



If you do not pay any relevant course fee to maintain Earth University BIT platform,
Be ethical enough to contribute while you are using our Academic Contents

www.earth.lk/community
2004-2024 (20th Anniversary)

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

2(b) Employee View

2(c) Employee Search

Design Search Criteria
Server App Modifications
Client App Modifications

2(d) Employee Insert

2(e) Employee Update

2(f) Employee Delete

2(c) Employee Search

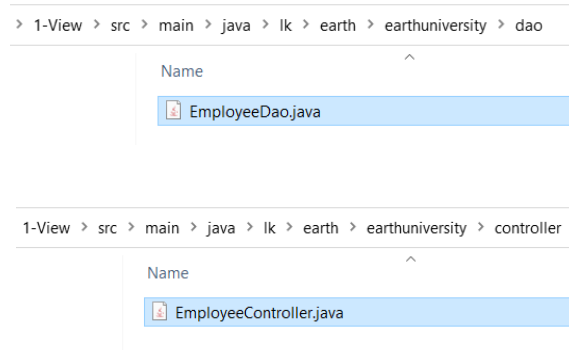
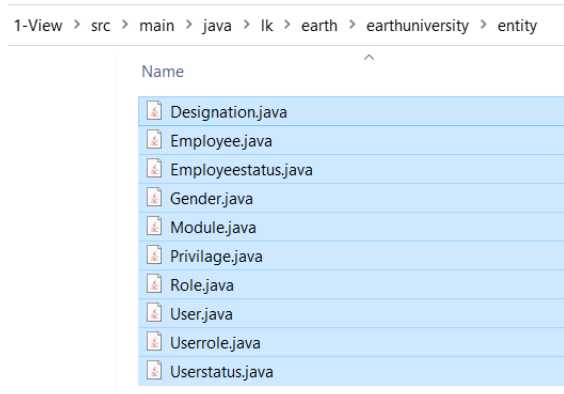
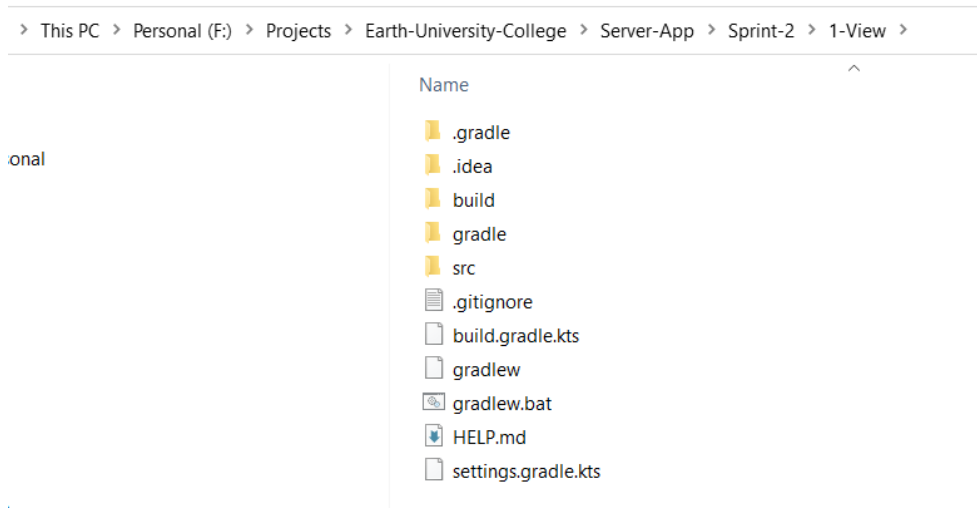
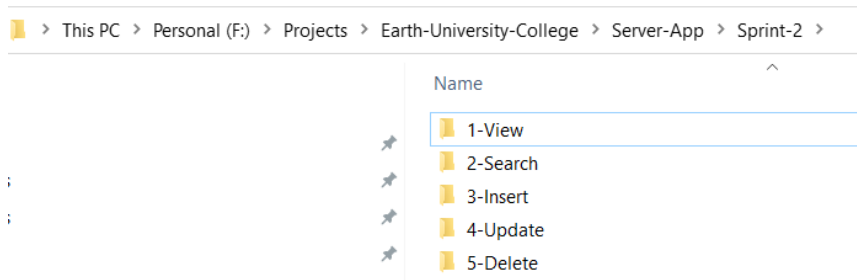
1. Design Search Criteria
Define Search Attributes
Define Lists Required for Search Panels
Define SQL Queries Needed
2. Server-App Modification

Implement Listing Services
Implement Searching Service
(1) Using Controller Filtering
(2) Search by QBE
(3) Search by Named Queries
(4) Search by Native Queries

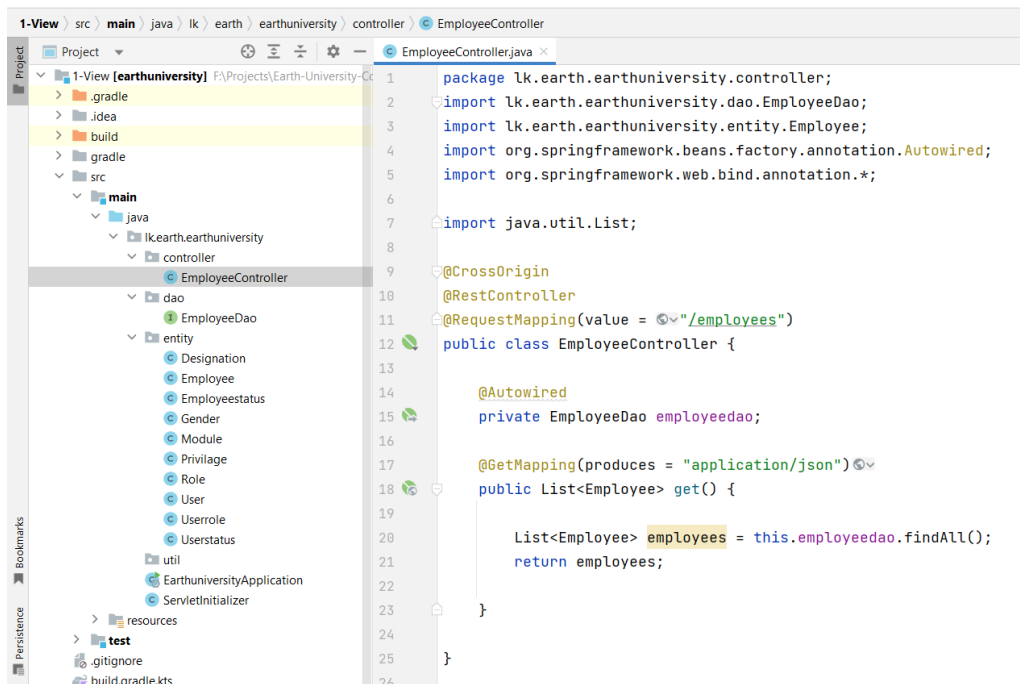
3. Client-App Modification

1. Content of the “Server-App” folder at the End of Sprint-2 “View” Completion

(Make sure you must start from here)

