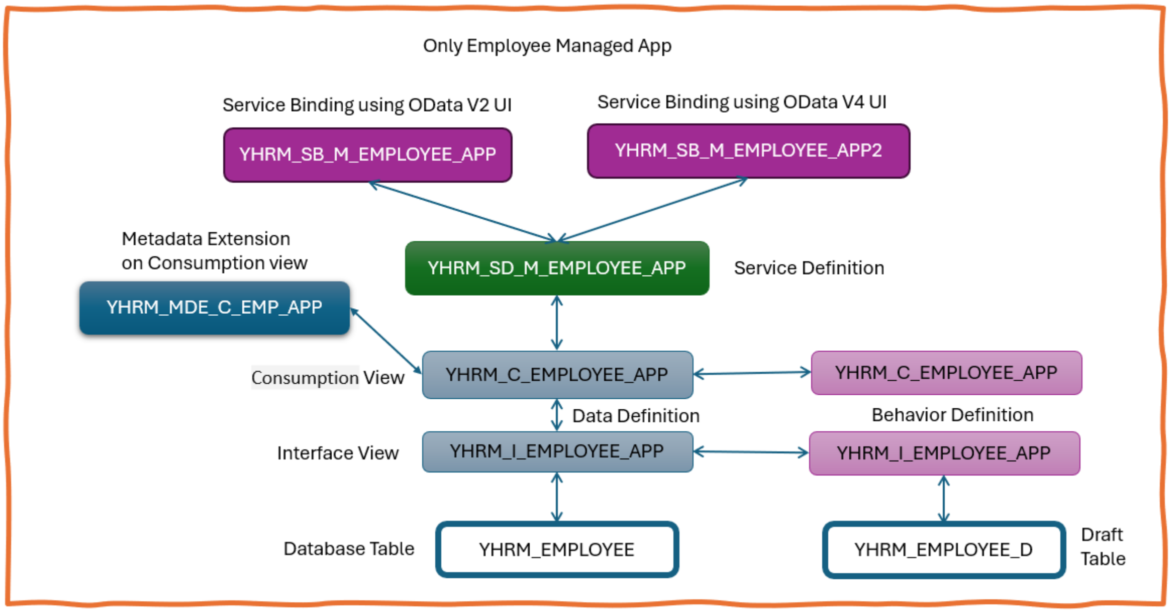
```
@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Help view F4 for Leave Type'
@Metadata.ignorePropagatedAnnotations: true
@ObjectModel.usageType:{
   serviceQuality: #X,
   sizeCategory: #S,
   dataClass: #MIXED
}
@ObjectModel.resultSet.sizeCategory: #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)



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)

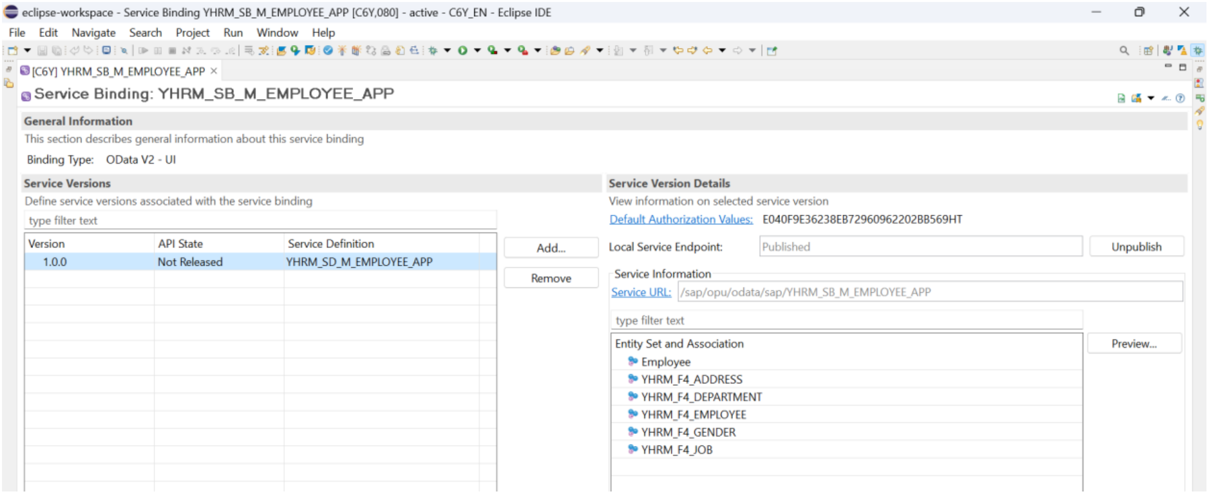
```
@EndUserText.label: 'SD for Employee App'
define service YHRM_SD_M_EMPLOYEE_APP {
  expose YHRM_C_EMPLOYEE_APP as Employee;
}
```

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)



eclipse-workspace - Service Binding YHRM_SB_M_EMPLOYEE_APP [C6Y,080] - active - C6Y_EN - Eclipse IDE

[C6Y] YHRM_SB_M_EMPLOYEE_APP

Service Binding: YHRM_SB_M_EMPLOYEE_APP

General Information

This section describes general information about this service binding

Binding Type: OData V2 - UI

Service Versions

Define service versions associated with the service binding

type filter text

Version	API State	Service Definition
1.0.0	Not Released	YHRM_SD_M_EMPLOYEE_APP

Service Version Details

View information on selected service version

Default Authorization Values: E040F9E36238EB72960962202BB569HT

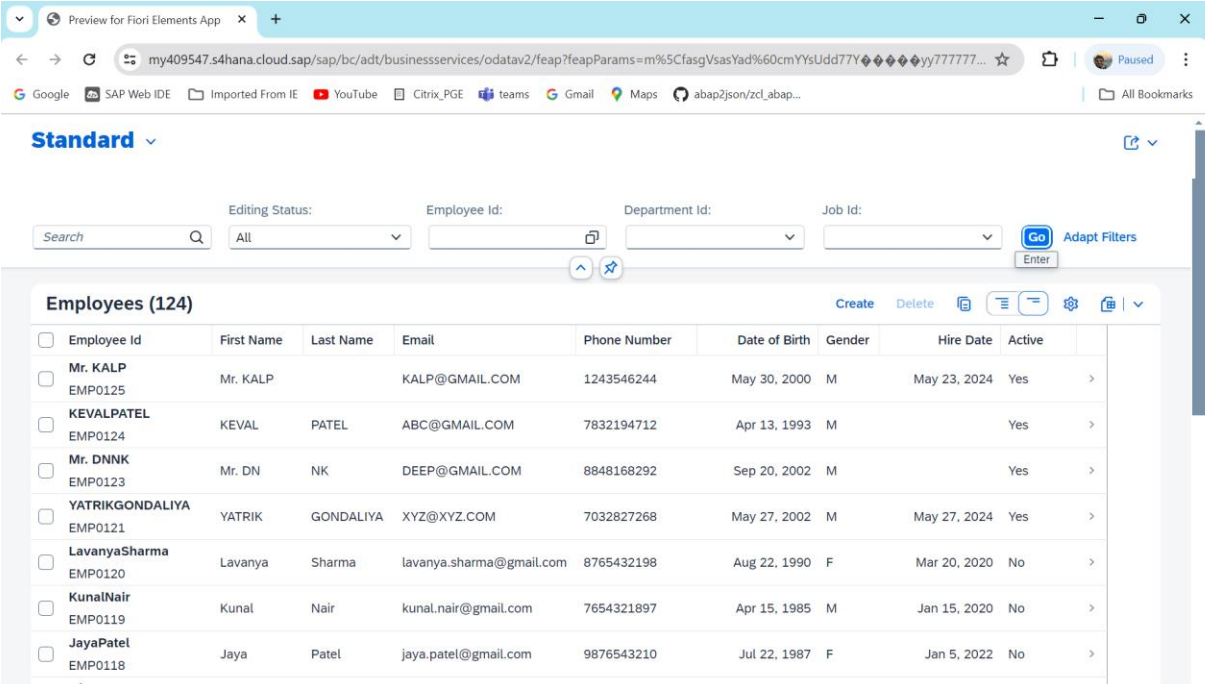
Local Service Endpoint: Published

Service Information

Service URL: /sap/opu/odata/sap/YHRM_SB_M_EMPLOYEE_APP

type filter text

Entity Set and Association
- Employee
- YHRM_F4_ADDRESS
- YHRM_F4_DEPARTMENT
- YHRM_F4_EMPLOYEE
- YHRM_F4_GENDER
- YHRM_F4_JOB



