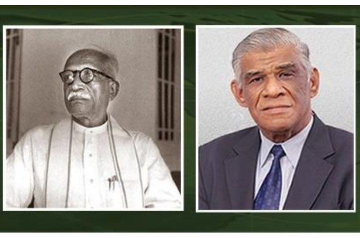


Sprint-2 Plan

Employee Detail Management

1. Establishing Project Folder Architecture
2. Server App Initialization
3. **Sprint-2 Execution** [2(a), **2(b)**, 2(c), 2(d), 2(e), 2(f)]
4. Completing Sprint-1 Objectives

2(a) Employee Module - Analysis, Design & DB-Preparation

Data Design, UI Design, Sample Data, Database Preparation

2(b) Employee View

1. Controller Design
2. Coding-Server App
    - Version-1 (Client Pagination)
        - Entity Coding, DAO Coding, Controller Coding, Testing
    - Version-2 (Server Pagination)
        - Controller Modification, Testing
3. Coding-Client App
    - Version-1 (Client Pagination)
        - UI Preparation, Entity Coding, Service Coding, Controller Coding, Testing
    - Version-2 (Server Pagination)
        - Service Modification, Controller Modification, Testing

2(c) Employee Search
2(d) Employee Insert
2(e) Employee Update
2(f) Employee Delete

==

2(b) Employee View

1. Controller Design
2. Coding-Server App

Version-1 (Client Pagination)

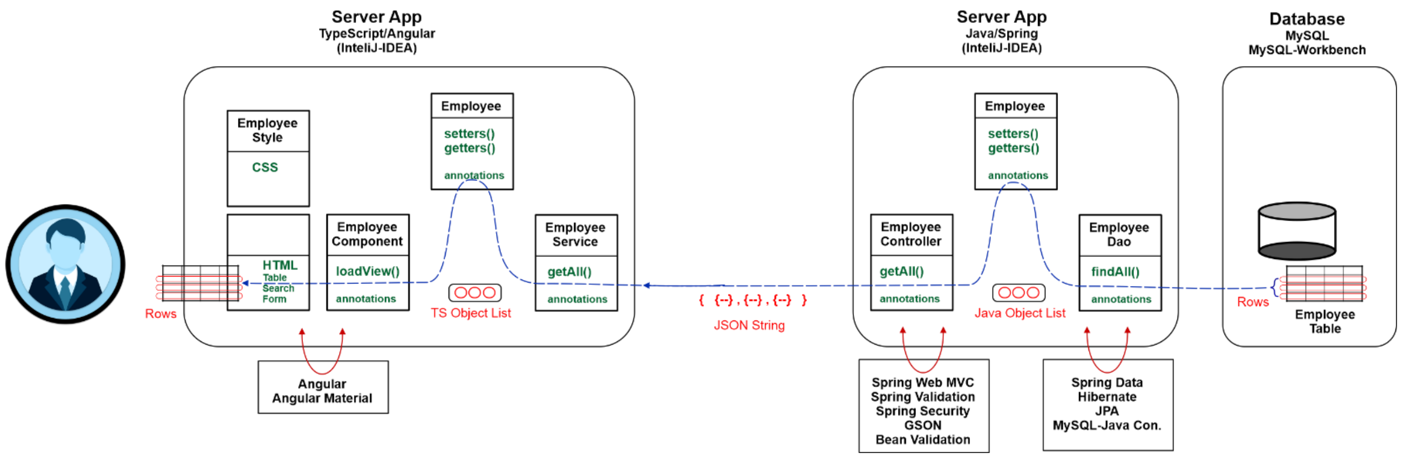
1. Entity Coding,
2. DAO Coding,
3. Controller Coding,
4. Testing

Version-2 (Server Pagination)

==

1. Controller Design

(Modified Sequence/Communication Diagram)



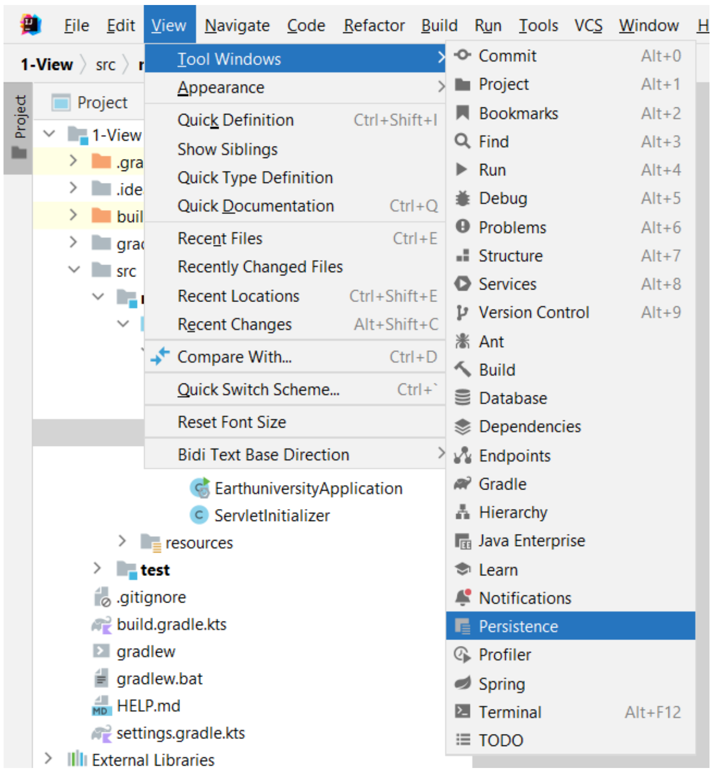
Server App	Client App
1. **Employee.java** → setters(), getters(), annotations **Gender.java** **Designation.java** **Employeestatus.java** 2. **EmployeeDao.java** → findAll(), annotations 3. **EmployeeController.java** → getAll(), annotations	1. employee-component.html 2. employee-componnet.css 3. EmployeeComponent.ts → loadView(), annotations 4. Employee.ts → setters(), getters(), annotations 5. EmployeeService.ts → getAll(), annotations

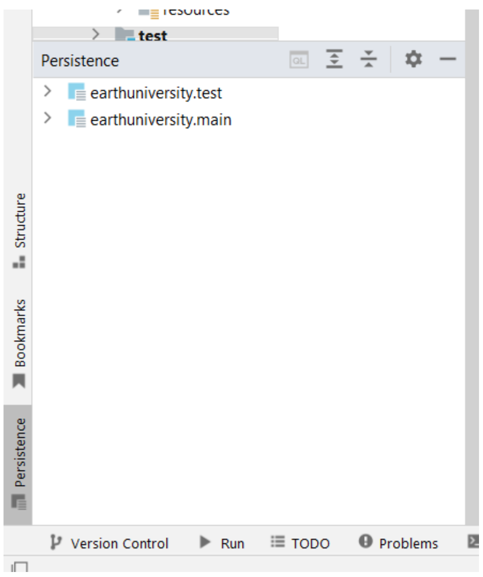
2. Coding Server-App

(1) Entity Coding
(2) Dao Coding
(3) Controller Coding
(4) Testing Employee Get Service