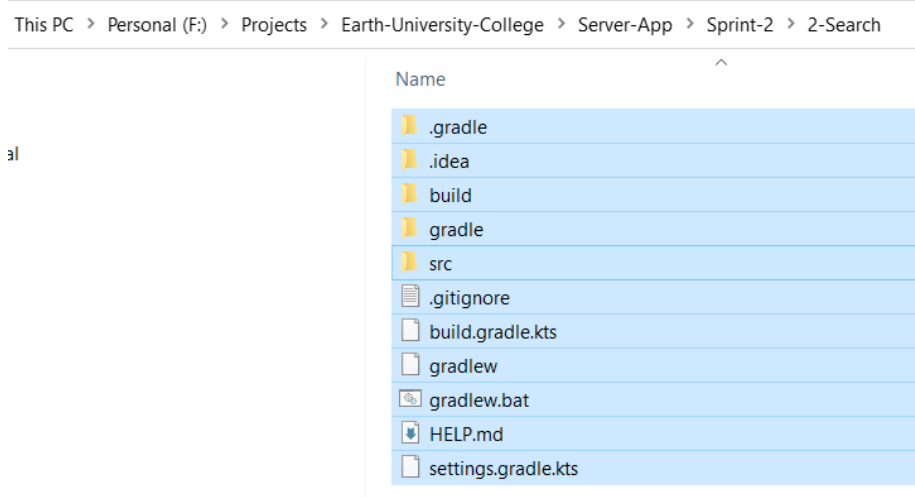
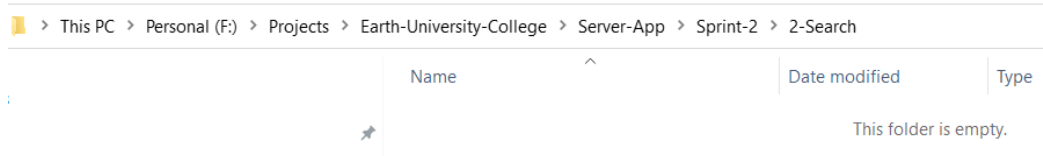
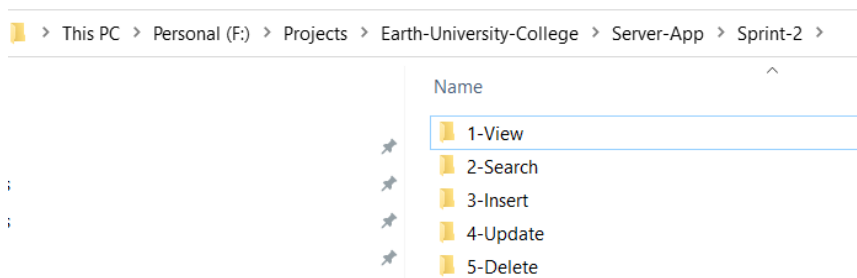


```
1 package lk.earth.earthuniversity.dao;
2
3 import lk.earth.earthuniversity.entity.Employee;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface EmployeeDao extends JpaRepository<Employee,Integer> {
7
8 }
```

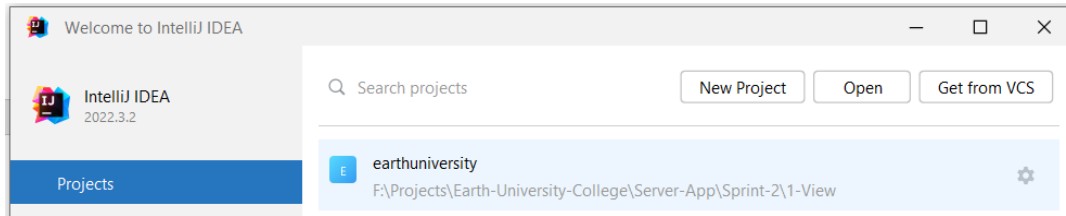
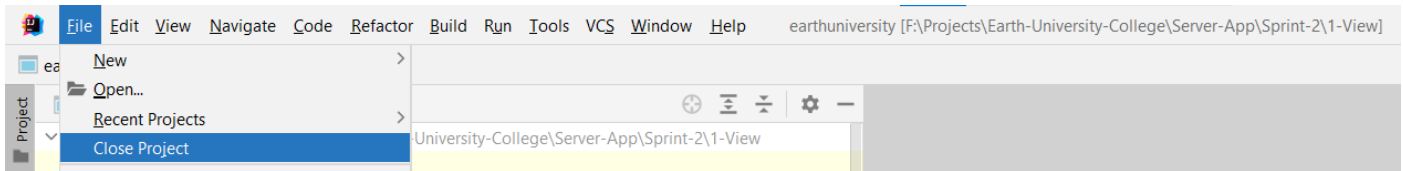