Preview for Fiori Elements App

my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav2/feap?feapParams=m%5CfasgVsasYad%60cmYYsUdd77Y����yy777777...

Search

Editing Status: All

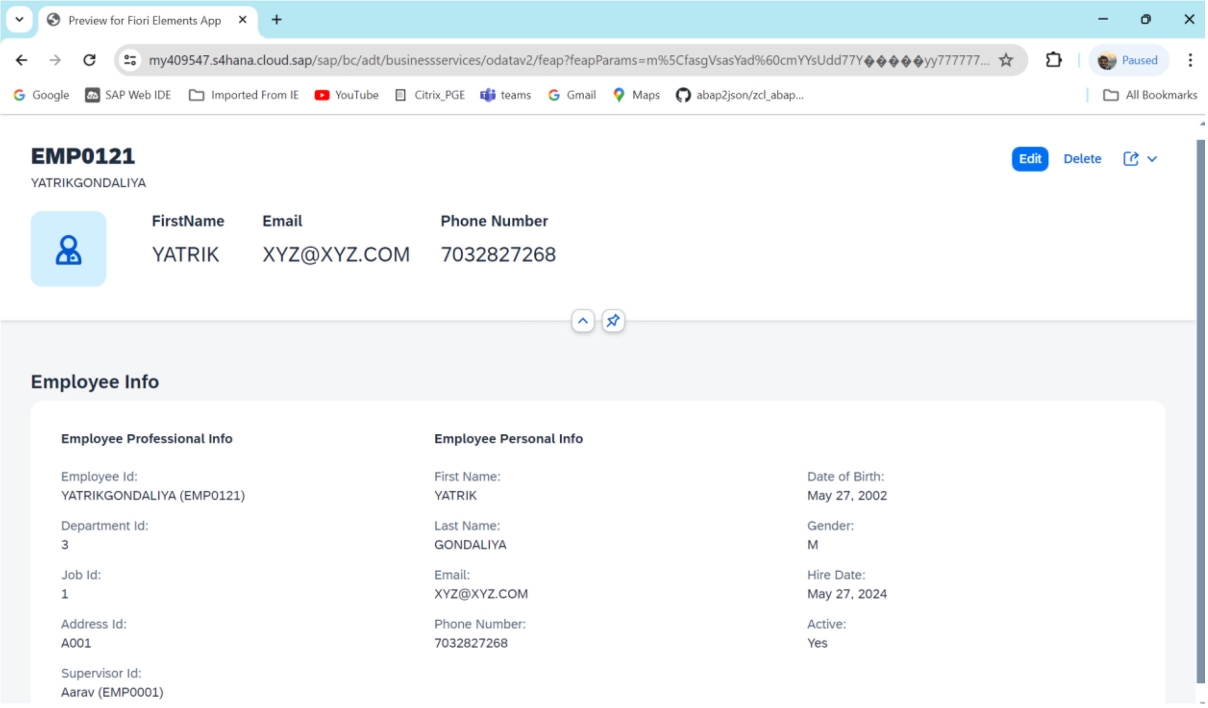
Employee Id:

Department Id:

Job Id:

Employees (124)

Employee Id	First Name	Last Name	Email	Phone Number	Date of Birth	Gender	Hire Date	Active
Mr. KALP EMP0125	Mr. KALP		KALP@GMAIL.COM	1243546244	May 30, 2000	M	May 23, 2024	Yes
KEVALPATEL EMP0124	KEVAL	PATEL	ABC@GMAIL.COM	7832194712	Apr 13, 1993	M		Yes
Mr. DNNK EMP0123	Mr. DN	NK	DEEP@GMAIL.COM	8848168292	Sep 20, 2002	M		Yes
YATRIKGONDALIYA EMP0121	YATRIK	GONDALIYA	XYZ@XYZ.COM	7032827268	May 27, 2002	M	May 27, 2024	Yes
LavanyaSharma EMP0120	Lavanya	Sharma	lavanya.sharma@gmail.com	8765432198	Aug 22, 1990	F	Mar 20, 2020	No
KunalNair EMP0119	Kunal	Nair	kunal.nair@gmail.com	7654321897	Apr 15, 1985	M	Jan 15, 2020	No
JayaPatel EMP0118	Jaya	Patel	jaya.patel@gmail.com	9876543210	Jul 22, 1987	F	Jan 5, 2022	No


Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav2/feap?feapParams=m%5CfasgVsasYad%60cmYYsUdd77Y����yy777777...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks
Edit
Delete

EMP0121

YATRIKGONDALIYA

FirstName	Email	Phone Number
YATRIK	XYZ@XYZ.COM	7032827268

Employee Info

Employee Professional Info

Employee Id:
YATRIKGONDALIYA (EMP0121)

Department Id:
3

Job Id:
1

Address Id:
A001

Supervisor Id:
Aarav (EMP0001)

Employee Personal Info

First Name:
YATRIK

Last Name:
GONDALIYA

Email:
XYZ@XYZ.COM

Phone Number:
7032827268

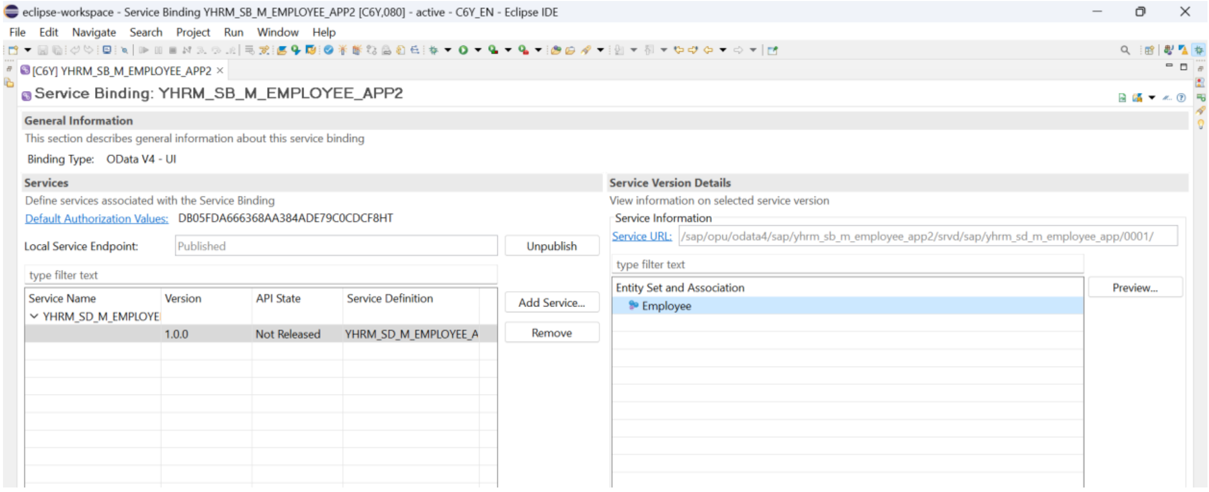
