

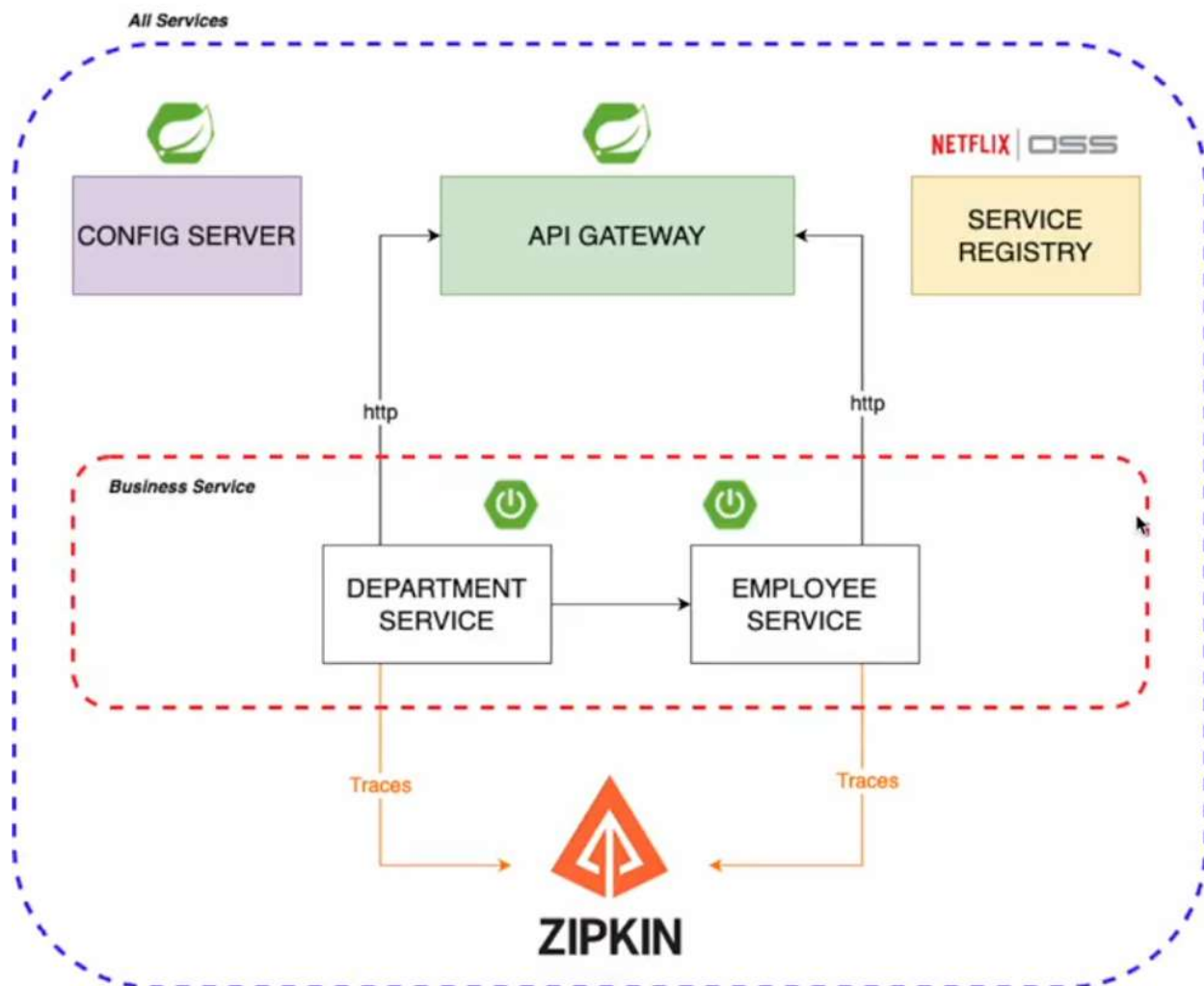
COURSE- SEZG533 SERVICE ORIENTED COMPUTING
ASSIGNMENT- DESIGN AND IMPLEMENTATION OF AN SOA APPLICATION

OBJECTIVE:

To design and implement a service-based application for Customer Care Portal.

PROBLEM STATEMENT:

PART 1: DESIGN



1. Identify the basic capabilities of the application:

The basic capabilities of the application revolve around managing employees and departments.