Entities were auto generated and have not done any modification

2. Copy the Content of “1-View” into “2-Search”

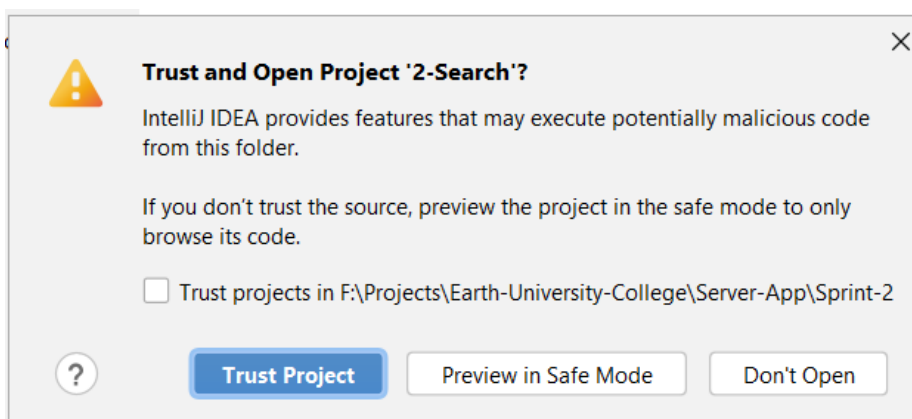
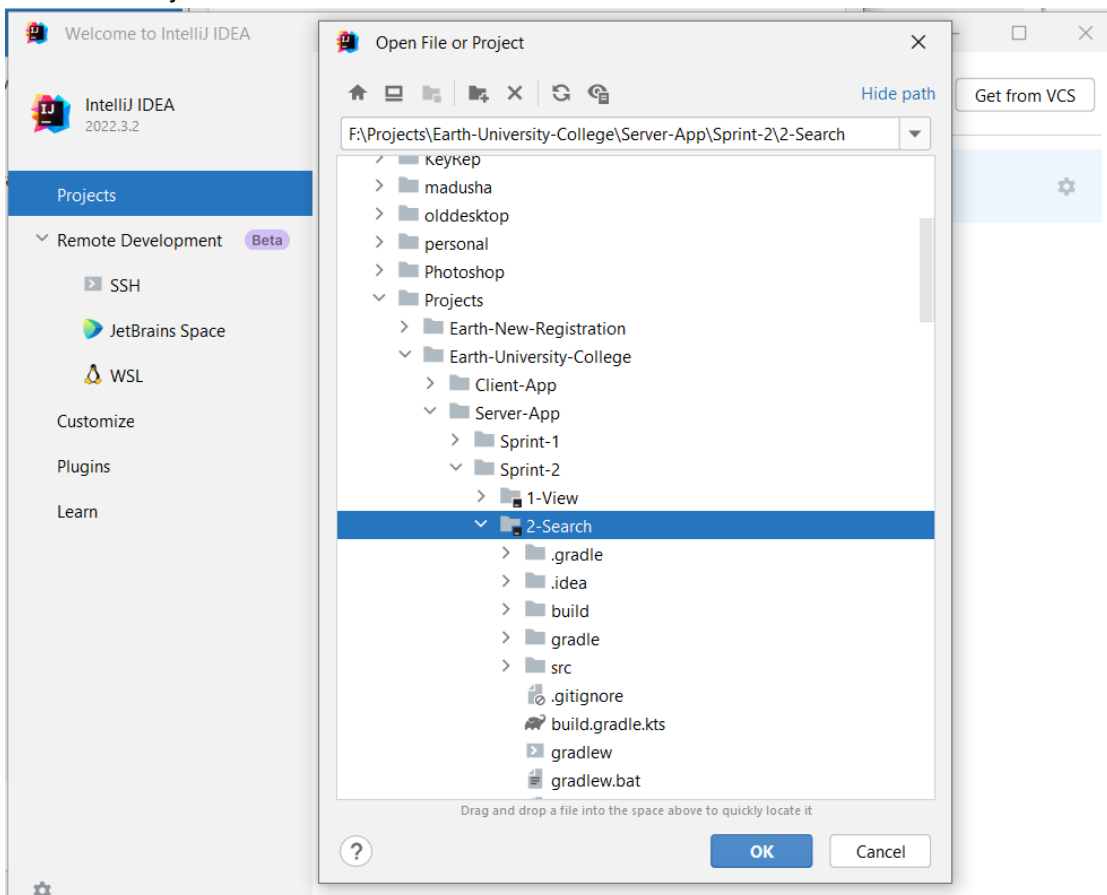


3. Open the IntelliJ-IDEA Project from the content in the “2-Search” folder

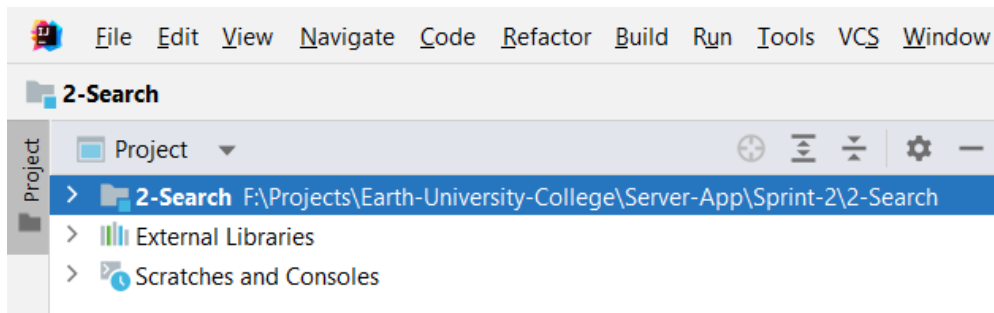


Click on Open

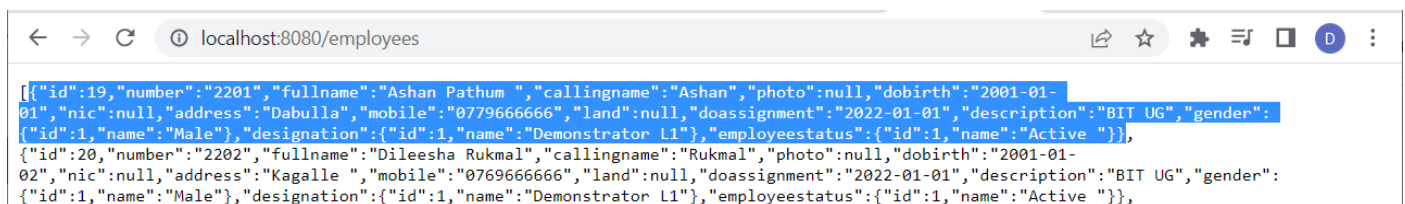
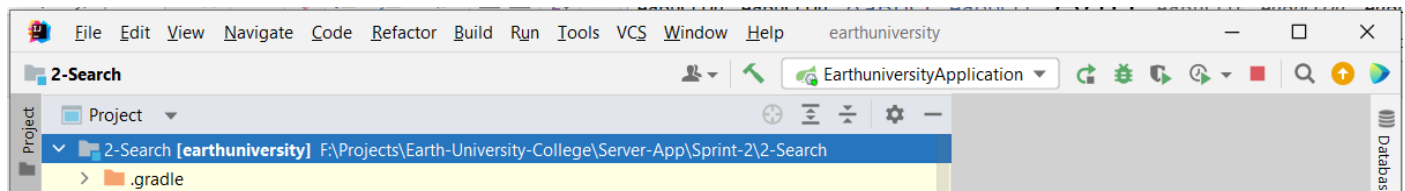
Select the Project and Click OK