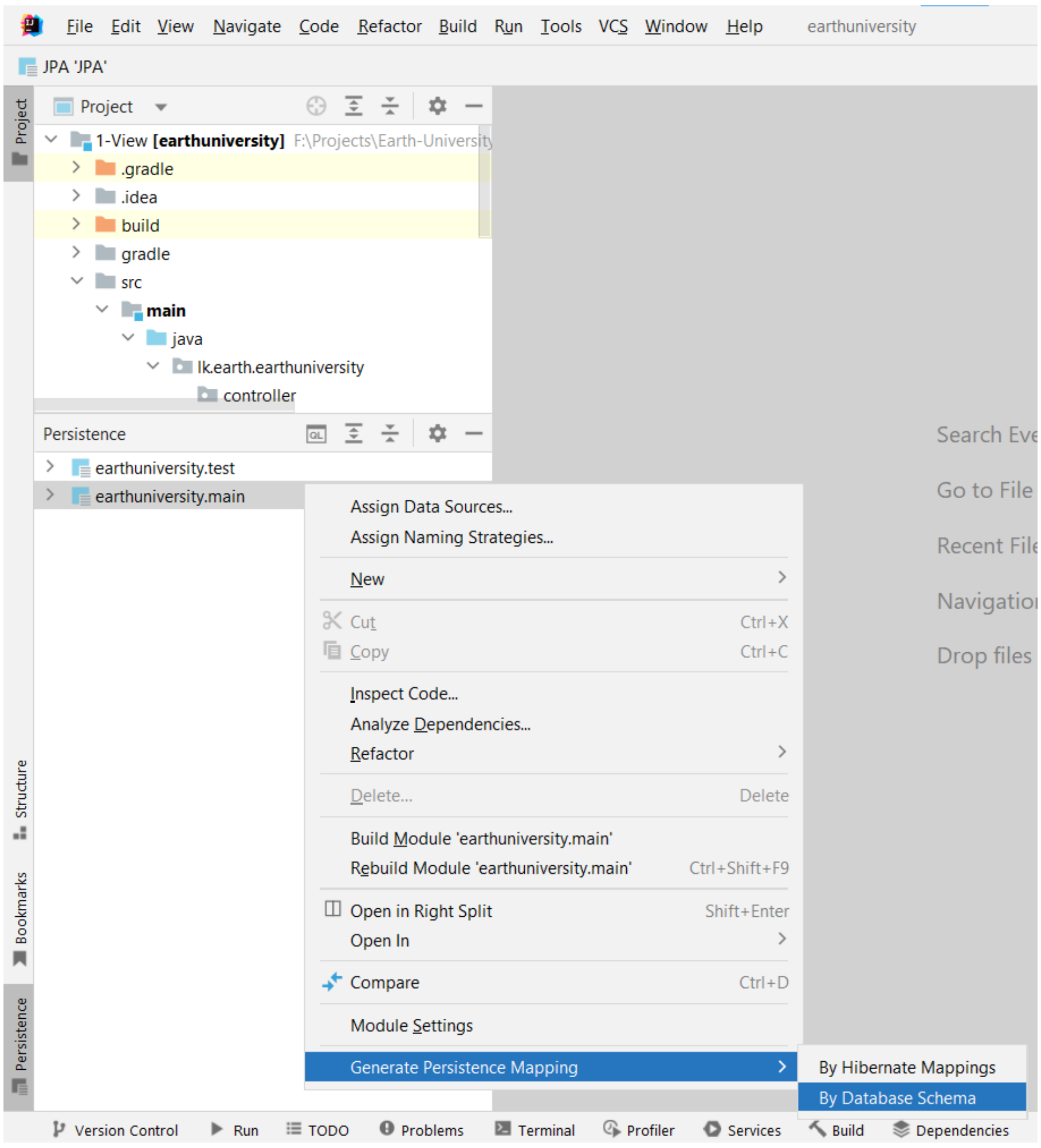
2.1 Entity Coding

(a) Enable Persistence Tool

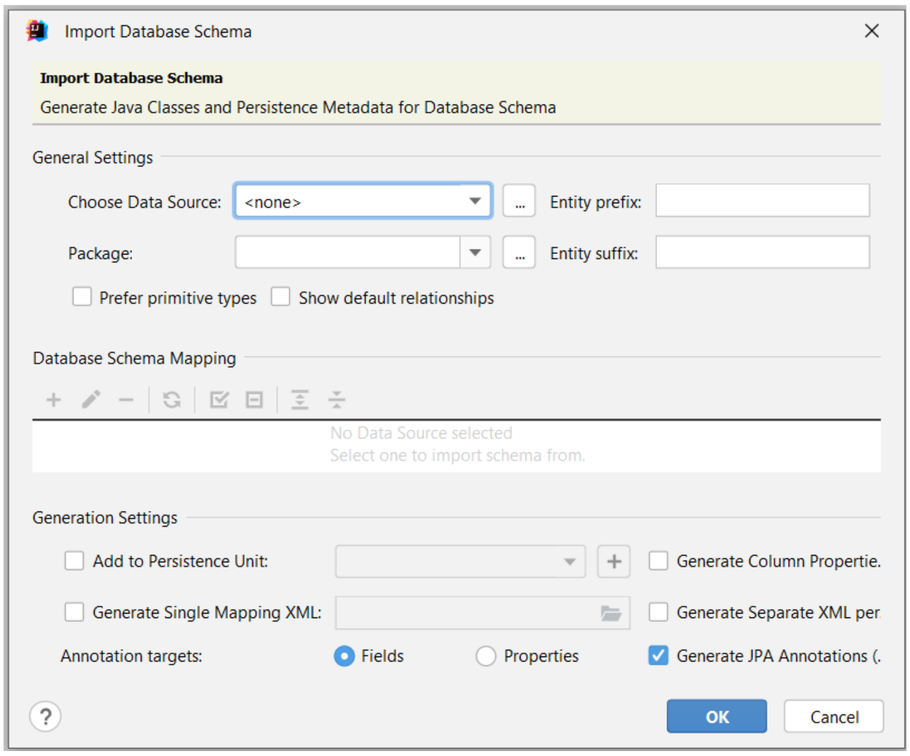




(b) Generate Persistence Mapping



(c) Select/Create Data Sources


Import Database Schema

Import Database Schema

Generate Java Classes and Persistence Metadata for Database Schema

General Settings

Choose Data Source: <none>

Entity prefix:

Package:

Entity suffix:

☐ Prefer primitive types ☐ Show default relationships

Database Schema Mapping

No Data Source selected

Select one to import schema from.

Generation Settings

☐ Add to Persistence Unit:

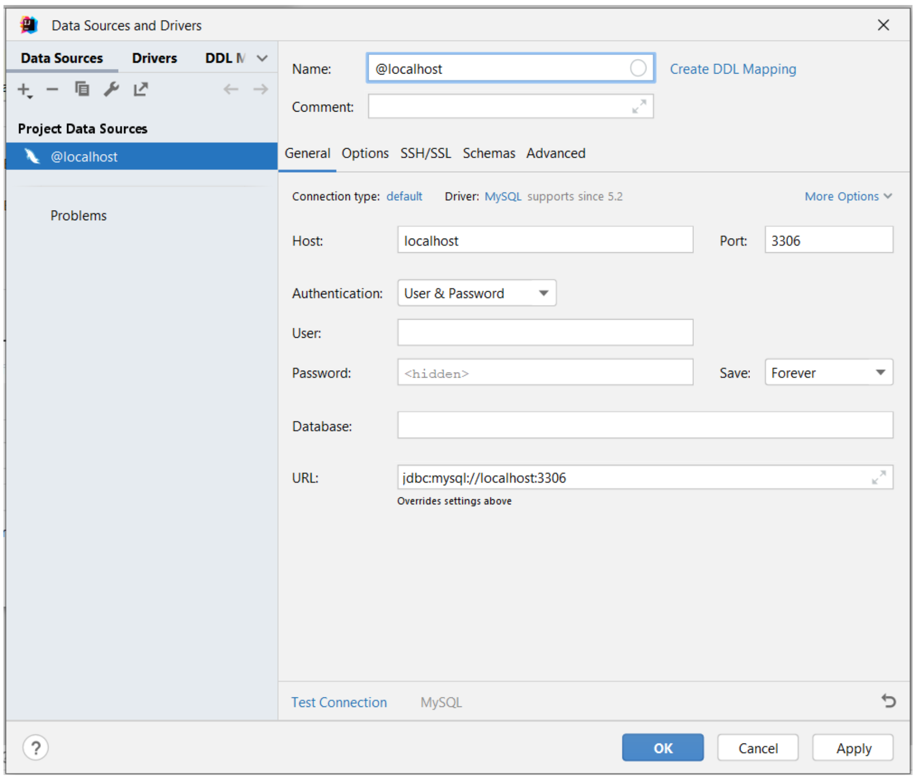
☐ Generate Column Propertie.

☐ Generate Single Mapping XML:

☐ Generate Separate XML per

Annotation targets: [✓ Fields] Properties

✓ Generate JPA Annotations (.