Date of Birth:
May 27, 2002

Gender:
M

Hire Date:
May 27, 2024

Active:
Yes

YHRM_SB_M_EMPLOYEE_APP2 (Business Services / Service Binding)


eclipse-workspace - Service Binding YHRM_SB_M_EMPLOYEE_APP2 [C6Y,080] - active - C6Y_EN - Eclipse IDE
File
Edit
Navigate
Search
Project
Run
Window
Help
[C6Y] YHRM_SB_M_EMPLOYEE_APP2
Unpublish
Add Service...
Remove
Preview...
type filter text
type filter text

Service Binding: YHRM_SB_M_EMPLOYEE_APP2

General Information

This section describes general information about this service binding

Binding Type: OData V4 - UI

Services

Define services associated with the Service Binding

Default Authorization Values: DB05FDA666368AA384ADE79C0CDCF8HT

Local Service Endpoint: Published

Service Name	Version	API State	Service Definition
YHRM_SD_M_EMPLOYE			
	1.0.0	Not Released	YHRM_SD_M_EMPLOYEE_A

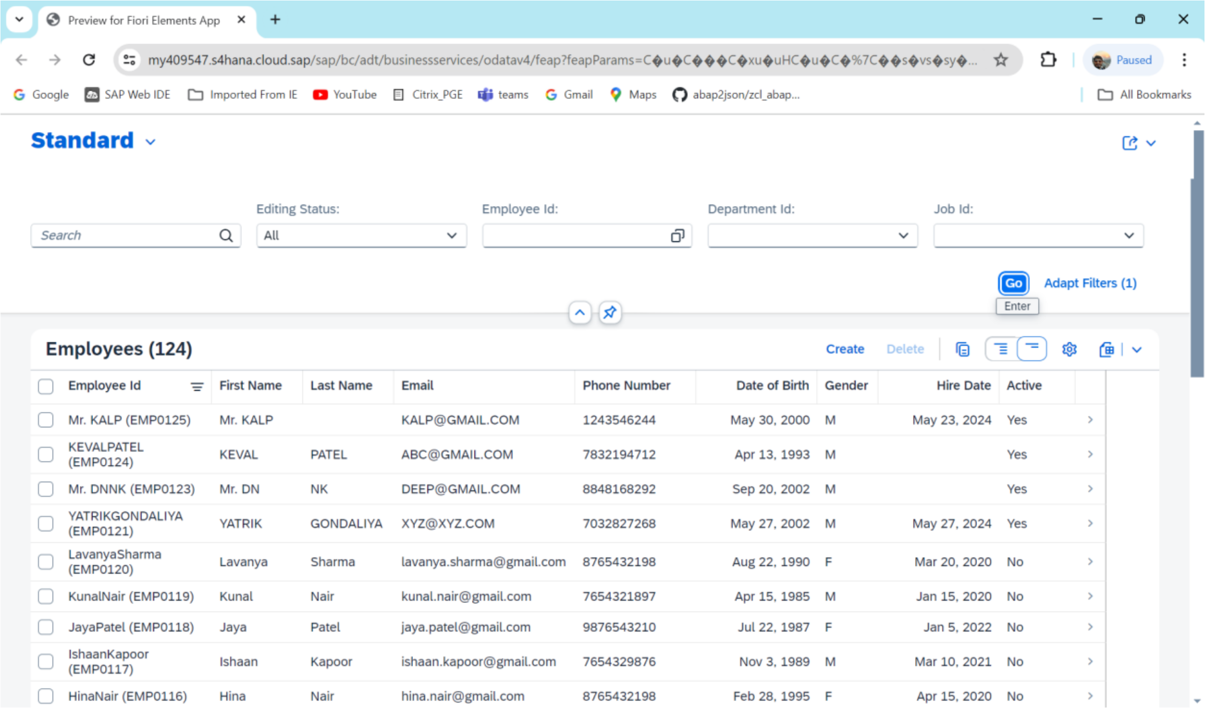
Service Version Details

View information on selected service version

Service Information

Service URL: /sap/opu/odata4/sap/yhrm_sb_m_employee_app2/srvd/sap/yhrm_sd_m_employee_app/0001/

Entity Set and Association
Employee

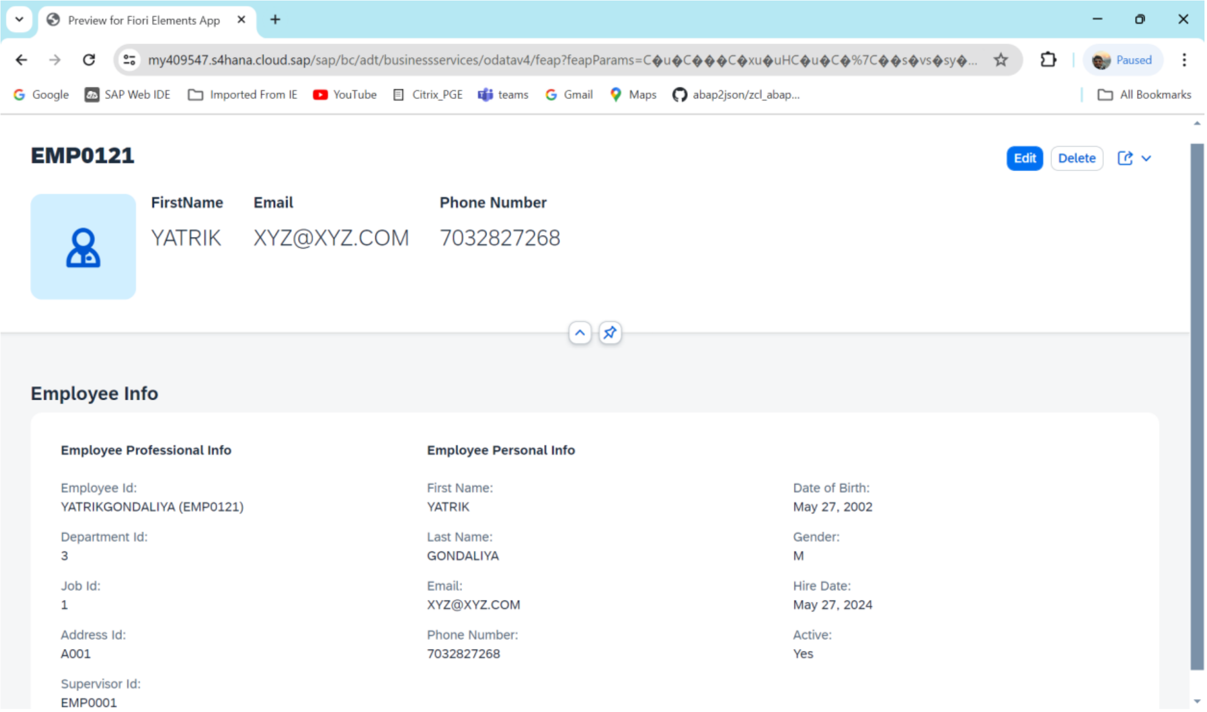

Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks
Go
Enter
Adapt Filters (1)
Create
Delete

Standard

Search	Editing Status:	Employee Id:	Department Id:	Job Id:
	All			

Employees (124)

Employee Id	First Name	Last Name	Email	Phone Number	Date of Birth	Gender	Hire Date	Active
Mr. KALP (EMP0125)	Mr. KALP		KALP@GMAIL.COM	1243546244	May 30, 2000	M	May 23, 2024	Yes