Click on “Trust Project”



Run the Project and Test the Output as follows



2(c) Employee Search

4. [Design Search Criteria](#)

Define Search Attributes

Define Lists Required for Search Panels

Define SQL Quarries Needed

5. [Server-App Modification](#)

Implement Listing Services

Implement Searching Service

(5) Using Controller Filtering

(6) Search by QBE

(7) Search by Named Queries

(8) Search by Native Queries

6. [Client-App Modification](#)

Define Search Attributes required Object Lists for Search Combos

1	number	Exact Search	
2	fullname	Like Search	
3	nic	Like Search	
4	designation	designation.id	Designation List
5	gender	gender.id	Gender List

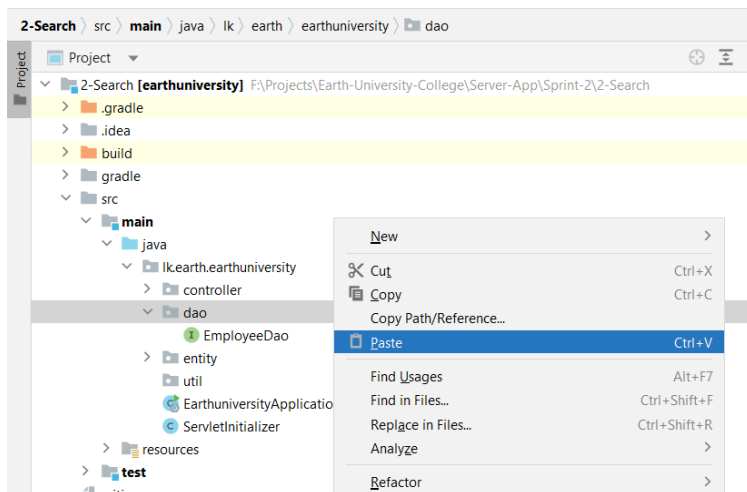
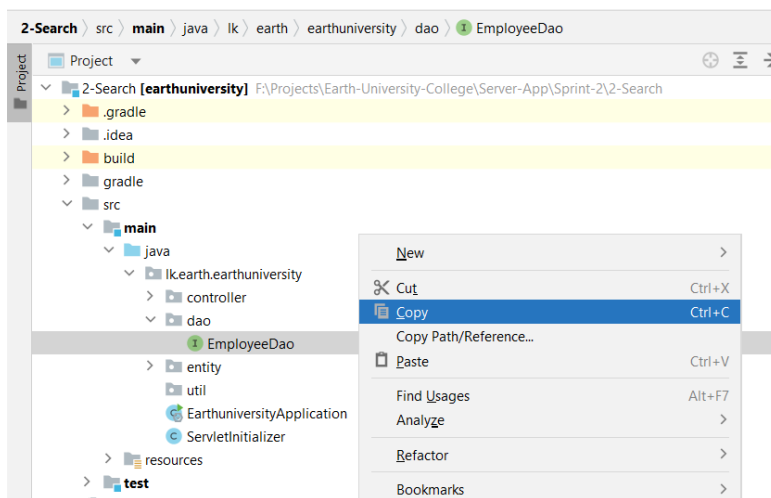
Object Lists – Vertically projected attribute list mostly with the “id” and “name” attributes.
These lists are important for Combo Boxes in both “Server-Search” & “Add-New” Forms.

- (1) – Implementing Listing Service
- (2) – Implementing Searching Service

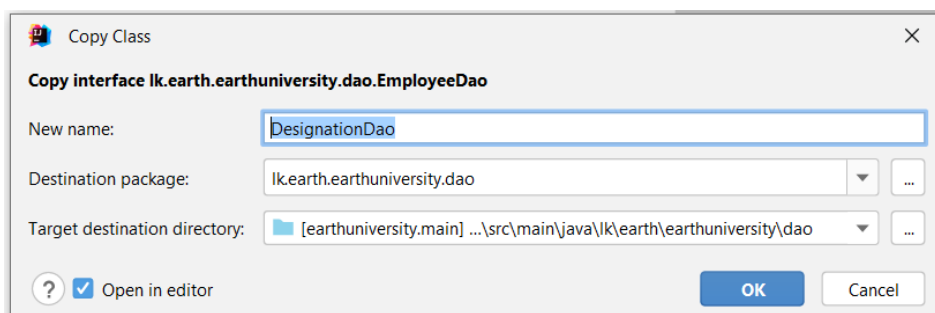
(1) Implement “listing” services

- (a) DAO – DesignationDao, GenderDao
- (b) Controller – DesignationController, GenderController
- (c) Testing “genders/list”, “designation/list”