Data Sources and Drivers

Data Sources | Drivers | DDL M

Project Data Sources

@localhost

Problems

Name: @localhost

Create DDL Mapping

Comment:

General | Options | SSH/SSL | Schemas | Advanced

Connection type: default Driver: MySQL supports since 5.2

More Options

Host: localhost Port: 3306

Authentication: User & Password

User:

Password: <hidden> Save: Forever

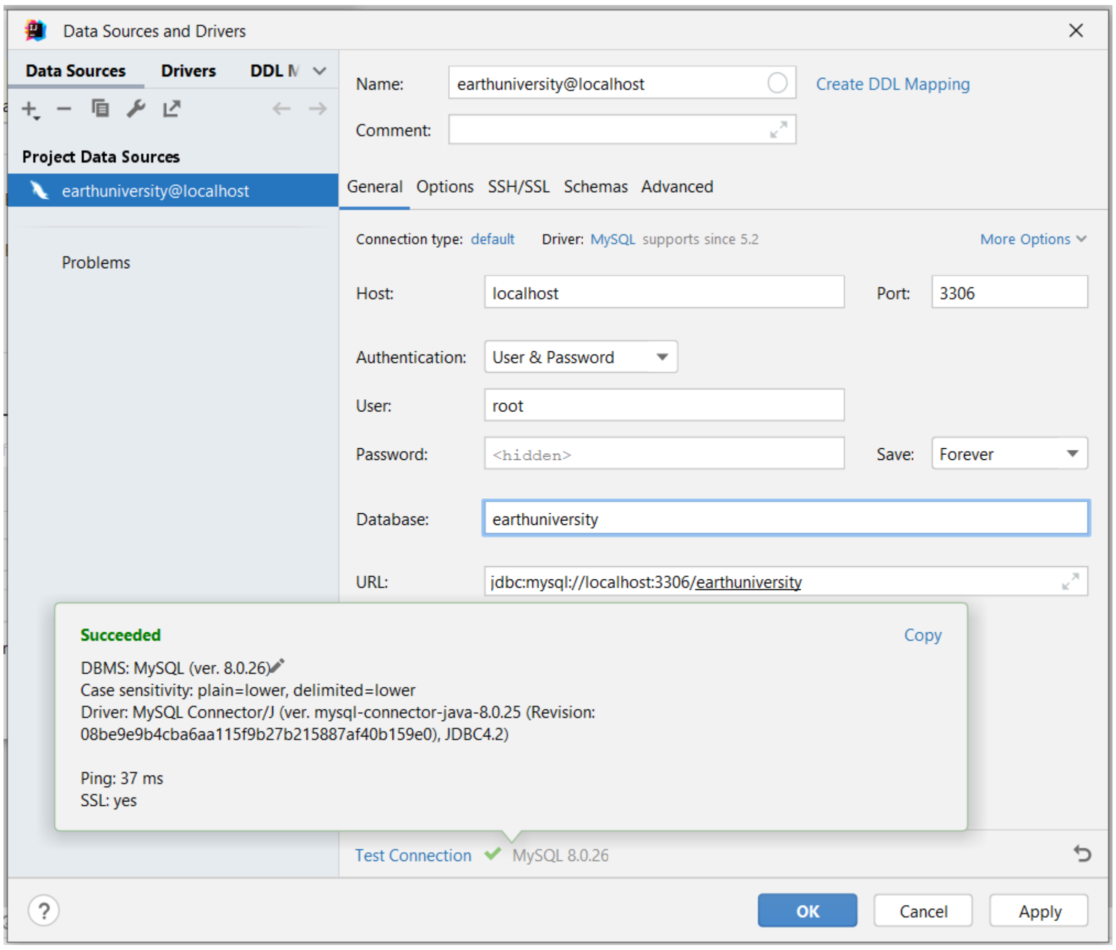
Database:

URL: jdbc:mysql://localhost:3306

Overrides settings above

When you do not have downloaded MySQL-Connector earlier, this window will have a link to download the "java-mysql-connector.jar". You must first download the driver.

Enter Username(root), Password and the Database Name ("earthuniversity") and Test The Connection



Data Sources and Drivers

Data Sources | Drivers | DDL M

Project Data Sources

earthuniversity@localhost

Problems

Name: earthuniversity@localhost | Create DDL Mapping

Comment:

General | Options | SSH/SSL | Schemas | Advanced

Connection type: default | Driver: MySQL supports since 5.2 | More Options

Host: localhost | Port: 3306

Authentication: User & Password

User: root

Password: <hidden> | Save: Forever

Database: earthuniversity

URL: jdbc:mysql://localhost:3306/earthuniversity

Succeeded | Copy

DBMS: MySQL (ver. 8.0.26)
Case sensitivity: plain=lower, delimited=lower
Driver: MySQL Connector/J (ver. mysql-connector-java-8.0.25 (Revision: 08be9e9b4cba6aa115f9b27b215887af40b159e0), JDBC4.2)

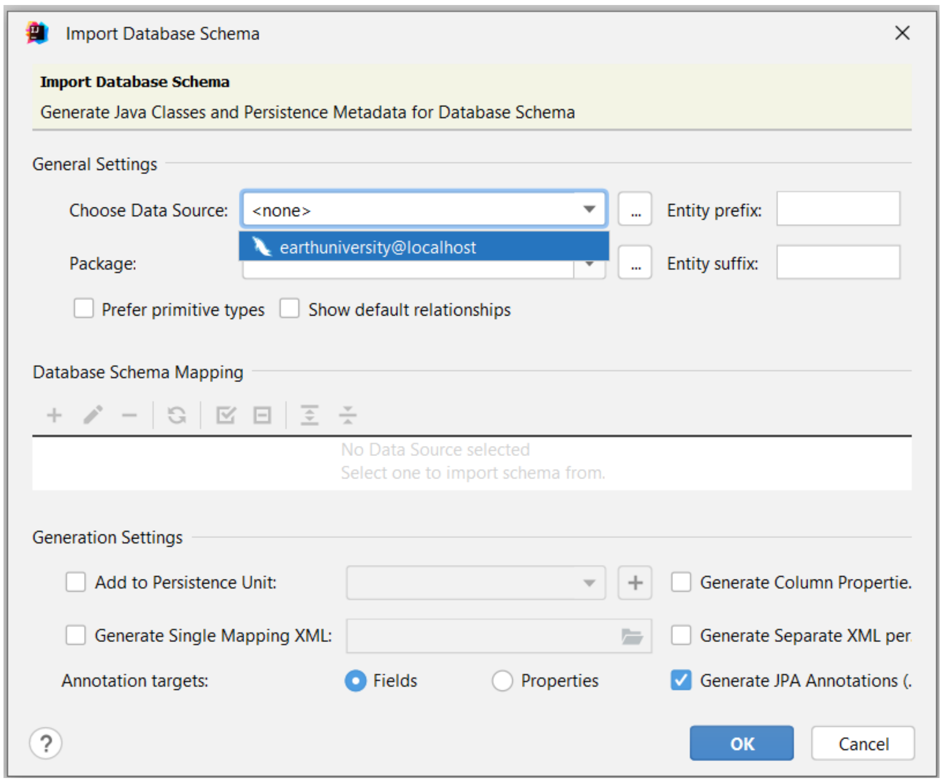
Ping: 37 ms
SSL: yes

Test Connection ✓ MySQL 8.0.26

OK | Cancel | Apply

Click Apply and OK

Select the Data Source



Import Database Schema

Import Database Schema

Generate Java Classes and Persistence Metadata for Database Schema

General Settings

Choose Data Source: <none> | Entity prefix:

earthuniversity@localhost

Package: | Entity suffix:

☐ Prefer primitive types ☐ Show default relationships

Database Schema Mapping

No Data Source selected
Select one to import schema from.