Key functionalities include:

- **Employee Management:**

- Add a new employee
- Retrieve information about an employee
- Update employee details
- Delete an employee

- **Department Management:**

- Add a new department
- Retrieve information about a department
- Update department details
- Delete a department
- Retrieve a list of all employees in a department

2. Entities and Models:

- **Department:**

- Properties: 'id', 'name', 'employees' (this is a list of employees in the department)

Department.java

```
public class Department {  
    private Long id;  
    private String name;  
    private List<Employee> employees = new ArrayList<>();  
  
    public Department() {  
    }  
}
```

- ****Employee:****

- Properties: `id`, `name`, `age`, `position`, `departmentId`

Employee.java

```
public record Employee(Long id, Long departmentId, String name, int age, String position) {  
    |  
}
```

3. Resources and Operations:

Employee Resource:

- ****Resource Name:**** `/employees`

- ****Operations:****

- `GET /employees`: Retrieve all employees
- `GET /employees/{employeeId}`: Retrieve a specific employee
- `POST /employees`: Create a new employee
- `PUT /employees/{employeeId}`: Update a specific employee
- `DELETE /employees/{employeeId}`: Delete a specific employee

Department Resource:

- ****Resource Name:**** `/departments`

- ****Operations:****

- `GET /departments`: Retrieve all departments
- `GET /departments/{departmentId}`: Retrieve a specific department
- `POST /departments`: Create a new department
- `PUT /departments/{departmentId}`: Update a specific department
- `DELETE /departments/{departmentId}`: Delete a specific department
- `GET /departments/{departmentId}/employees`: Retrieve all employees in a specific department

4. URI Design, HTTP Methods, Headers:

- URI Design: Follows the RESTful conventions outlined above.
- HTTP Methods: `GET`, `POST`, `PUT`, `DELETE`.
- Headers: Common headers like `Content-Type` for specifying the format of the request or response body (e.g., JSON).

5. Representations:

- **Accepted Representations (Request):**

- For operations like `POST` and `PUT`, the client may send JSON representations of the entities in the request body.

- **Served Representations (Response):**

- Responses for `GET` operations may include JSON representations of the retrieved entities.
- Successful `POST`, `PUT`, and `DELETE` operations may return a simple success message or the updated/deleted entity.

6. Resource Categorization:

- **Employee Resource Type:**

- `Collection Resource`: `/employees` (for managing a collection of employees)
- `Singleton Resource`: `/employees/{employeeId}` (for managing a single employee)

- **Department Resource Type:**

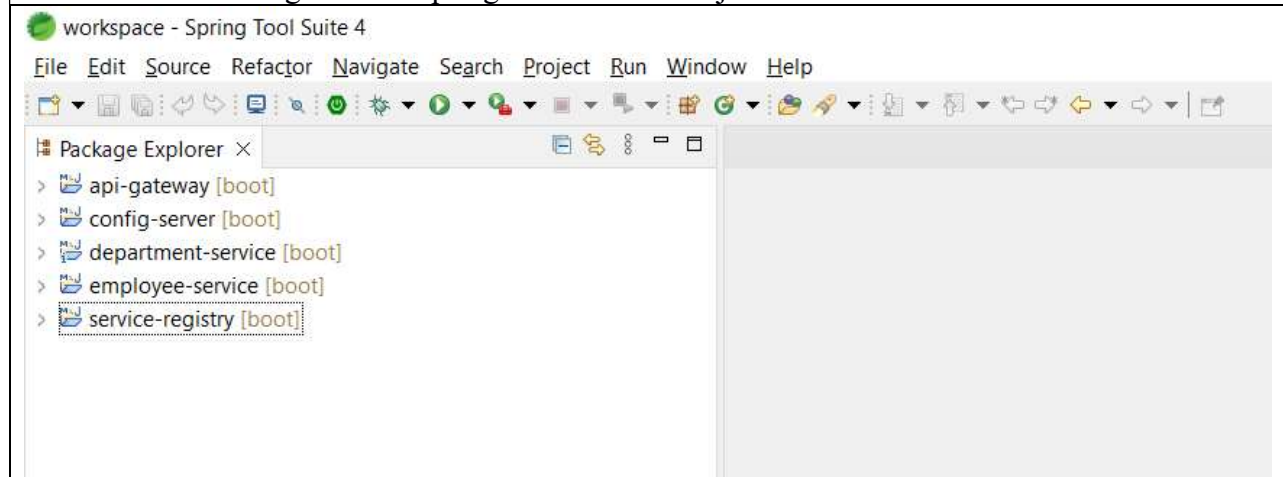
- `Collection Resource`: `/departments` (for managing a collection of departments)
- `Singleton Resource`: `/departments/{departmentId}` (for managing a single department)
- `Subordinate Resource`: `/departments/{departmentId}/employees` (for managing employees within a specific department)