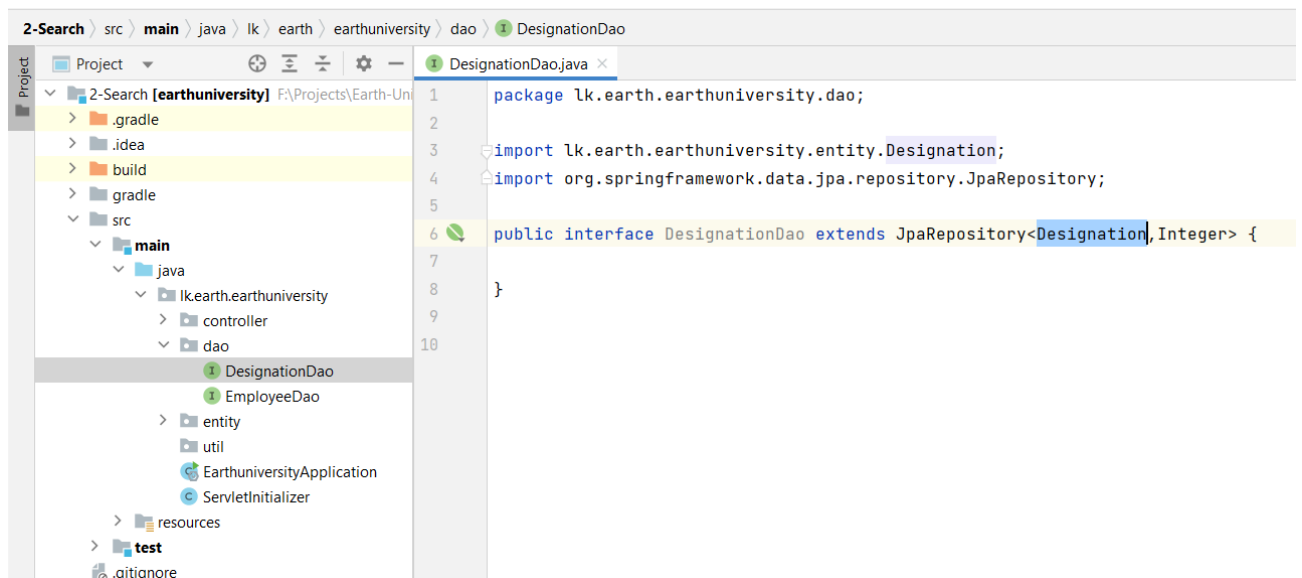
(a) DAO – DesignationDao, GenderDao



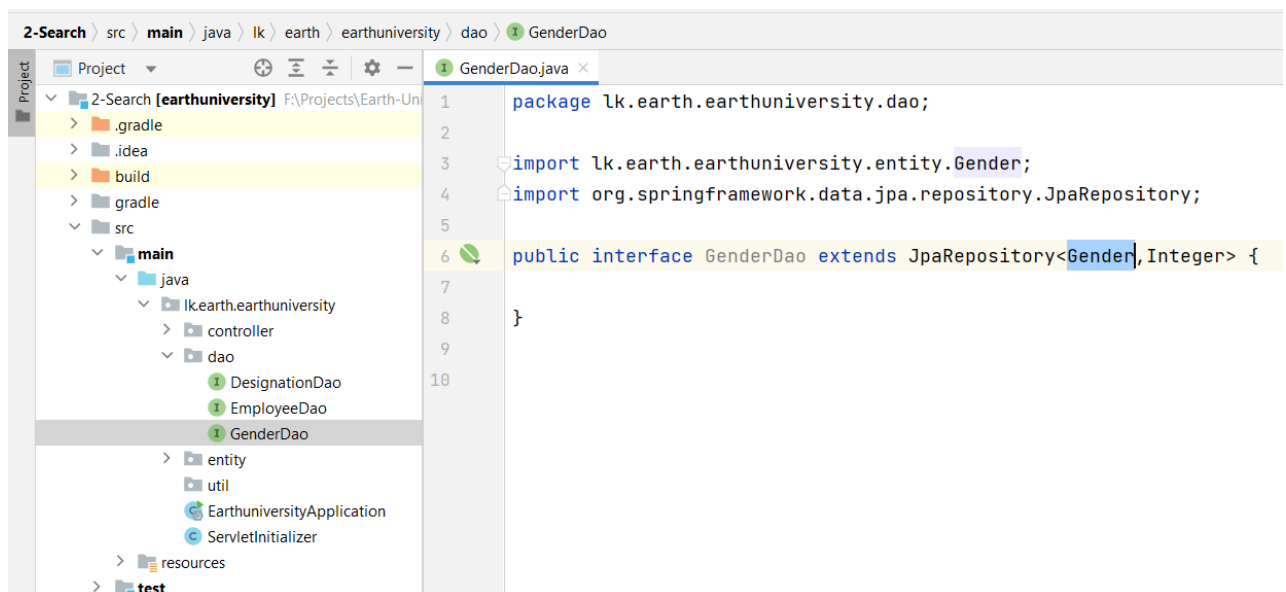
Change “EmployeeDao” to “DesignationDao”



Change the “Employee” into “Designation”



Repeat the Process to create “GenderDao” as same as the above



(b) Controller – DesignationController, GenderController

Copy the “EmployeeController” and Paste as “GenderController”
Change the codes in the “GenderController”.