Generation Settings

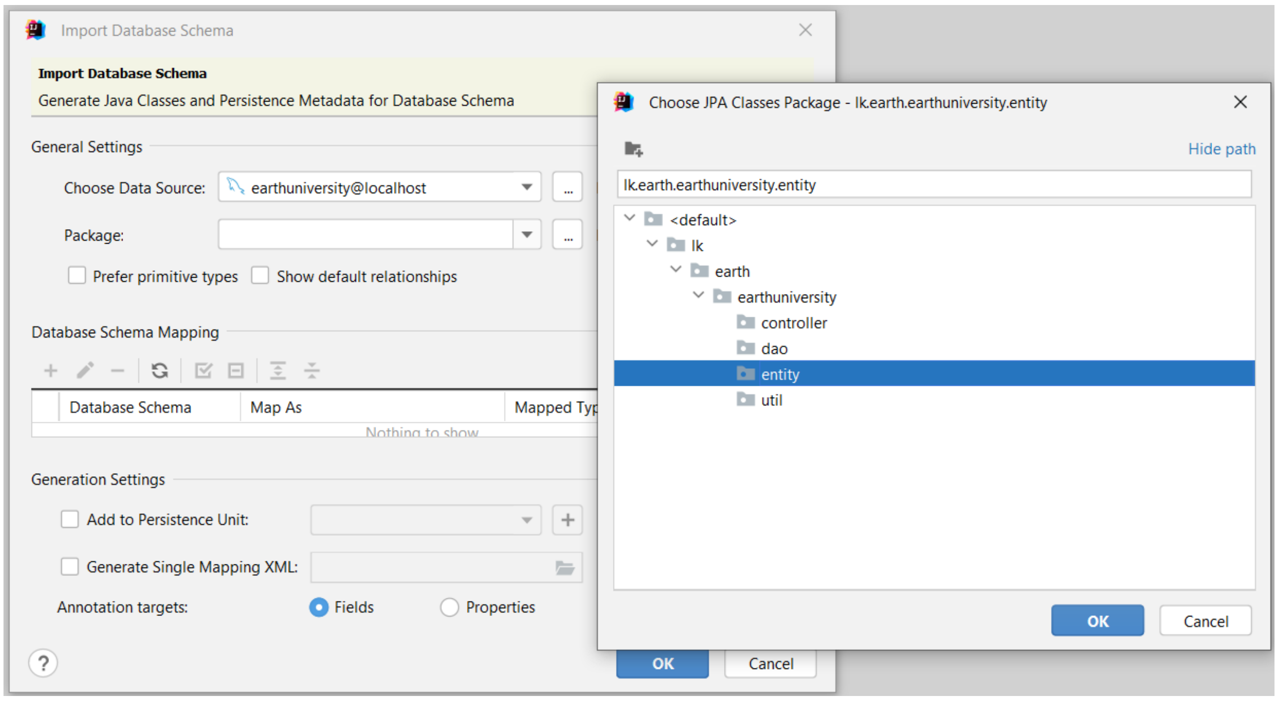
☐ Add to Persistence Unit: | ☐ Generate Column Propertie.

☐ Generate Single Mapping XML: | ☐ Generate Separate XML per

Annotation targets: [✓ Fields] Properties | ☑ Generate JPA Annotations (.

OK | Cancel

Select the Target Package where the auto-generated entities are to be placed



Import Database Schema

Import Database Schema

Generate Java Classes and Persistence Metadata for Database Schema

General Settings

Choose Data Source: earthuniversity@localhost

Package:

[ ] Prefer primitive types [ ] Show default relationships

Database Schema Mapping

Database Schema	Map As	Mapped Typ

Nothing to show

Generation Settings

[ ] Add to Persistence Unit:

[ ] Generate Single Mapping XML:

Annotation targets: [● Fields] ○ Properties

OK Cancel

Choose JPA Classes Package - lk.earth.earthuniversity.entity

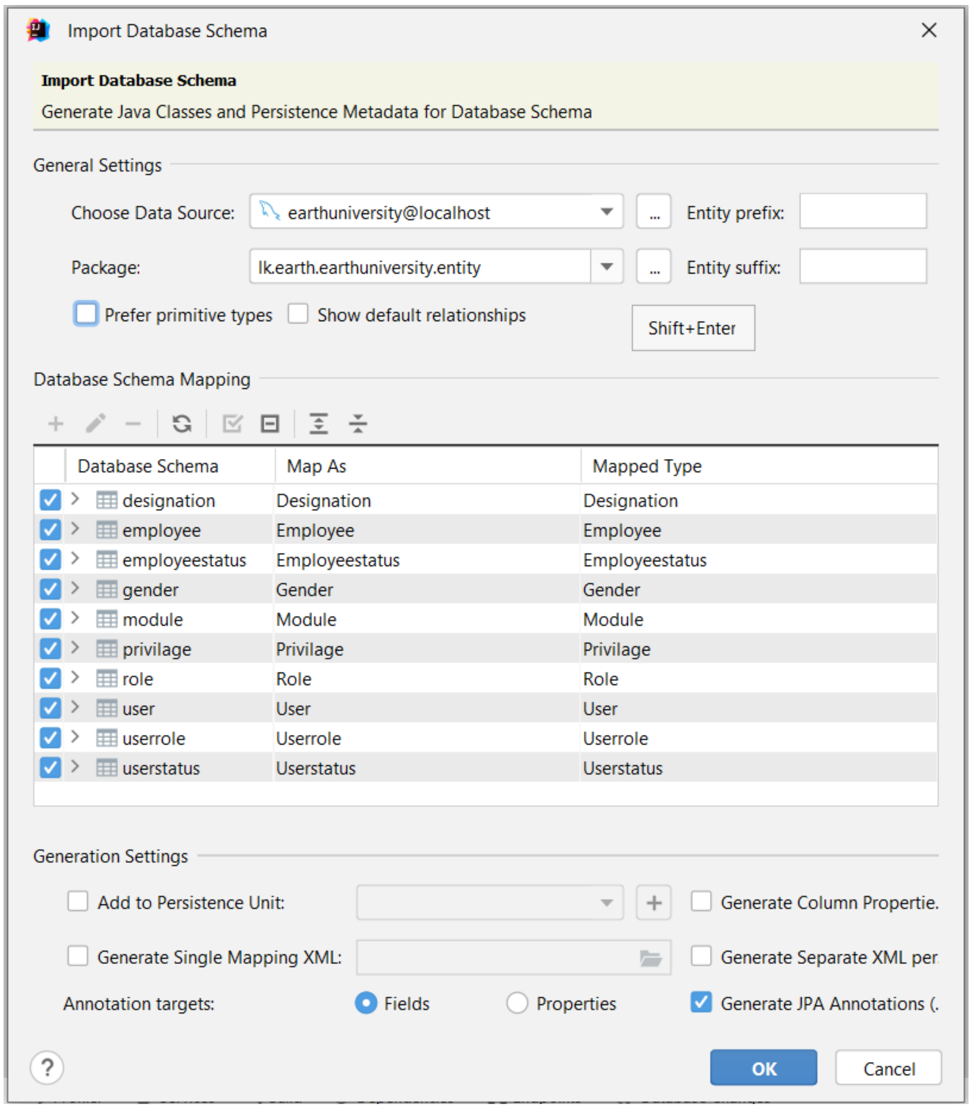
Hide path

lk.earth.earthuniversity.entity

- <default>
  - lk
    - earth
      - earthuniversity
        - controller
        - dao
        - entity
        - util

OK Cancel

Tick & Untick "Prefer Primitive Types" to get Updated Entities,
Then Select All, Need to tick "Genarate JPA Annotation"



Import Database Schema

Import Database Schema

Generate Java Classes and Persistence Metadata for Database Schema

General Settings

Choose Data Source: earthuniversity@localhost — Entity prefix:

Package: lk.earth.earthuniversity.entity — Entity suffix:

[ ] Prefer primitive types [ ] Show default relationships — Shift+Enter

Database Schema Mapping

	Database Schema	Map As	Mapped Type
✓	designation	Designation	Designation
✓	employee	Employee	Employee
✓	employeestatus	Employeestatus	Employeestatus
✓	gender	Gender	Gender
✓	module	Module	Module
✓	privilage	Privilage	Privilage
✓	role	Role	Role
✓	user	User	User
✓	userrole	Userrole	Userrole
✓	userstatus	Userstatus	Userstatus