KEVALPATEL (EMP0124)	KEVAL	PATEL	ABC@GMAIL.COM	7832194712	Apr 13, 1993	M		Yes
Mr. DNNK (EMP0123)	Mr. DN	NK	DEEP@GMAIL.COM	8848168292	Sep 20, 2002	M		Yes
YATRIKGONDALIYA (EMP0121)	YATRIK	GONDALIYA	XYZ@XYZ.COM	7032827268	May 27, 2002	M	May 27, 2024	Yes
LavanyaSharma (EMP0120)	Lavanya	Sharma	lavanya.sharma@gmail.com	8765432198	Aug 22, 1990	F	Mar 20, 2020	No
KunalNair (EMP0119)	Kunal	Nair	kunal.nair@gmail.com	7654321897	Apr 15, 1985	M	Jan 15, 2020	No
JayaPatel (EMP0118)	Jaya	Patel	jaya.patel@gmail.com	9876543210	Jul 22, 1987	F	Jan 5, 2022	No
IshaanKapoor (EMP0117)	Ishaan	Kapoor	ishaan.kapoor@gmail.com	7654329876	Nov 3, 1989	M	Mar 10, 2021	No
HinaNair (EMP0116)	Hina	Nair	hina.nair@gmail.com	8765432198	Feb 28, 1995	F	Apr 15, 2020	No


Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks
Edit
Delete

EMP0121

FirstName	Email	Phone Number
YATRIK	XYZ@XYZ.COM	7032827268

Employee Info

Employee Professional Info

Employee Id:
YATRIKGONDALIYA (EMP0121)

Department Id:
3

Job Id:
1

Address Id:
A001

Supervisor Id:
EMP0001

Employee Personal Info

First Name:
YATRIK

Last Name:
GONDALIYA

Email:
XYZ@XYZ.COM

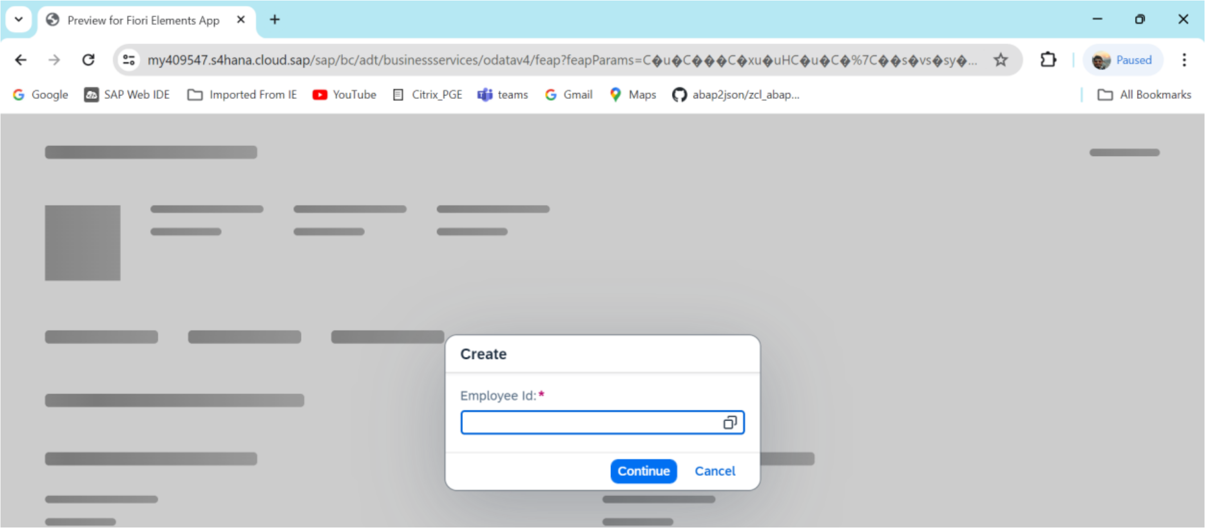
Phone Number:
7032827268

Date of Birth:
May 27, 2002

Gender:
M

Hire Date:
May 27, 2024

Active:
Yes


Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks

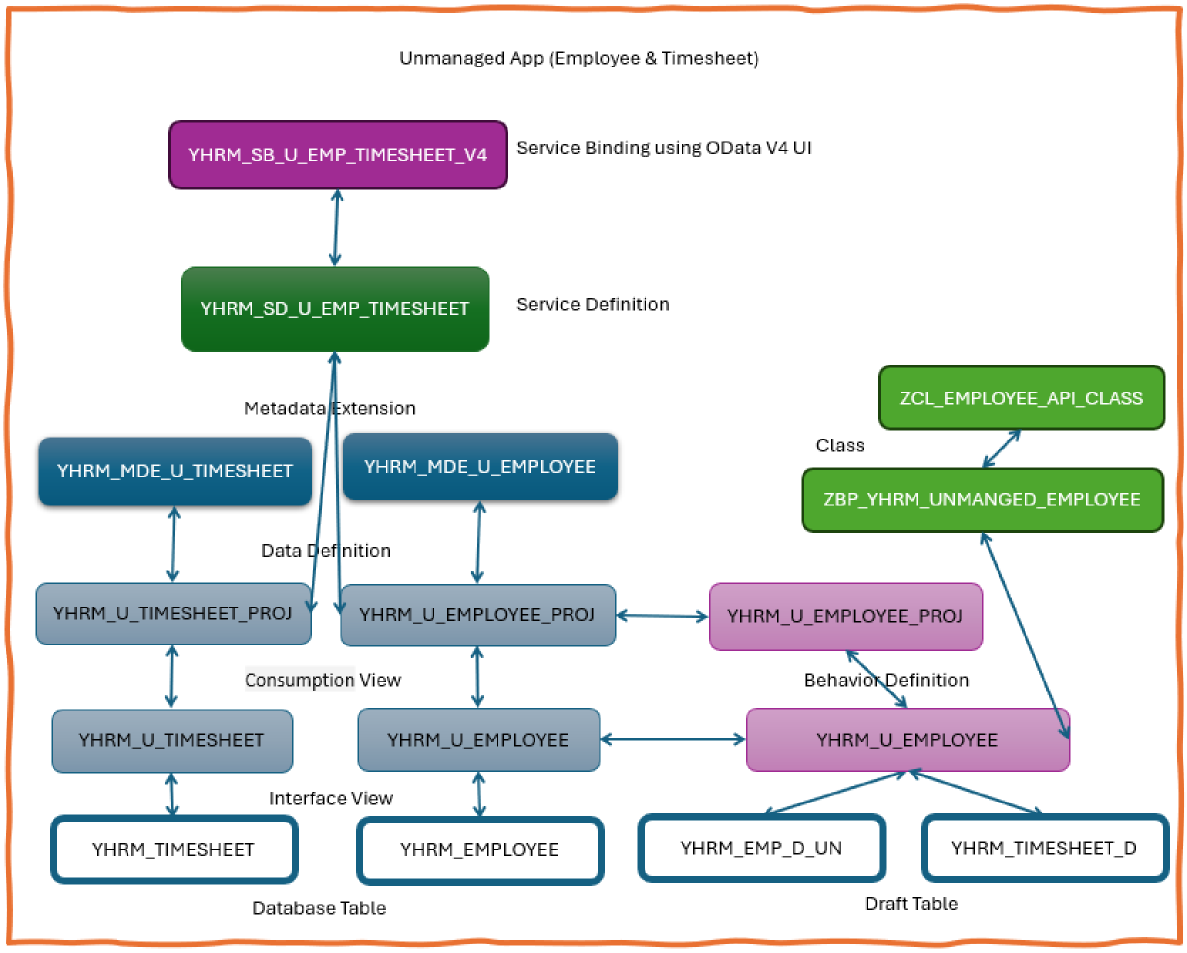
Create

Employee Id:*

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.

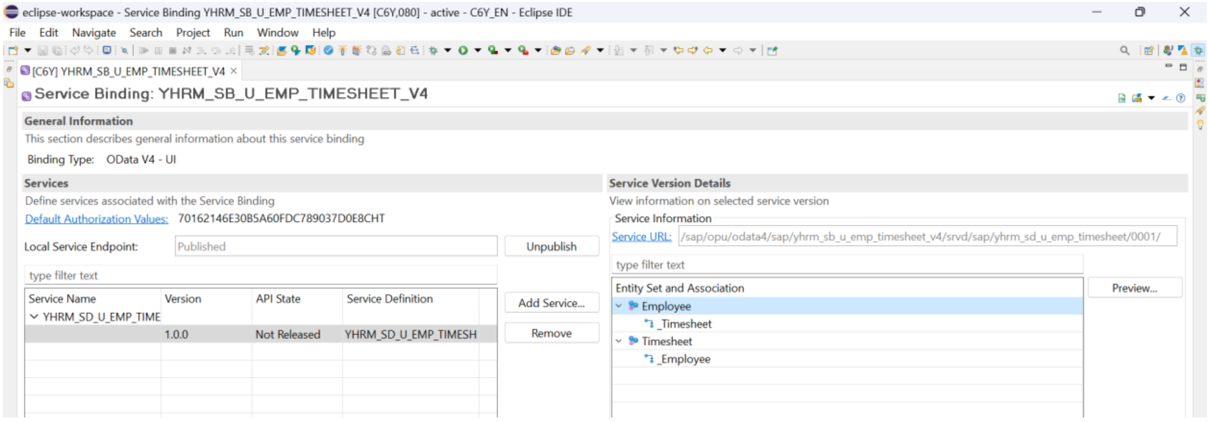
```
METHOD cba_timesheet.
  gt_timesheet = VALUE #(
    FOR ls_entities_cba IN entities_cba
```