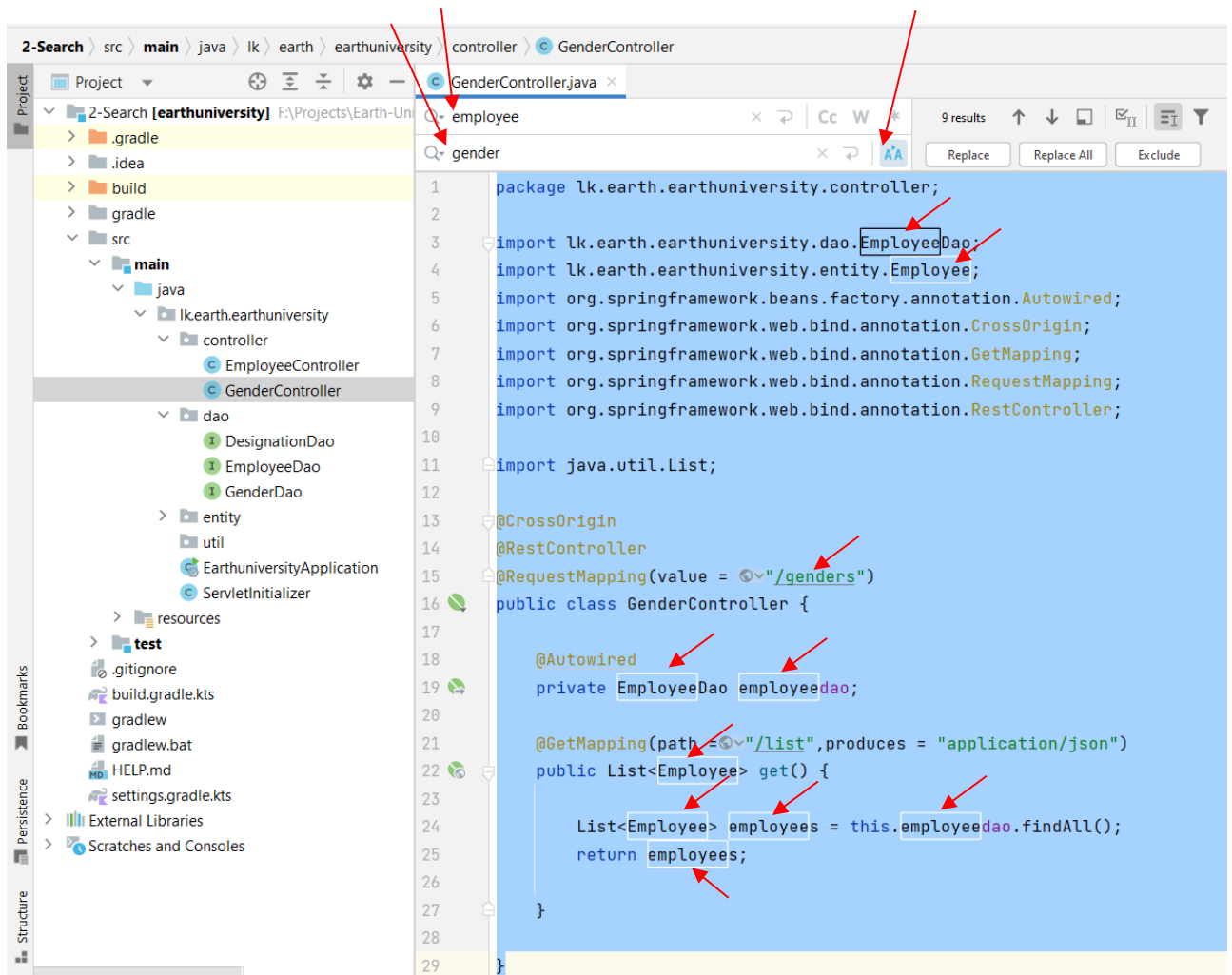
Select the Codes

Click “Ctrl” + “R” → Find & Replace

Coding use shortest time period of the Project when you a member of a Software Firm with High CMM Level.

Use automotive tools and techniques as far as possible to reduce coding time.

Here, Expected time to code a Module must get less than 6Hours.



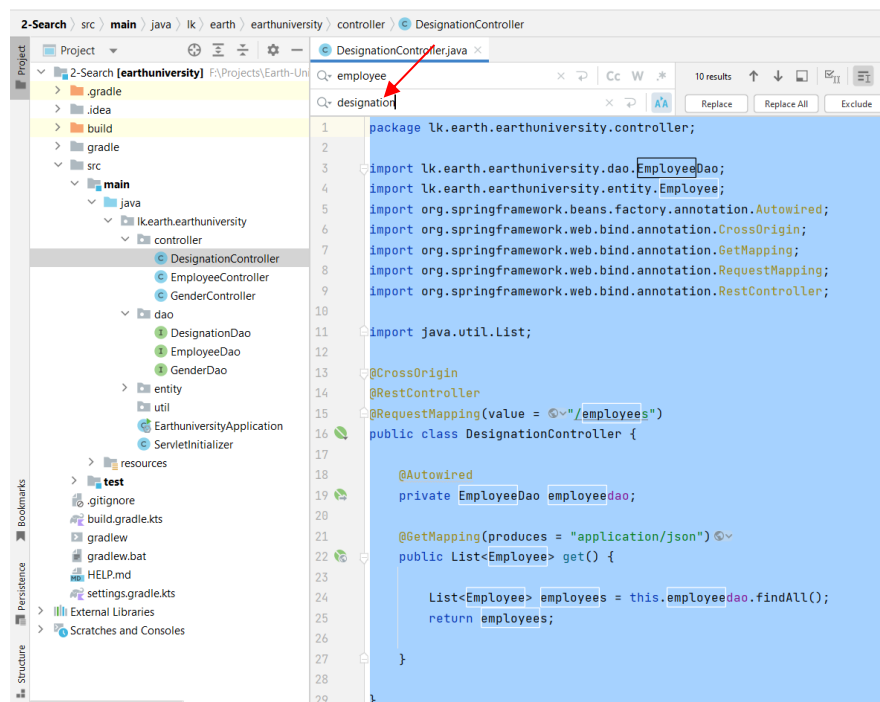
Replace All. Add → **path="/list"** into the @GetMapping annotation

Run the Project and Test as follows

← → ↻ ⓘ localhost:8080/genders/list

```
[{"id":1,"name":"Male"}, {"id":2,"name":"Female "}, {"id":3,"name":"Other"}]
```

Repeat the Process with DesignationController and test as the following




```
[{"id":1,"name":"Demonstrator L1"}, {"id":2,"name":"Assistent G1"}, {"id":3,"name":"Assistent G2"}, {"id":4,"name":"Demonstrator L2"}, {"id":5,"name":"Demonstrator L3"}]
```

Change the controller to project only the attribute “name” as follows.

```
GenderController.java
14 @RestController
15 @RequestMapping(value = "/genders")
16 public class GenderController {
17
18     @Autowired
19     private GenderDao genderdao;
20
21     @GetMapping(path = "/list", produces = "application/json")
22     public List<Gender> get() {
23
24         List<Gender> genders = this.genderdao.findAll();
25         return genders;
26
27     }
28
29 }
```

```
GenderController.java
1 package lk.earth.earthuniversity.controller;
2
3 import ...
4
13
14 @CrossOrigin
15 @RestController
16 @RequestMapping(value = "/genders")
17 public class GenderController {
18
19     @Autowired
20     private GenderDao genderdao;
21
22     @GetMapping(path = "/list", produces = "application/json")
23     public List<Gender> get() {
24
25         List<Gender> genders = this.genderdao.findAll();
26
27         genders = genders.stream().map(
28             gender -> {Gender g = new Gender(); g.setName(gender.getName()); return g;}
29         ).collect(Collectors.toList());
30
31         return genders;
32
33     }
34
35 }
```

Observe the outcome

```
[{"id":null,"name":"Male"}, {"id":null,"name":"Female "}, {"id":null,"name":"Other"}]
```

Change the code to get Both ID and Name.

```
genders = genders.stream().map(
    gender -> { Gender g = new Gender();
                g.setId(gender.getId());
                g.setName(gender.getName());
                return g; }
).collect(Collectors.toList());
```