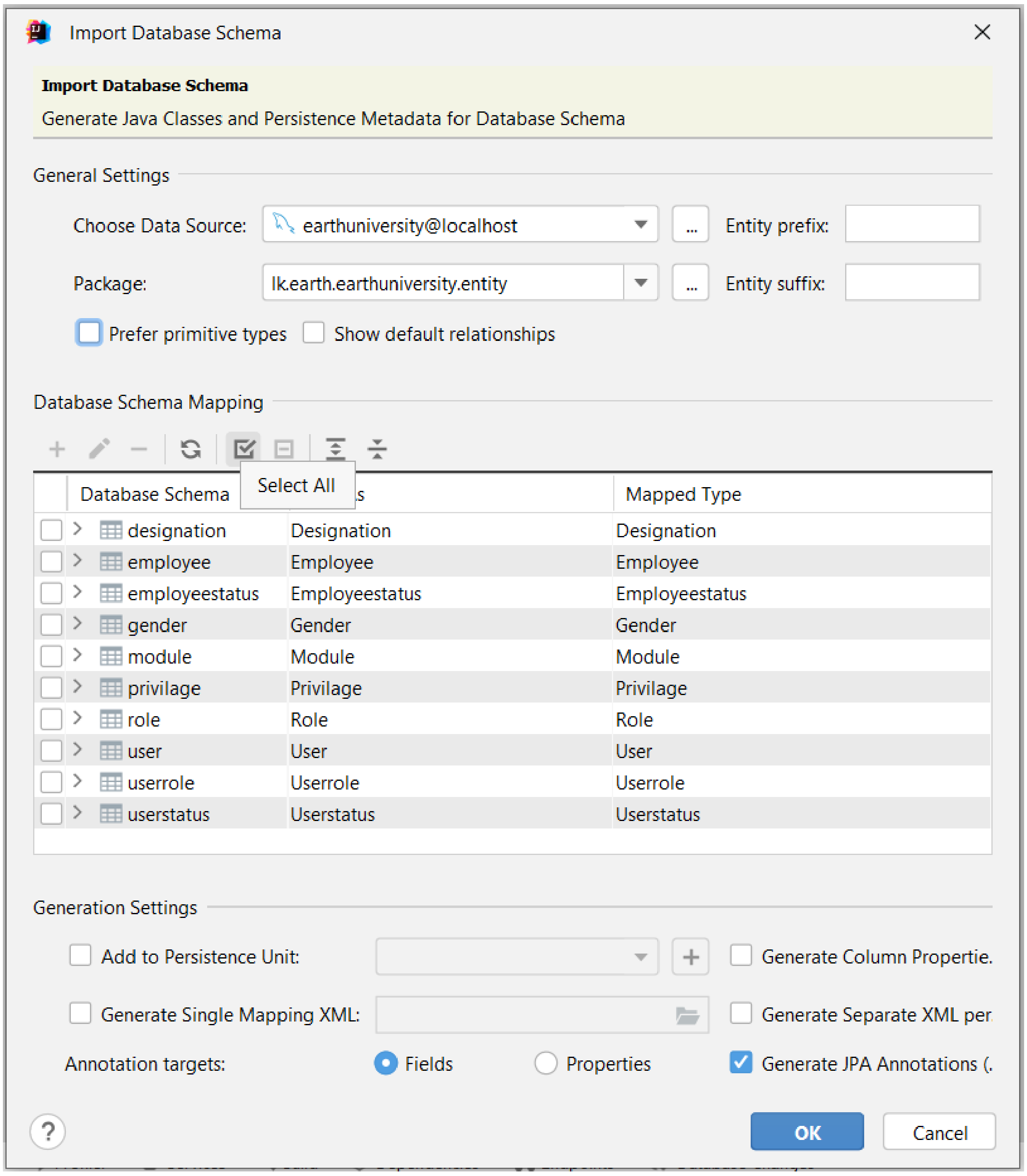
Generation Settings

[ ] Add to Persistence Unit: — [ ] Generate Column Propertie.

[ ] Generate Single Mapping XML: — [ ] Generate Separate XML per

Annotation targets: [● Fields] ○ Properties — [✓] Generate JPA Annotations (.

OK Cancel



Import Database Schema
Generate Java Classes and Persistence Metadata for Database Schema

General Settings

Choose Data Source: earthuniversity@localhost
Entity prefix:
Package: lk.earth.earthuniversity.entity
Entity suffix:
[ ] Prefer primitive types [ ] Show default relationships

Database Schema Mapping

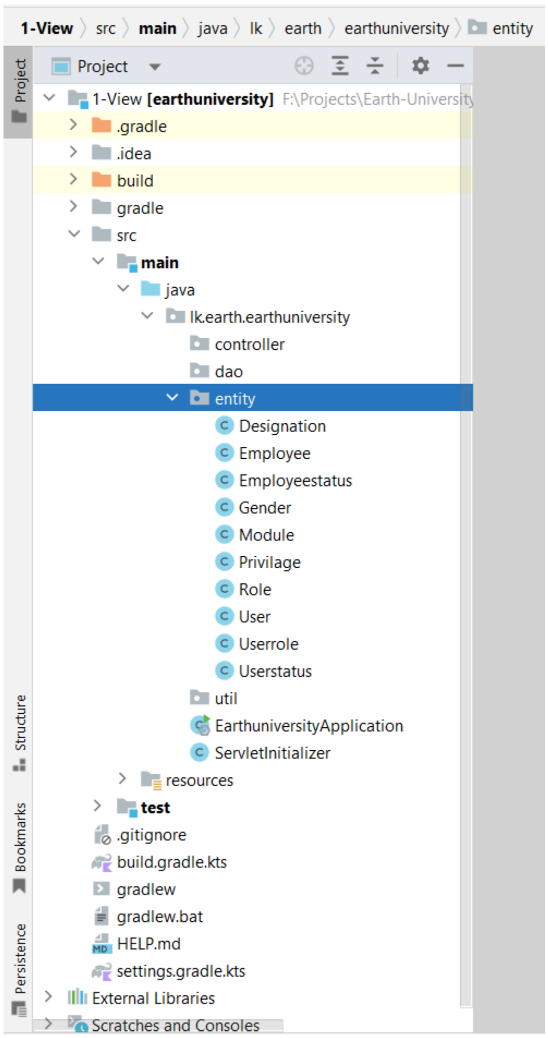
Database Schema	Class	Mapped Type
[ ] designation	Designation	Designation
[ ] employee	Employee	Employee
[ ] employeestatus	Employeestatus	Employeestatus
[ ] gender	Gender	Gender
[ ] module	Module	Module
[ ] privilage	Privilage	Privilage
[ ] role	Role	Role
[ ] user	User	User
[ ] userrole	Userrole	Userrole
[ ] userstatus	Userstatus	Userstatus

Generation Settings

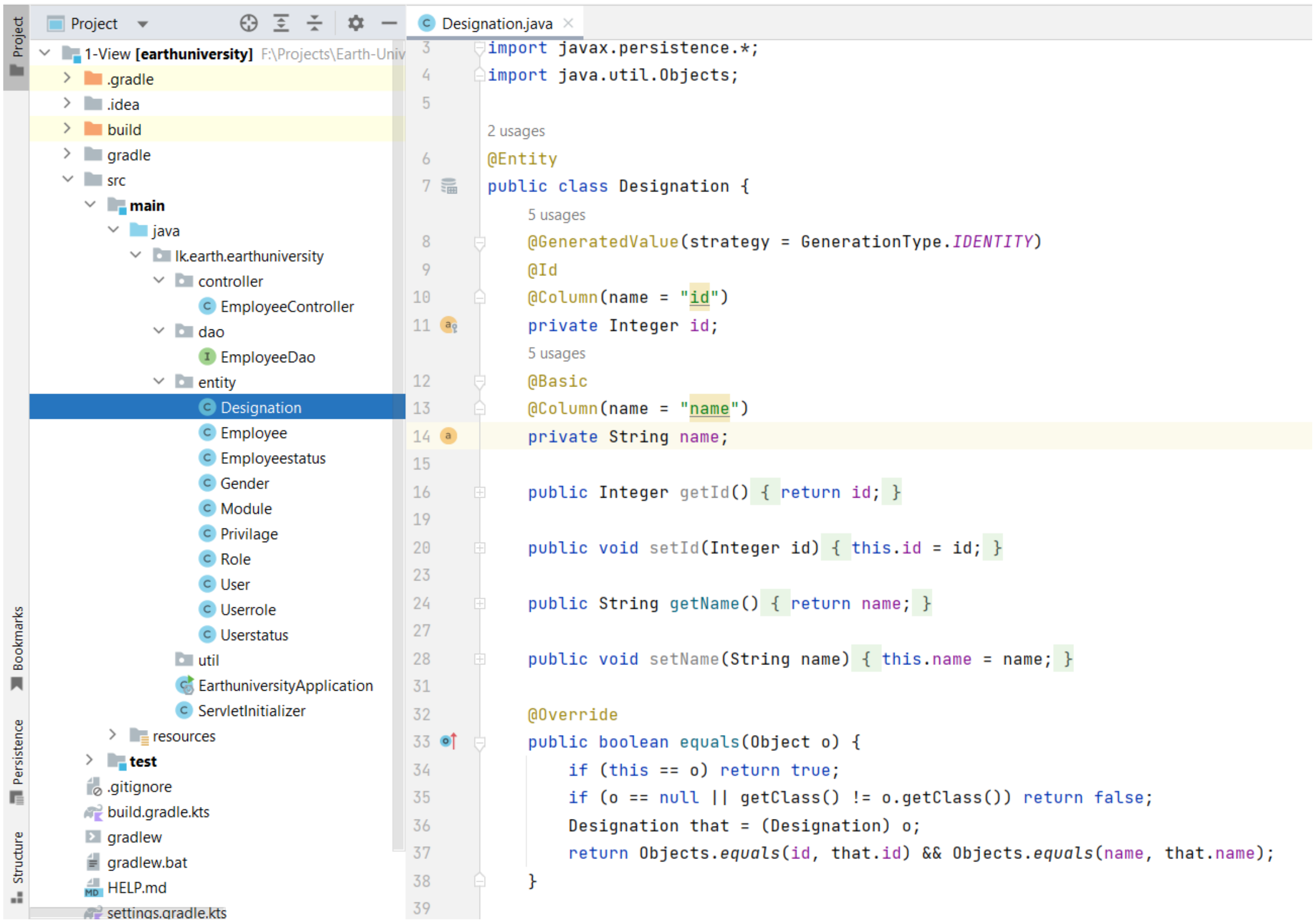
[ ] Add to Persistence Unit:
[ ] Generate Column Propertie.
[ ] Generate Single Mapping XML:
[ ] Generate Separate XML per
Annotation targets: [✓ Fields] Properties
[✓] Generate JPA Annotations (.