PART 2.1: IMPLEMENTATION OF SERVICE ENDPOINTS

Technology Stack used :

- Spring Boot 3
- Java 17
- Spring Framework
- Spring Cloud Netflix OSS
- API Gateway
- Spring Config Server
- Spring Eureka discovery Server

5 microservices designed as 5 Spring Boot Maven Projects



DepartmentController.java

workspace - department-service/src/main/java/com/socassignment/departmentservice/controller/DepartmentController.java - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer

- api-gateway [boot]
- config-server [boot]
- department-service [boot]
 - src/main/java
 - com.socassignment.departmentservice
 - com.socassignment.departmentservice.client
 - com.socassignment.departmentservice.config
 - com.socassignment.departmentservice.controller
 - DepartmentController.java
 - com.socassignment.departmentservice.model
 - com.socassignment.departmentservice.repository
 - src/main/resources
 - application.yaml
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - target/generated-sources/annotations
 - src
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml
- employee-service [boot]
- service-registry [boot]

Boot Dashboard

Type tags, projects, or working set names to match (incl. * and ? wild)

- local

```

13 @RestController
14 @RequestMapping("/department")
15 public class DepartmentController {
16
17     private static final Logger LOGGER
18         = LoggerFactory.getLogger(DepartmentController.class);
19
20     @Autowired
21     private DepartmentRepository repository;
22
23     @Autowired
24     private EmployeeClient employeeClient;
25
26     @PostMapping
27     public Department add(@RequestBody Department department) {
28         LOGGER.info("Department add: {}", department);
29         return repository.addDepartment(department);
30     }
31
32     @GetMapping
33     public List<Department> findAll() {
34         LOGGER.info("Department find");
35         return repository.findAll();
36     }
37
38     @GetMapping("/{id}")
39     public Department findById(@PathVariable Long id) {
40         LOGGER.info("Department find: id={}", id);
41         return repository.findById(id);
42     }
43
44     @GetMapping("/with-employees")
45     public List<Department> findAllWithEmployees() {
46         LOGGER.info("Department find");
47         List<Department> departments
48             = repository.findAll();
49         departments.forEach(department ->
50             department.setEmployees(
51                 employeeClient.findByDepartment(department.getId())));
52         return departments;
53     }
54 }
55

```


EmployeeController.java

```
workspace - employee-service/src/main/java/com/socassignment/employeeservice/controller/EmployeeController.java - Spring Tool Suite 4
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer
  > api-gateway [boot]
  > config-server [boot]
  > department-service [boot]
  > employee-service [boot]
    > src/main/java
      > com.socassignment.employeeservice
        > com.socassignment.employeeservice.controller
          > EmployeeController.java
        > com.socassignment.employeeservice.model
        > com.socassignment.employeeservice.repository
      > src/main/resources
      > src/test/java
      > JRE System Library [JavaSE-17]
      > Maven Dependencies
      > target/generated-sources/annotations
      > src
      > target
        mvnw
        mvnw.cmd
        pom.xml
      > service-registry [boot]
Boot Dashboard
  Type tags, projects, or working set names to match (incl. * and ? wild)
  > local

EmployeeController.java
1 package com.socassignment.employeeservice.controller;
2
3 import com.socassignment.employeeservice.model.Employee;
4
5
6
7
8
9
10
11
12 @RestController
13 @RequestMapping("/employee")
14 public class EmployeeController {
15     private static final Logger LOGGER
16         = LoggerFactory.getLogger(EmployeeController.class);
17
18     @Autowired
19     EmployeeRepository repository;
20
21     @PostMapping
22     public Employee add(@RequestBody Employee employee) {
23         LOGGER.info("Employee add: {}", employee);
24         return repository.add(employee);
25     }
26
27     @GetMapping
28     public List<Employee> findAll() {
29         LOGGER.info("Employee find");
30         return repository.findAll();
31     }
32
33     @GetMapping("/{id}")
34     public Employee findById(@PathVariable("id") Long id) {
35         LOGGER.info("Employee find: id={}", id);
36         return repository.findById(id);
37     }
38
39     @GetMapping("/department/{departmentId}")
40     public List<Employee> findByDepartment(@PathVariable("departmentId") Long departmentId) {
41         LOGGER.info("Employee find: departmentId={}", departmentId);
42         return repository.findByDepartment(departmentId);
43     }
44 }
45
46
```