```
        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)



eclipse-workspace - Service Binding YHRM_SB_U_EMP_TIMESHEET_V4 [C6Y,080] - active - C6Y_EN - Eclipse IDE

[C6Y] YHRM_SB_U_EMP_TIMESHEET_V4

Service Binding: YHRM_SB_U_EMP_TIMESHEET_V4

General Information

This section describes general information about this service binding

Binding Type: OData V4 - UI

Services

Define services associated with the Service Binding

Default Authorization Values: 70162146E30B5A60FDC789037D0E8CHT

Local Service Endpoint: Published

Service Name	Version	API State	Service Definition
YHRM_SD_U_EMP_TIME			
	1.0.0	Not Released	YHRM_SD_U_EMP_TIMESH

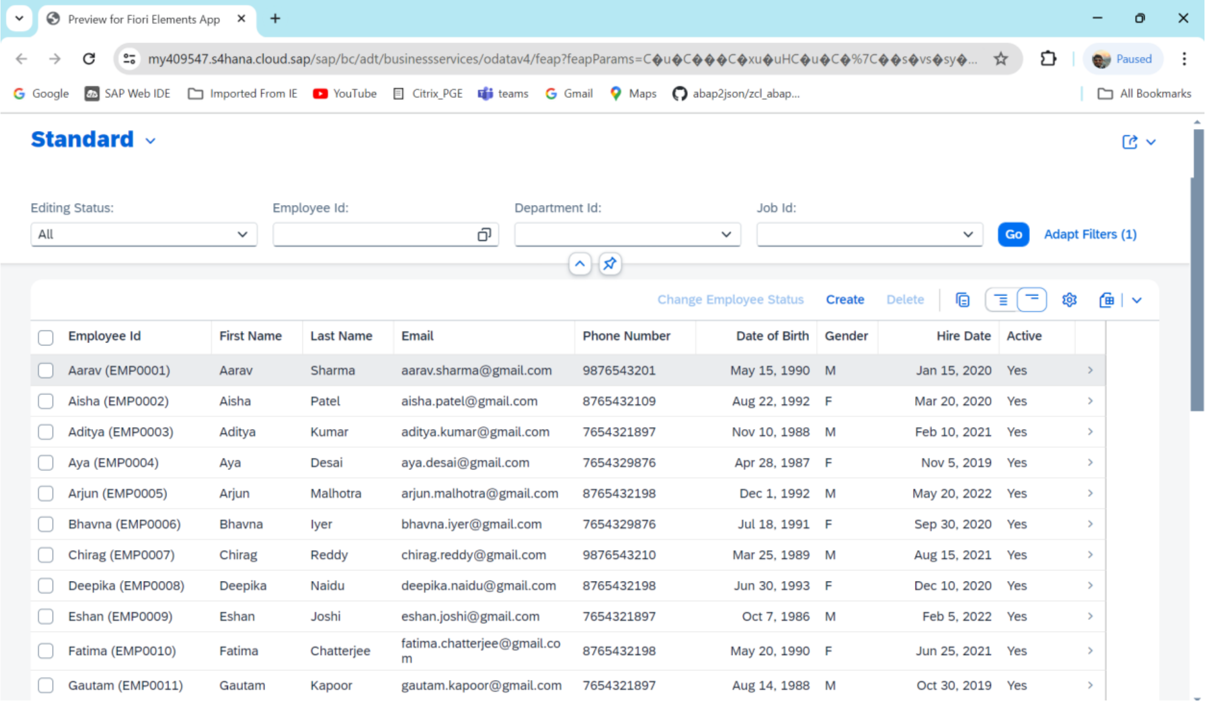
Service Version Details

View information on selected service version

Service Information

Service URL: /sap/opu/odata4/sap/yhrm_sb_u_emp_timesheet_v4/srvd/sap/yhrm_sd_u_emp_timesheet/0001/

Entity Set and Association
- Employee
  - _Timesheet
- Timesheet
  - _Employee



Preview for Fiori Elements App

my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...

Paused
Google SAP Web IDE Imported From IE YouTube Citrix_PGE teams Gmail Maps abap2json/zcl_abap... All Bookmarks

Standard

Editing Status: All

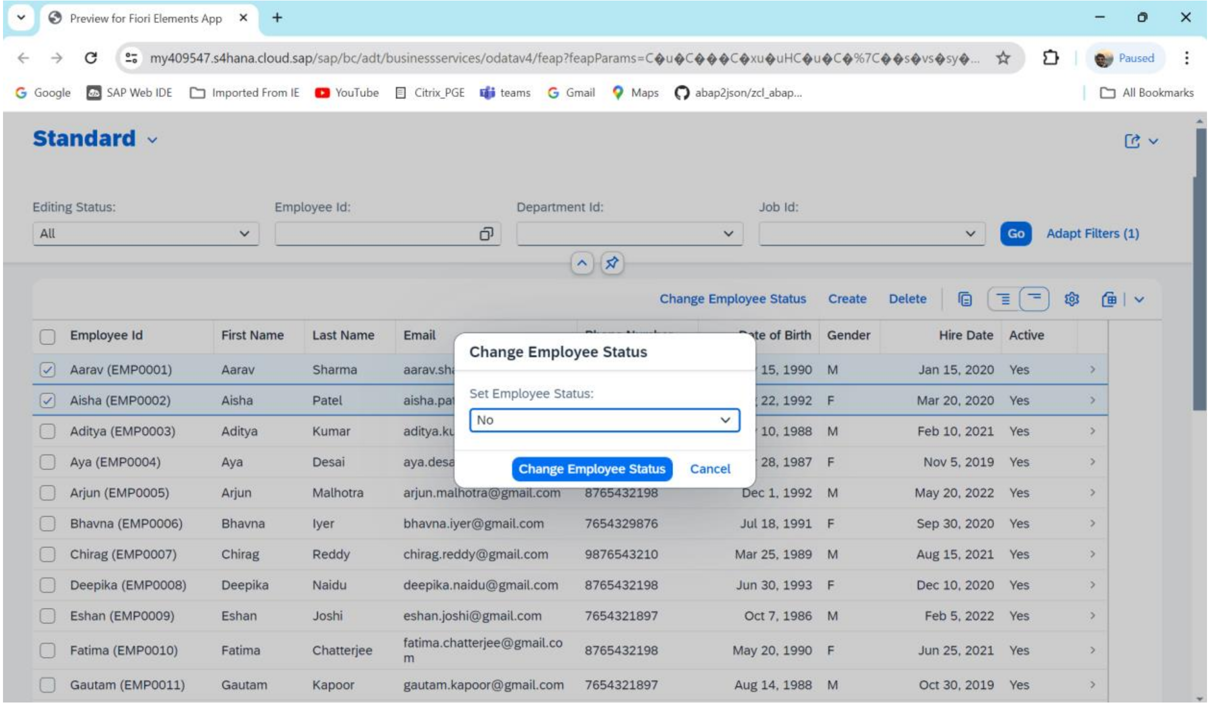
Employee Id:

Department Id:

Job Id:

Go
Adapt Filters (1)
Change Employee Status
Create
Delete

	Employee Id	First Name	Last Name	Email	Phone Number	Date of Birth	Gender	Hire Date	Active
☐	Aarav (EMP0001)	Aarav	Sharma	aarav.sharma@gmail.com	9876543201	May 15, 1990	M	Jan 15, 2020	Yes
☐	Aisha (EMP0002)	Aisha	Patel	aisha.patel@gmail.com	8765432109	Aug 22, 1992	F	Mar 20, 2020	Yes
☐	Aditya (EMP0003)	Aditya	Kumar	aditya.kumar@gmail.com	7654321897	Nov 10, 1988	M	Feb 10, 2021	Yes
☐	Aya (EMP0004)	Aya	Desai	aya.desai@gmail.com	7654329876	Apr 28, 1987	F	Nov 5, 2019	Yes
☐	Arjun (EMP0005)	Arjun	Malhotra	arjun.malhotra@gmail.com	8765432198	Dec 1, 1992	M	May 20, 2022	Yes
☐	Bhavna (EMP0006)	Bhavna	Iyer	bhavna.iyer@gmail.com	7654329876	Jul 18, 1991	F	Sep 30, 2020	Yes
☐	Chirag (EMP0007)	Chirag	Reddy	chirag.reddy@gmail.com	9876543210	Mar 25, 1989	M	Aug 15, 2021	Yes
☐	Deepika (EMP0008)	Deepika	Naidu	deepika.naidu@gmail.com	8765432198	Jun 30, 1993	F	Dec 10, 2020	Yes
☐	Eshan (EMP0009)	Eshan	Joshi	eshan.joshi@gmail.com	7654321897	Oct 7, 1986	M	Feb 5, 2022	Yes
☐	Fatima (EMP0010)	Fatima	Chatterjee	fatima.chatterjee@gmail.com	8765432198	May 20, 1990	F	Jun 25, 2021	Yes
☐	Gautam (EMP0011)	Gautam	Kapoor	gautam.kapoor@gmail.com	7654321897	Aug 14, 1988	M	Oct 30, 2019	Yes


Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks

Standard

Editing Status: All

Employee Id:

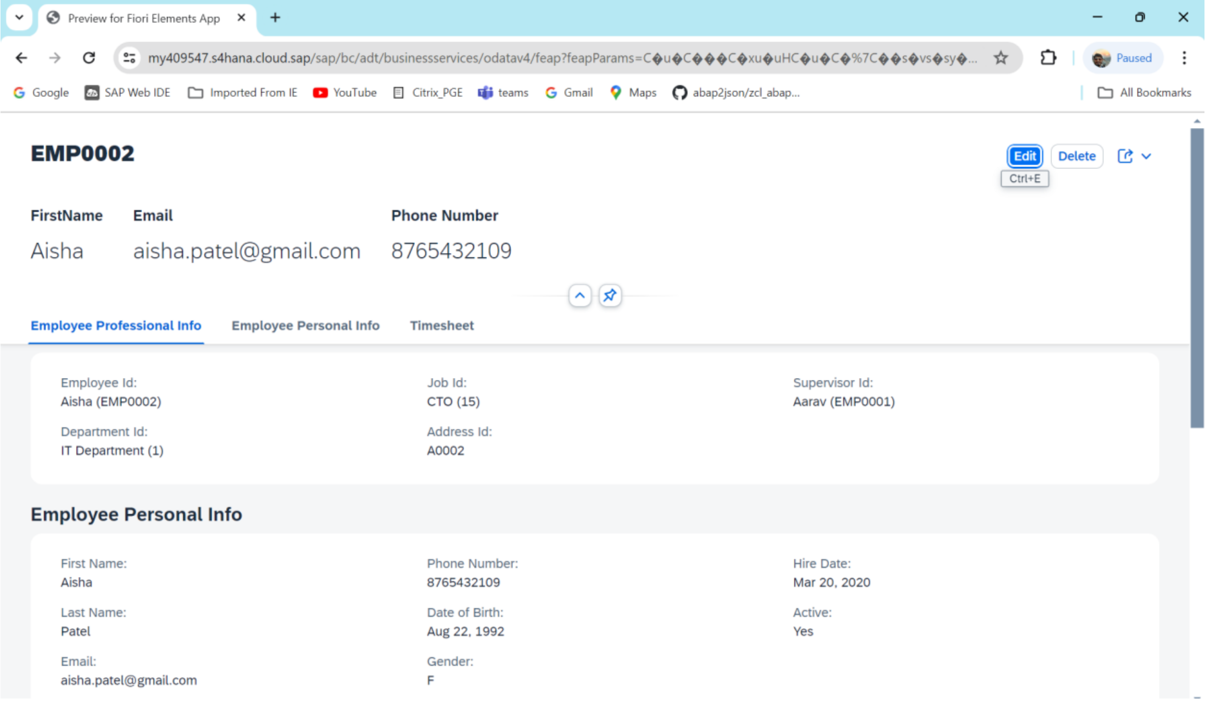
Department Id:

Job Id:

	Employee Id	First Name	Last Name	Email	Phone Number	Date of Birth	Gender	Hire Date	Active
☑	Aarav (EMP0001)	Aarav	Sharma	aarav.sh		15, 1990	M	Jan 15, 2020	Yes
☑	Aisha (EMP0002)	Aisha	Patel	aisha.pa		22, 1992	F	Mar 20, 2020	Yes
□	Aditya (EMP0003)	Aditya	Kumar	aditya.ku		10, 1988	M	Feb 10, 2021	Yes
□	Aya (EMP0004)	Aya	Desai	aya.desa		28, 1987	F	Nov 5, 2019	Yes
□	Arjun (EMP0005)	Arjun	Malhotra	arjun.malhotra@gmail.com	8765432198	Dec 1, 1992	M	May 20, 2022	Yes
□	Bhavna (EMP0006)	Bhavna	Iyer	bhavna.iyer@gmail.com	7654329876	Jul 18, 1991	F	Sep 30, 2020	Yes
□	Chirag (EMP0007)	Chirag	Reddy	chirag.reddy@gmail.com	9876543210	Mar 25, 1989	M	Aug 15, 2021	Yes
□	Deepika (EMP0008)	Deepika	Naidu	deepika.naidu@gmail.com	8765432198	Jun 30, 1993	F	Dec 10, 2020	Yes
□	Eshan (EMP0009)	Eshan	Joshi	eshan.joshi@gmail.com	7654321897	Oct 7, 1986	M	Feb 5, 2022	Yes
□	Fatima (EMP0010)	Fatima	Chatterjee	fatima.chatterjee@gmail.com	8765432198	May 20, 1990	F	Jun 25, 2021	Yes
□	Gautam (EMP0011)	Gautam	Kapoor	gautam.kapoor@gmail.com	7654321897	Aug 14, 1988	M	Oct 30, 2019	Yes