(d) Observe the Content of the Entity Classes

Attributes
Relationships
Constructors
Setters
Getters
Override Methods of Object Class
JPA Annotation for Object Relational Mapping(ORM) by the Hibernate



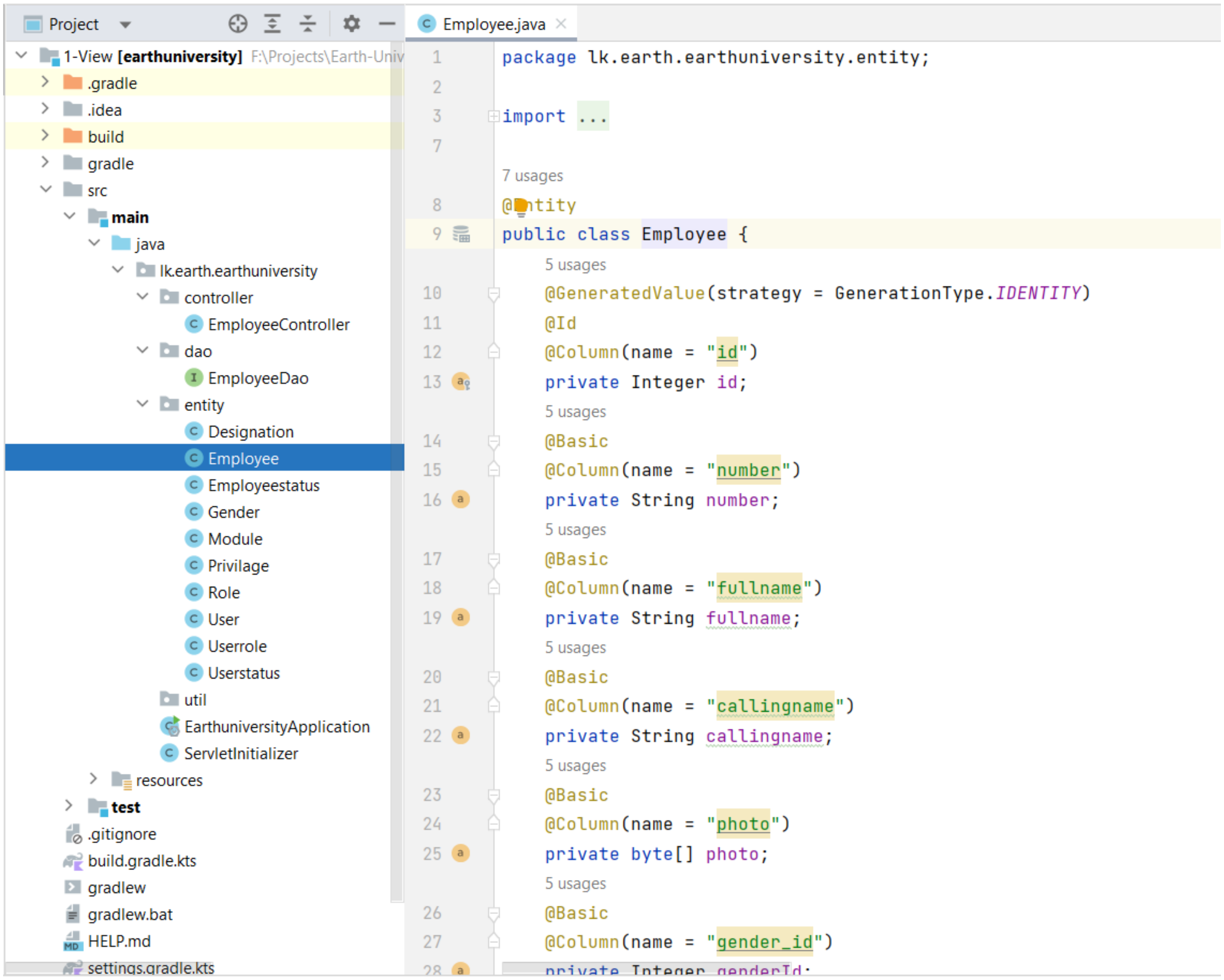
```
1-View 〉 src 〉 main 〉 java 〉 lk 〉 earth 〉 earthuniversity 〉 entity

Project ▾
1-View [earthuniversity] F:\Projects\Earth-University
  .gradle
  .idea
  build
  gradle
  src
    main
      java
        lk.earth.earthuniversity
          controller
          dao
          entity
            Designation
            Employee
            Employeestatus
            Gender
            Module
            Privilage
            Role
            User
            Userrole
            Userstatus
          util
          EarthuniversityApplication
          ServletInitializer
      resources
    test
  .gitignore
  build.gradle.kts
  gradlew
  gradlew.bat
  HELP.md
  settings.gradle.kts
External Libraries
Scratches and Consoles
```



```
Project ▾
1-View [earthuniversity] F:\Projects\Earth-Univ
  .gradle
  .idea
  build
  gradle
  src
    main
      java
        lk.earth.earthuniversity
          controller
            EmployeeController
          dao
            EmployeeDao
          entity
            Designation
            Employee
            Employeestatus
            Gender
            Module
            Privilage
            Role
            User
            Userrole
            Userstatus
          util
          EarthuniversityApplication
          ServletInitializer
      resources
    test
  .gitignore
  build.gradle.kts
  gradlew
  gradlew.bat
  HELP.md
  settings.gradle.kts
```