Re run the Project and observe the outcome.

```
localhost:8080/genders/list

[{"id":1,"name":"Male"}, {"id":2,"name":"Female "}, {"id":3,"name":"Other"}]
```

Do the Same Changers in Designation Controller and Test the outcome

```
@GetMapping(path = "/list", produces = "application/json")
public List<Designation> get() {

    List<Designation> designations = this.designationdao.findAll();

    designations = designations.stream().map(
        designation -> { Designation d = new Designation();
                        d.setId(designation.getId());
                        d.setName(designation.getName());
                        return d; }
    ).collect(Collectors.toList());

    return designations;
}
```

```
localhost:8080/designations/list

[{"id":1,"name":"Demonstrator L1"}, {"id":2,"name":"Assistant G1"}, {"id":3,"name":"Assistant G2"}, {"id":4,"name":"Demonstrator L2"}, {"id":5,"name":"Demonstrator L3"}]
```

(2) Implementing Searching Service

Using Query Parameters (**employees?designation=1&number=2201**) ← Method used by our project

Using Path Parameters (**employees/designation/1/number/2201**)

(a) Convert the Get method of the Controller as follows to get Search by Number Facility,

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

    List<Employee> employees = this.employeeDao.findAll();
    return employees;
}
```

→

```

@GetMapping(produces = "application/json")
public List<Employee> get(@RequestParam HashMap<String, String> params) {

    String number = params.get("number");

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

    if(params.isEmpty()) return employees;

    employees = employees.stream().filter(
        employee -> {
            if(number!=null) return employee.getNumber().equals(number);
            return false;
        }).collect(Collectors.toList());

    return employees;
}

```

Re Run the Project and Observe the Outcome

localhost:8080/employees?number=2201

```

[{"id":19,"number":"2201","fullname":"Ashan Pathum ","callingname":"Ashan","photo":null,"dobirth":"2001-01-01","nic":null,"address":"Dabulla","mobile":"0779666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}]

```

localhost:8080/employees?number=2202

```

[{"id":20,"number":"2202","fullname":"Dileesha Rukmal","callingname":"Rukmal","photo":null,"dobirth":"2001-01-02","nic":null,"address":"Kagalle ","mobile":"0769666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}]

```

Regression Testing

localhost:8080/employees

```

[{"id":19,"number":"2201","fullname":"Ashan Pathum ","callingname":"Ashan","photo":null,"dobirth":"2001-01-01","nic":null,"address":"Dabulla","mobile":"0779666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":20,"number":"2202","fullname":"Dileesha Rukmal","callingname":"Rukmal","photo":null,"dobirth":"2001-01-02","nic":null,"address":"Kagalle ","mobile":"0769666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":21,"number":"2203","fullname":"Thahir Imran","callingname":"Imran ","photo":null,"dobirth":"2001-01-03","nic":null,"address":"Waththala","mobile":"0719666666","land":null,"doassignment":"2022-11-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":22,"number":"2301","fullname":"Lakshan Ruwinda","callingname":"Lakshan ","photo":null,"dobirth":"2001-01-04","nic":null,"address":"Kirindiwala","mobile":"0729666666","land":null,"doassignment":"2023-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":23,"number":"2302","fullname":"Arshad Ahamad","callingname":"Arshad","photo":null,"dobirth":"2001-01-05","nic":null,"address":"Galle ","mobile":"0759666666","land":null,"doassignment":"2023-01-01","description":"BIT/BSC UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":24,"number":"2303","fullname":"Imasha Gaupadi","callingname":"Imasha ","photo":null,"dobirth":"2001-01-06","nic":null,"address":"Galle","mobile":"0789666666","land":null,"doassignment":"2023-01-01","description":"BIT UG/TTC","gender":{"id":2,"name":"Female"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}]

```

(3) Convert the above Controller to Filter employees both based on "gender" and "number"

```
@GetMapping(produces = "application/json")
public List<Employee> get(@RequestParam HashMap<String, String> params) {

    String number = params.get("number");
    String genderid= params.get("genderid");

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

    if(params.isEmpty()) return employees;

    employees = employees.stream().filter(
        employee -> {
            if(number!=null) return employee.getNumber().equals(number);
            if(genderid!=null) return employee.getGender().getId()==Integer.parseInt(genderid);
            return false;
        }).collect(Collectors.toList());

    return employees;
}
```

localhost:8080/employees?genderid=2

```
[{"id":24,"number":"2303","fullname":"Imasha Gaupadi","callingname":"Imasha ","photo":null,"dobirth":"2001-01-06","nic":null,"address":"Galle","mobile":"0789666666","land":null,"doassignment":"2023-01-01","description":"BIT UG/TTC","gender":{"id":2,"name":"Female"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}]
```

localhost:8080/employees?genderid=1

```
[{"id":19,"number":"2201","fullname":"Ashan Pathum ","callingname":"Ashan","photo":null,"dobirth":"2001-01-01","nic":null,"address":"Dabulla","mobile":"0779666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":20,"number":"2202","fullname":"Dileesha Rukmal","callingname":"Rukmal","photo":null,"dobirth":"2001-01-02","nic":null,"address":"Kagalle ","mobile":"0769666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":21,"number":"2203","fullname":"Thahir Imran","callingname":"Imran ","photo":null,"dobirth":"2001-01-03","nic":null,"address":"Waththala","mobile":"0719666666","land":null,"doassignment":"2022-11-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":22,"number":"2301","fullname":"Lakshan Ruwinda","callingname":"Lakshan ","photo":null,"dobirth":"2001-01-04","nic":null,"address":"Kirindiwala","mobile":"0729666666","land":null,"doassignment":"2023-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}, {"id":23,"number":"2302","fullname":"Arshad Ahamad","callingname":"Arshad","photo":null,"dobirth":"2001-01-05","nic":null,"address":"Galle ","mobile":"0759666666","land":null,"doassignment":"2023-01-01","description":"BIT/BSC UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}]
```

localhost:8080/employees?genderid=1&number=2201

```
[{"id":19,"number":"2201","fullname":"Ashan Pathum ","callingname":"Ashan","photo":null,"dobirth":"2001-01-01","nic":null,"address":"Dabulla","mobile":"0779666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}]
```

localhost:8080/employees?genderid=1&number=2202