Change Employee Status

Set Employee Status: No


Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks
Edit
Ctrl+E
Delete

EMP0002

FirstName	Email	Phone Number
Aisha	aisha.patel@gmail.com	8765432109

Employee Id: Aisha (EMP0002)

Job Id: CTO (15)

Supervisor Id: Aarav (EMP0001)

Department Id: IT Department (1)

Address Id: A0002

Employee Personal Info

First Name: Aisha

Phone Number: 8765432109

Hire Date: Mar 20, 2020

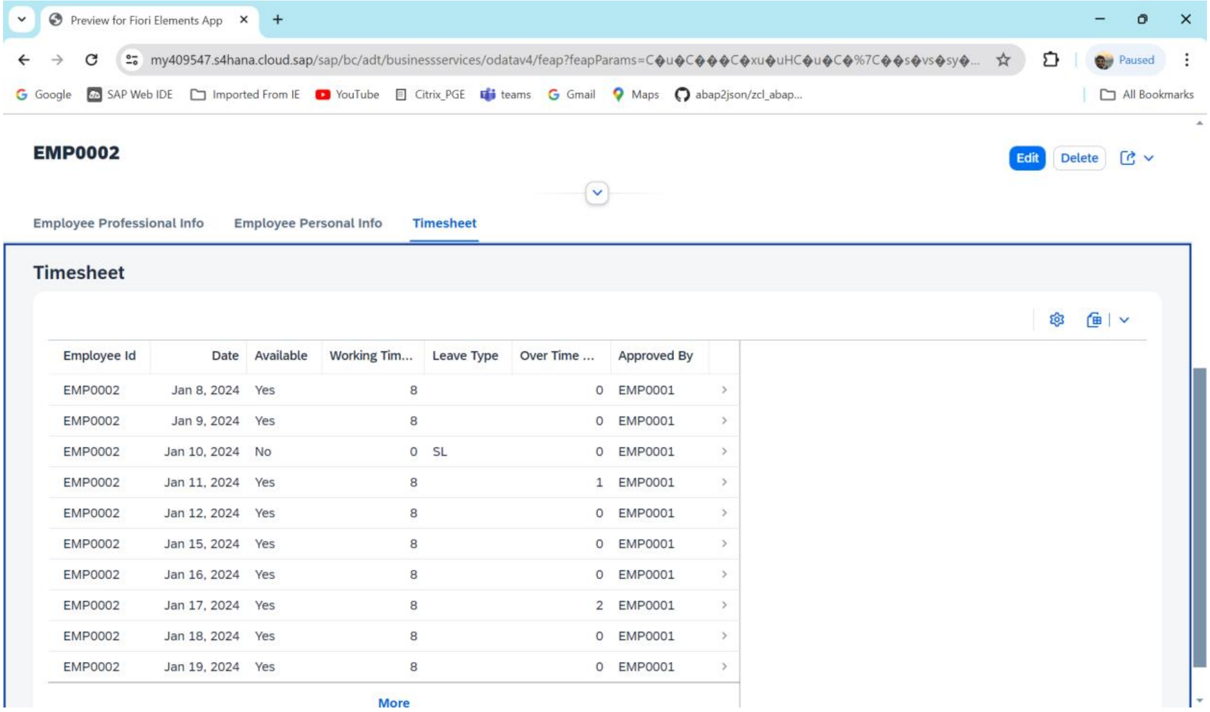
Last Name: Patel

Date of Birth: Aug 22, 1992

Active: Yes

Email: aisha.patel@gmail.com

Gender: F

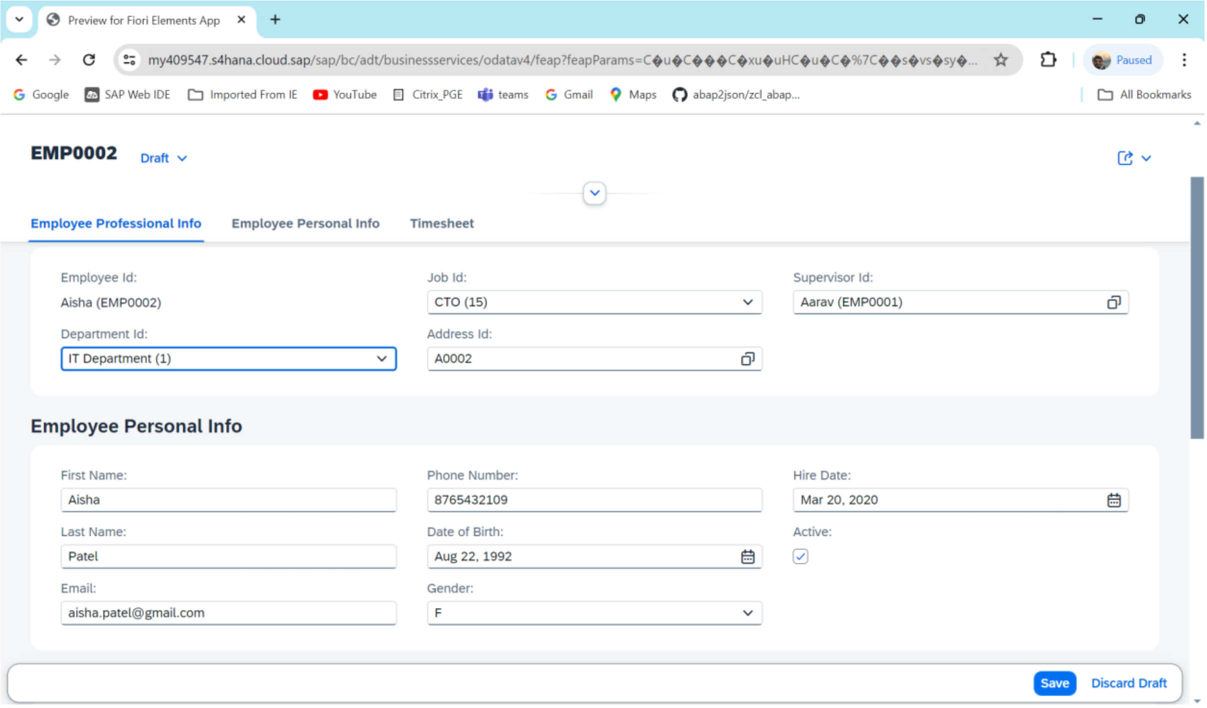

Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks
Edit
Delete
More

EMP0002

Employee Professional Info | Employee Personal Info | Timesheet

Timesheet

Employee Id	Date	Available	Working Tim...	Leave Type	Over Time ...	Approved By
EMP0002	Jan 8, 2024	Yes	8		0	EMP0001
EMP0002	Jan 9, 2024	Yes	8		0	EMP0001
EMP0002	Jan 10, 2024	No	0	SL	0	EMP0001
EMP0002	Jan 11, 2024	Yes	8		1	EMP0001
EMP0002	Jan 12, 2024	Yes	8		0	EMP0001
EMP0002	Jan 15, 2024	Yes	8		0	EMP0001
EMP0002	Jan 16, 2024	Yes	8		0	EMP0001
EMP0002	Jan 17, 2024	Yes	8		2	EMP0001
EMP0002	Jan 18, 2024	Yes	8		0	EMP0001
EMP0002	Jan 19, 2024	Yes	8		0	EMP0001


Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks
Save
Discard Draft

EMP0002 Draft

Employee Professional Info | Employee Personal Info | Timesheet

Employee Id: Aisha (EMP0002)

Job Id: CTO (15)

Supervisor Id: Aarav (EMP0001)

Department Id: IT Department (1)

Address Id: A0002

Employee Personal Info

First Name: Aisha

Phone Number: 8765432109

Hire Date: Mar 20, 2020

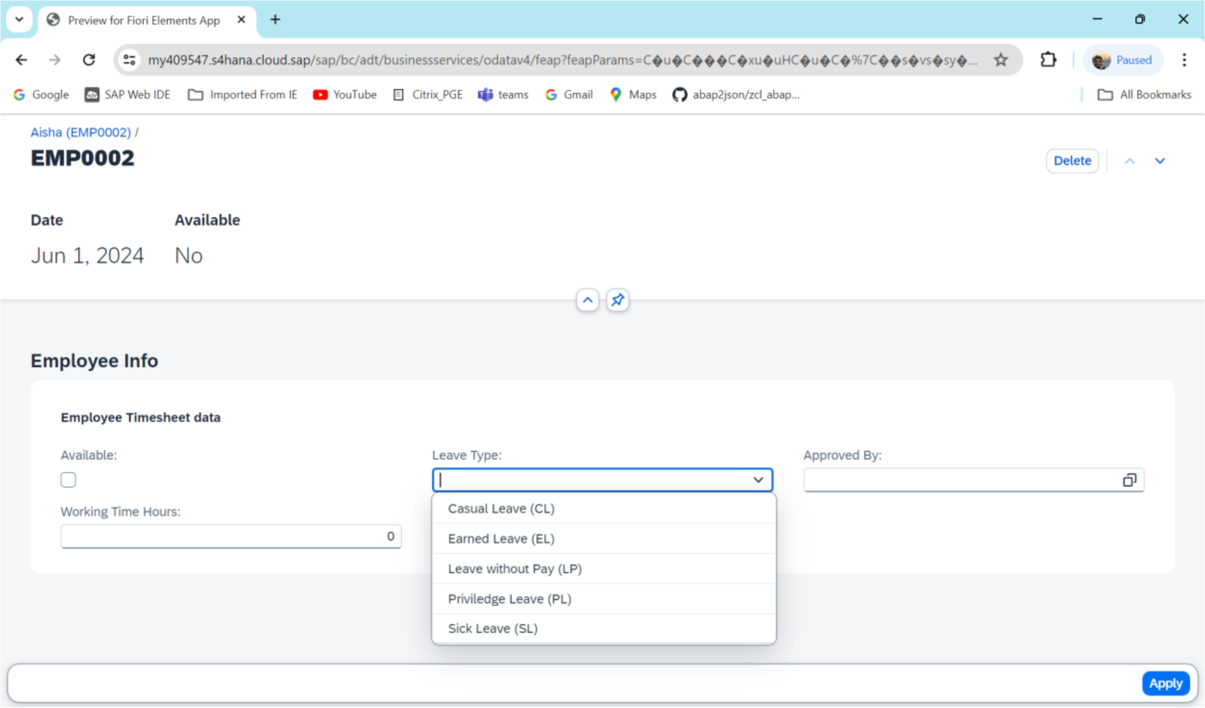
Last Name: Patel

Date of Birth: Aug 22, 1992

Active: ☑

Email: aisha.patel@gmail.com

Gender: F


Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks
Delete
Apply

Aisha (EMP0002) /

EMP0002

Date	Available
Jun 1, 2024	No

Employee Info

Employee Timesheet data

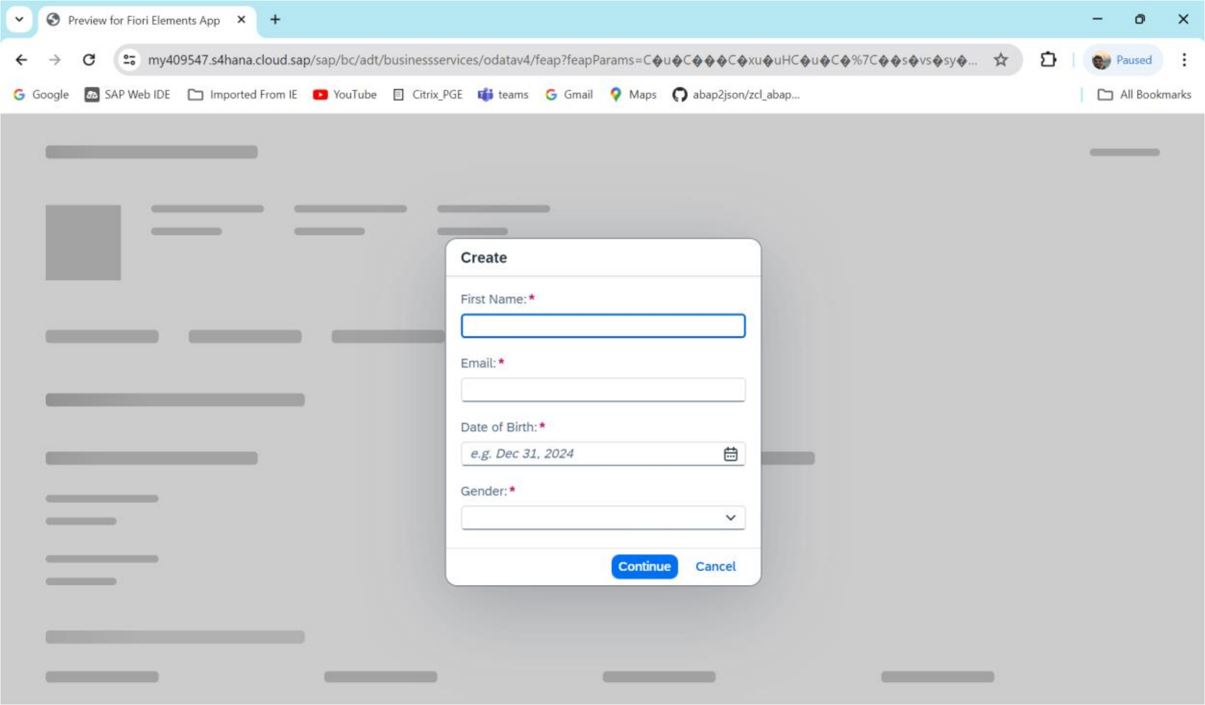
Available: □

Working Time Hours: 0

Leave Type:

- Casual Leave (CL)
- Earned Leave (EL)
- Leave without Pay (LP)
- Priviledge Leave (PL)
- Sick Leave (SL)

Approved By:


Preview for Fiori Elements App
my409547.s4hana.cloud.sap/sap/bc/adt/businessservices/odatav4/feap?feapParams=C�u�C���C�xu�uHC�u�C�%7C��s�vs�sy�...
Paused
Google
SAP Web IDE
Imported From IE
YouTube
Citrix_PGE
teams
Gmail
Maps
abap2json/zcl_abap...
All Bookmarks
Continue
Cancel

Create

First Name:*

Email:*

Date of Birth:* e.g. Dec 31, 2024

Gender:*

Important Links:

Unmanaged RAP Model (Student Example)

RAP Model CodeInMins YouTube

SAP TECHNOMANIAC YouTube (travel booking example)

Managed Scenario Travel Booking Complete Example

Unmanaged Scenario Travel Booking Example part 1

Unmanaged Scenario Travel Booking Example part 2

Unmanaged Scenario Travel Booking Example part 3