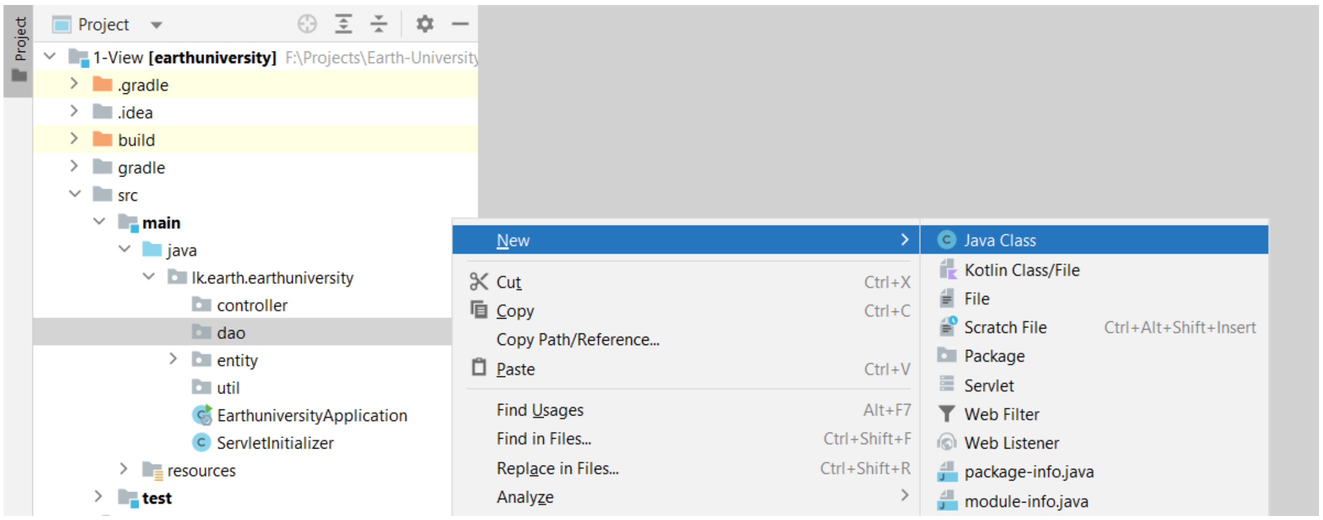
Designation.java

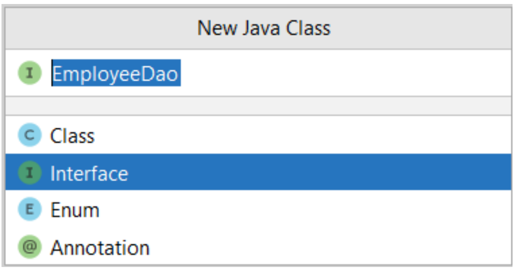
```java
import javax.persistence.*;
import java.util.Objects;

2 usages
@Entity
public class Designation {
    5 usages
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    @Column(name = "id")
    private Integer id;
    5 usages
    @Basic
    @Column(name = "name")
    private String name;

    public Integer getId() { return id; }

    public void setId(Integer id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Designation that = (Designation) o;
        return Objects.equals(id, that.id) && Objects.equals(name, that.name);
    }

```

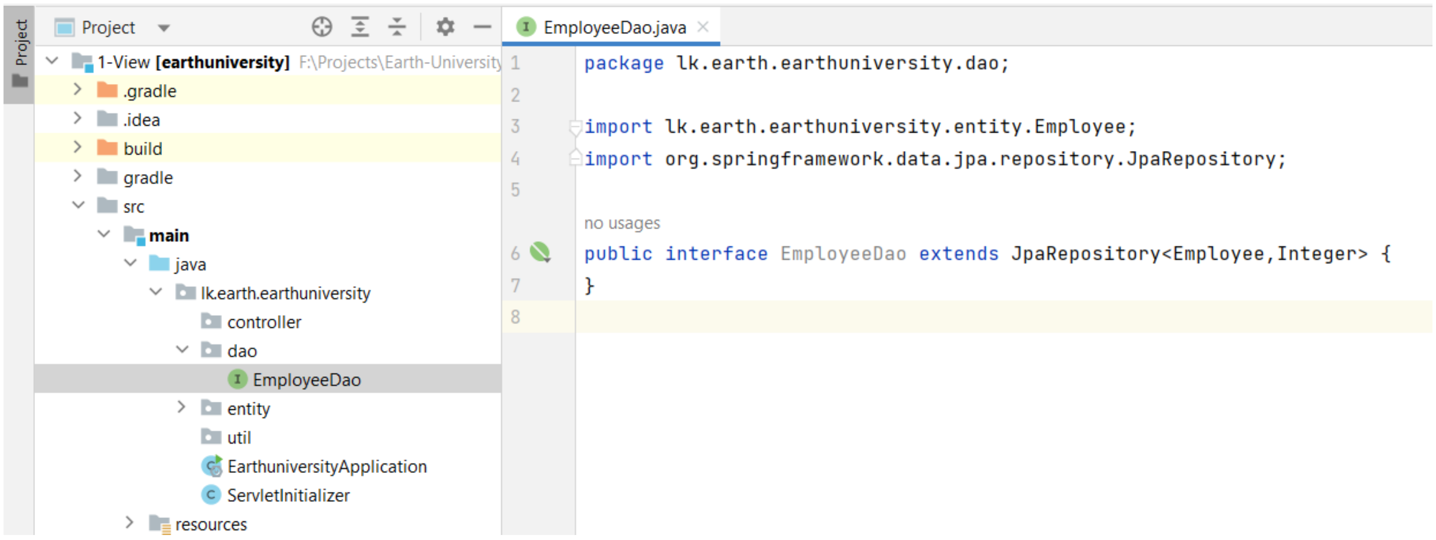


```java
package lk.earth.earthuniversity.entity;

import ...

7 usages
@Entity
public class Employee {
    5 usages
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    @Column(name = "id")
    private Integer id;
    5 usages
    @Basic
    @Column(name = "number")
    private String number;
    5 usages
    @Basic
    @Column(name = "fullname")
    private String fullname;
    5 usages
    @Basic
    @Column(name = "callingname")
    private String callingname;
    5 usages
    @Basic
    @Column(name = "photo")
    private byte[] photo;
    5 usages
    @Basic
    @Column(name = "gender_id")
    private Integer genderId;
```