```
[{"id":20,"number":"2202","fullname":"Dileesha Rukmal","callingname":"Rukmal","photo":null,"dobirth":"2001-01-02","nic":null,"address":"Kagalle ","mobile":"0769666666","land":null,"doassignment":"2022-01-01","description":"BIT UG","gender":{"id":1,"name":"Male"},"designation":{"id":1,"name":"Demonstrator L1"},"employeeestatus":{"id":1,"name":"Active "}}]
```

Regression Testing : Test all possible combinations including all employees

- All,
- By Number,
- By Gender ID,
- By Number & Gender ID

(4) Convert the above Controller to Filter employees both based on "gender", "number" & "fullname"

```
@GetMapping(produces = "application/json")
public List<Employee> get(@RequestParam HashMap<String, String> params) {

    String number = params.get("number");
    String genderid= params.get("genderid");
    String fullname= params.get("fullname");

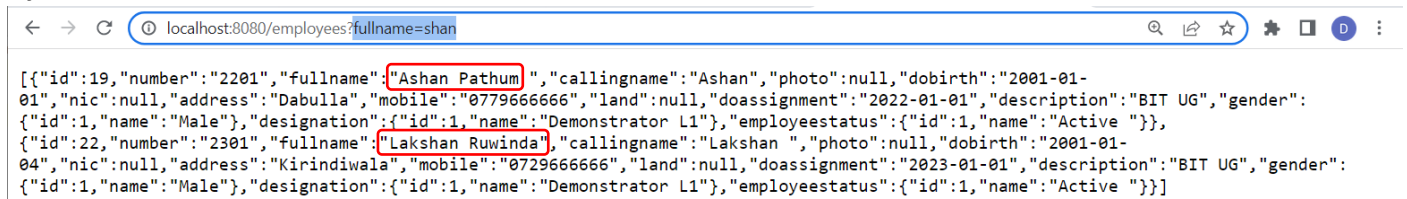
    List<Employee> employees = this.employeeDao.findAll();

    if(params.isEmpty()) return employees;

    employees = employees.stream().filter(
        employee -> {
            if(number!=null) return employee.getNumber().equals(number);
            if(genderid!=null) return employee.getGender().getId()==Integer.parseInt(genderid);
            if(fullname!=null) return employee.getFullname().contains(fullname);
            return false;
        }).collect(Collectors.toList());

    return employees;
}
```

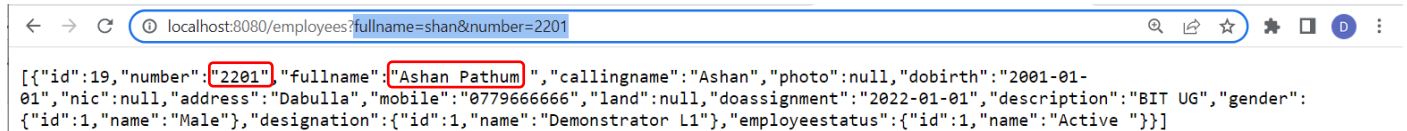
By Fullname



localhost:8080/employees?fullname=shan

```
[{"id":19,"number":"2201","fullname":"Ashan Pathum", "callingname":"Ashan", "photo":null, "dobirth":"2001-01-01", "nic":null, "address":"Dabulla", "mobile":"0779666666", "land":null, "doassignment":"2022-01-01", "description":"BIT UG", "gender":{"id":1, "name":"Male"}, "designation":{"id":1, "name":"Demonstrator L1"}, "employeeestatus":{"id":1, "name":"Active "}}, {"id":22,"number":"2301","fullname":"Lakshan Ruwinda", "callingname":"Lakshan", "photo":null, "dobirth":"2001-01-04", "nic":null, "address":"Kirindiwala", "mobile":"0729666666", "land":null, "doassignment":"2023-01-01", "description":"BIT UG", "gender":{"id":1, "name":"Male"}, "designation":{"id":1, "name":"Demonstrator L1"}, "employeeestatus":{"id":1, "name":"Active "}}]
```

By Fullname & Number



localhost:8080/employees?fullname=shan&number=2201

```
[{"id":19,"number":"2201","fullname":"Ashan Pathum", "callingname":"Ashan", "photo":null, "dobirth":"2001-01-01", "nic":null, "address":"Dabulla", "mobile":"0779666666", "land":null, "doassignment":"2022-01-01", "description":"BIT UG", "gender":{"id":1, "name":"Male"}, "designation":{"id":1, "name":"Demonstrator L1"}, "employeeestatus":{"id":1, "name":"Active "}}]
```

Regression Testing – Run all the previous testing relevant to the change

- All,
- By Number,
- By Gender ID,
- By Fullname
- By Number & Gender ID
- By Number & Fullname
- By Gender ID & Fullname
- By Gender ID & Fullname & Number

(5) Repeat the Process with Designation ID

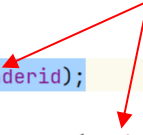
Use Copy and Paste the most relevant code segment strategy to enhance coding speed

```
String number = params.get("number");
String genderid= params.get("genderid");
String fullname= params.get("fullname");
String designationid= params.get("designationid");
```

```

employee -> {
    if(number!=null) return employee.getNumber().equals(number);
    if(genderid!=null) return employee.getGender().getId()==Integer.parseInt(genderid);
    if(fullname!=null) return employee.getFullname().contains(fullname);
    if(designationid!=null) return employee.getDesignation().getId()==Integer.parseInt(designationid);
    return false;
}).collect(Collectors.toList());

```



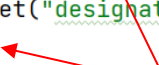
(6) Repeat the Process with NIC

(Before testing with NIC, insert values into NIC column in the Database if do not have)

```

String number = params.get("number");
String genderid= params.get("genderid");
String fullname= params.get("fullname");
String designationid= params.get("designationid");
String nic= params.get("nic");

```



```

employee -> {
    if(number!=null) return employee.getNumber().equals(number);
    if(genderid!=null) return employee.getGender().getId()==Integer.parseInt(genderid);
    if(fullname!=null) return employee.getFullname().contains(fullname);
    if(designationid!=null) return employee.getDesignation().getId()==Integer.parseInt(designationid);
    if(nic!=null) return employee.getFullname().contains(nic);
    return false;
}).collect(Collectors.toList());

```