2.2 Dao Coding

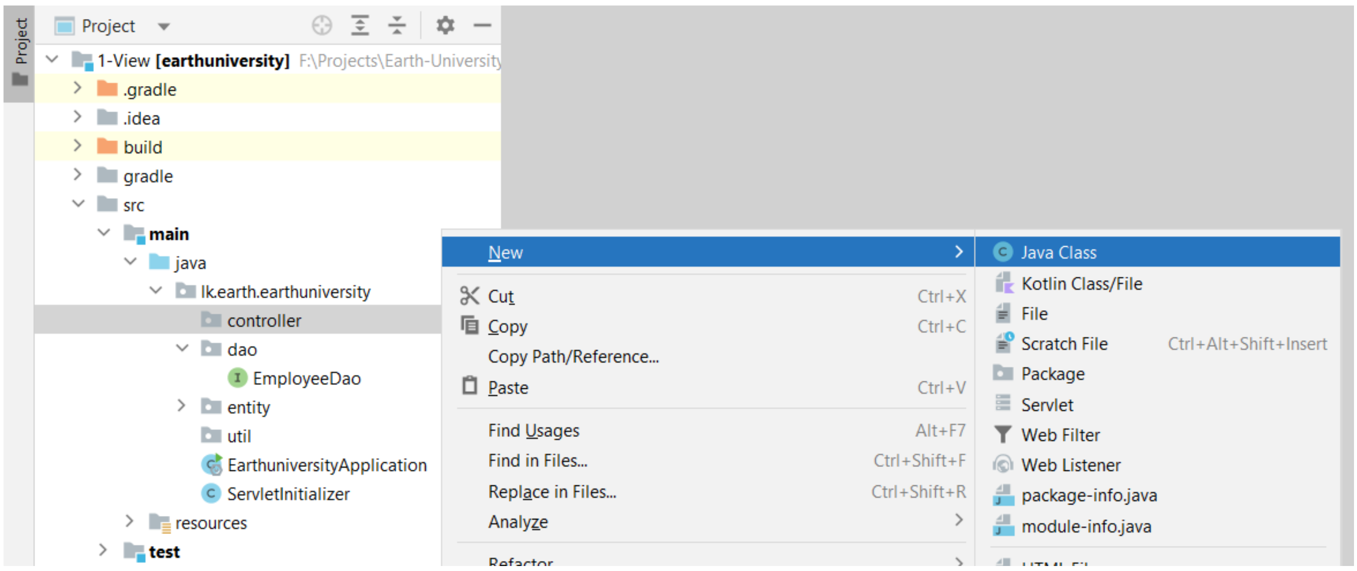


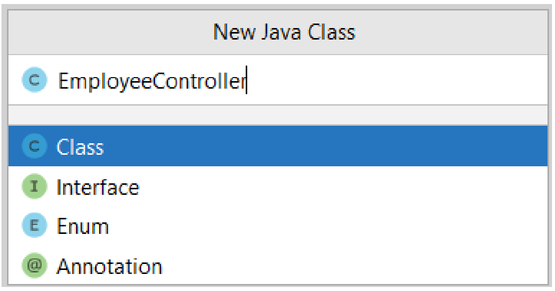


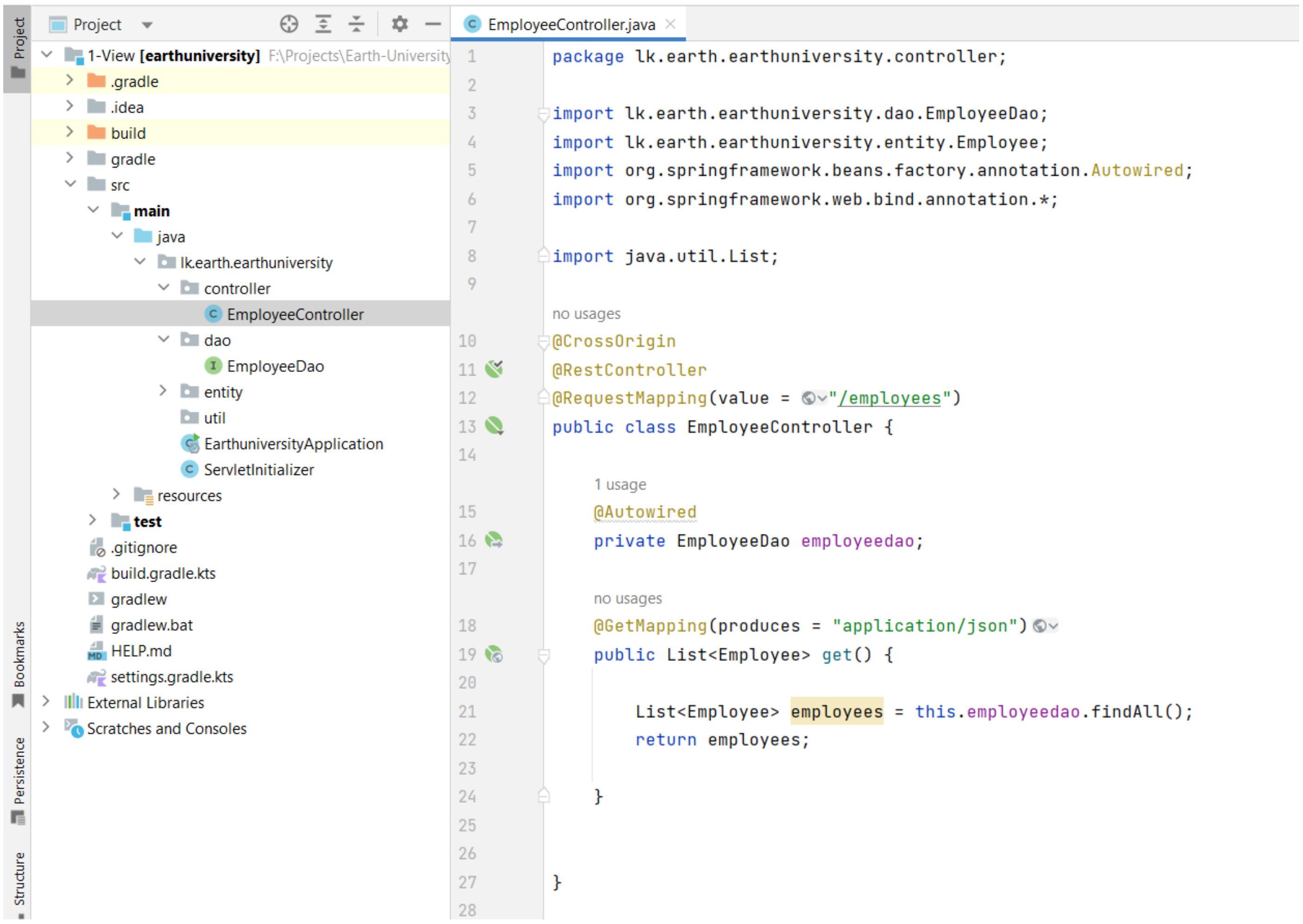


```java
package lk.earth.earthuniversity.dao;

import lk.earth.earthuniversity.entity.Employee;
import org.springframework.data.jpa.repository.JpaRepository;

no usages
public interface EmployeeDao extends JpaRepository<Employee,Integer> {
}
```

2.3 Controller Coding





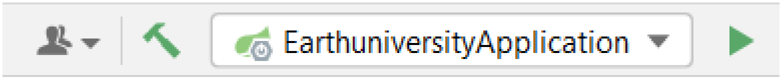


```
1-View [earthuniversity] F:\Projects\Earth-University
  .gradle
  .idea
  build
  gradle
  src
    main
      java
        lk.earth.earthuniversity
          controller
            EmployeeController
          dao
            EmployeeDao
          entity
          util
          EarthuniversityApplication
          ServletInitializer
      resources
    test
  .gitignore
  build.gradle.kts
  gradlew
  gradlew.bat
  HELP.md
  settings.gradle.kts
External Libraries
Scratches and Consoles
```

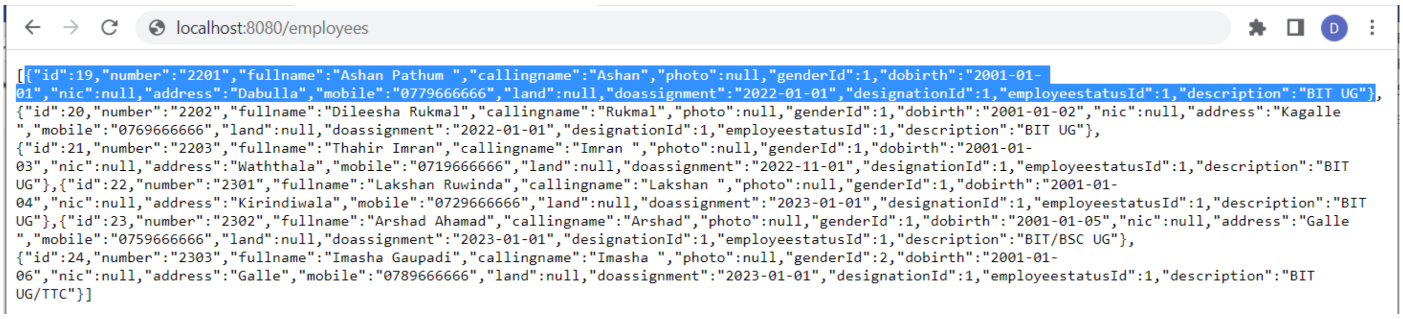
```java
package lk.earth.earthuniversity.controller;

import lk.earth.earthuniversity.dao.EmployeeDao;
import lk.earth.earthuniversity.entity.Employee;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@CrossOrigin
@RestController
@RequestMapping(value = "/employees")
public class EmployeeController {

    @Autowired
    private EmployeeDao employeedao;

    @GetMapping(produces = "application/json")
    public List<Employee> get() {

        List<Employee> employees = this.employeedao.findAll();
        return employees;

    }


}
```

2.4 Testing Employee Service

Run the Project (Compile → Build → Configure with Spring Boot → Server Startup → Deploy)



Accessing the Service using Chrome



localhost:8080/employees

```
[{"id":19,"number":"2201","fullname":"Ashan Pathum ","callingname":"Ashan","photo":null,"genderId":1,"dobirth":"2001-01-
01","nic":null,"address":"Dabulla","mobile":"0779666666","land":null,"doassignment":"2022-01-01","designationId":1,"employeestatusId":1,"description":"BIT UG"},
{"id":20,"number":"2202","fullname":"Dileesha Rukmal","callingname":"Rukmal","photo":null,"genderId":1,"dobirth":"2001-01-02","nic":null,"address":"Kagalle
","mobile":"0769666666","land":null,"doassignment":"2022-01-01","designationId":1,"employeestatusId":1,"description":"BIT UG"},
{"id":21,"number":"2203","fullname":"Thahir Imran","callingname":"Imran ","photo":null,"genderId":1,"dobirth":"2001-01-
03","nic":null,"address":"Waththala","mobile":"0719666666","land":null,"doassignment":"2022-11-01","designationId":1,"employeestatusId":1,"description":"BIT
UG"},{"id":22,"number":"2301","fullname":"Lakshan Ruwinda","callingname":"Lakshan ","photo":null,"genderId":1,"dobirth":"2001-01-
04","nic":null,"address":"Kirindiwala","mobile":"0729666666","land":null,"doassignment":"2023-01-01","designationId":1,"employeestatusId":1,"description":"BIT
UG"},{"id":23,"number":"2302","fullname":"Arshad Ahamad","callingname":"Arshad","photo":null,"genderId":1,"dobirth":"2001-01-05","nic":null,"address":"Galle
","mobile":"0759666666","land":null,"doassignment":"2023-01-01","designationId":1,"employeestatusId":1,"description":"BIT/BSC UG"},
{"id":24,"number":"2303","fullname":"Imasha Gaupadi","callingname":"Imasha ","photo":null,"genderId":2,"dobirth":"2001-01-
06","nic":null,"address":"Galle","mobile":"0789666666","land":null,"doassignment":"2023-01-01","designationId":1,"employeestatusId":1,"description":"BIT
UG/TTC"}]
```