7 Endpoints implemented

▼ SOC Assignment

- GET Get All Departments
- POST Post Add Department
- GET Get Department By ID
- GET Get All Departments WITH EM...
- GET Get All Employees
- POST Post Add Employee
- GET Get Employee By ID

Endpoints Request & Response Screenshots –

GET Get All Departments + ... No Environment

SOC Assignment / Get All Departments Save

GET http://localhost:8060/department Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Headers 6 hidden

Key	Value	Description	...	Bulk Edit	Presets
Key	Value	Description			

Body Cookies Headers (3) Test Results Status: 200 OK Time: 19 ms Size: 238 B Save as example

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": 1,
4     "name": "IT",
5     "employees": [
6       {
7         "id": 111,
8         "departmentId": 1,
9         "name": "John Doe",
10        "age": 25,
11        "position": "Software Engineer"
12      }
13    ]
14  }
15 ]

```

POST Post Add Department + ...

SOC Assignment / Post Add Department

POST http://localhost:8060/department

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "id": 2,
3   "name": "CS",
4   "employees": []
5 }

```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 15 n

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 2,
3   "name": "CS",
4   "employees": []
5 }

```


GET Get Department By ID + ...

HTTP SOC Assignment / Get Department By ID

GET http://localhost:8060/department/2

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 2,
3   "name": "CS",
4   "employees": []
5 }
```

GET Get All Departments WITH + ... No Environment

HTTP SOC Assignment / Get All Departments WITH EMPLOYEES Save

GET http://localhost:8060/department/with-employees

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 30 ms Size: 274 B

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": 1,
4     "name": "IT",
5     "employees": [
6       {
7         "id": 111,
8         "departmentId": 1,
9         "name": "John Doe",
10        "age": 25,
11        "position": "Software Engineer"
12      }
13    ]
14  },
15  {
16    "id": 2,
17    "name": "CS",
18    "employees": []
19  }
20 ]
```

Student Name : SHELKE AKSHAY NANDKUMAR
 Student ID : 2022mt93331
 Student Email : 2022mt93331@wilp.bits-pilani.ac.in
 Course : SEZG533 SERVICE ORIENTED COMPUTING

Assignment : DESIGN AND IMPLEMENTATION OF AN SOA APPLICATION



GET Get All Employees + ... No Environment

SOC Assignment / Get All Employees Save

GET http://localhost:8060/employee Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Headers 6 hidden

Key	Value	Description	...	Bulk Edit	Presets
Key	Value	Description			

Body Cookies Headers (3) Test Results Status: 200 OK Time: 14 ms Size: 203 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 111,
3   "departmentId": 1,
4   "name": "John Doe",
5   "age": 25,
6   "position": "Software Engineer"
7 }
  
```

POST Post Add Employee + ... No Environment

SOC Assignment / Post Add Employee Save

POST http://localhost:8060/employee Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

Body Cookies Headers (3) Test Results Status: 200 OK Time: 12 ms Size: 206 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 222,
3   "departmentId": 2,
4   "name": "Akshay Shelke",
5   "age": 30,
6   "position": "Software Engineer"
7 }
  
```

Student Name : SHELKE AKSHAY NANDKUMAR

Student ID : 2022mt93331

Student Email : 2022mt93331@wilp.bits-pilani.ac.in

Course : SEZG533 SERVICE ORIENTED COMPUTING

Assignment : DESIGN AND IMPLEMENTATION OF AN SOA APPLICATION



GET Get Employee By ID + ... No Environment

HTTP SOC Assignment / Get Employee By ID Save Send

GET http://localhost:8060/employee/222

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (3) Test Results Status: 200 OK Time: 15 ms Size: 206 B Save as example

Pretty Raw Preview Visualize JSON

```

1  {
2    "id": 222,
3    "departmentId": 2,
4    "name": "Akshay Shelke",
5    "age": 30,
6    "position": "Software Engineer"
7  }
  
```

PART 2.2: API GATEWAY

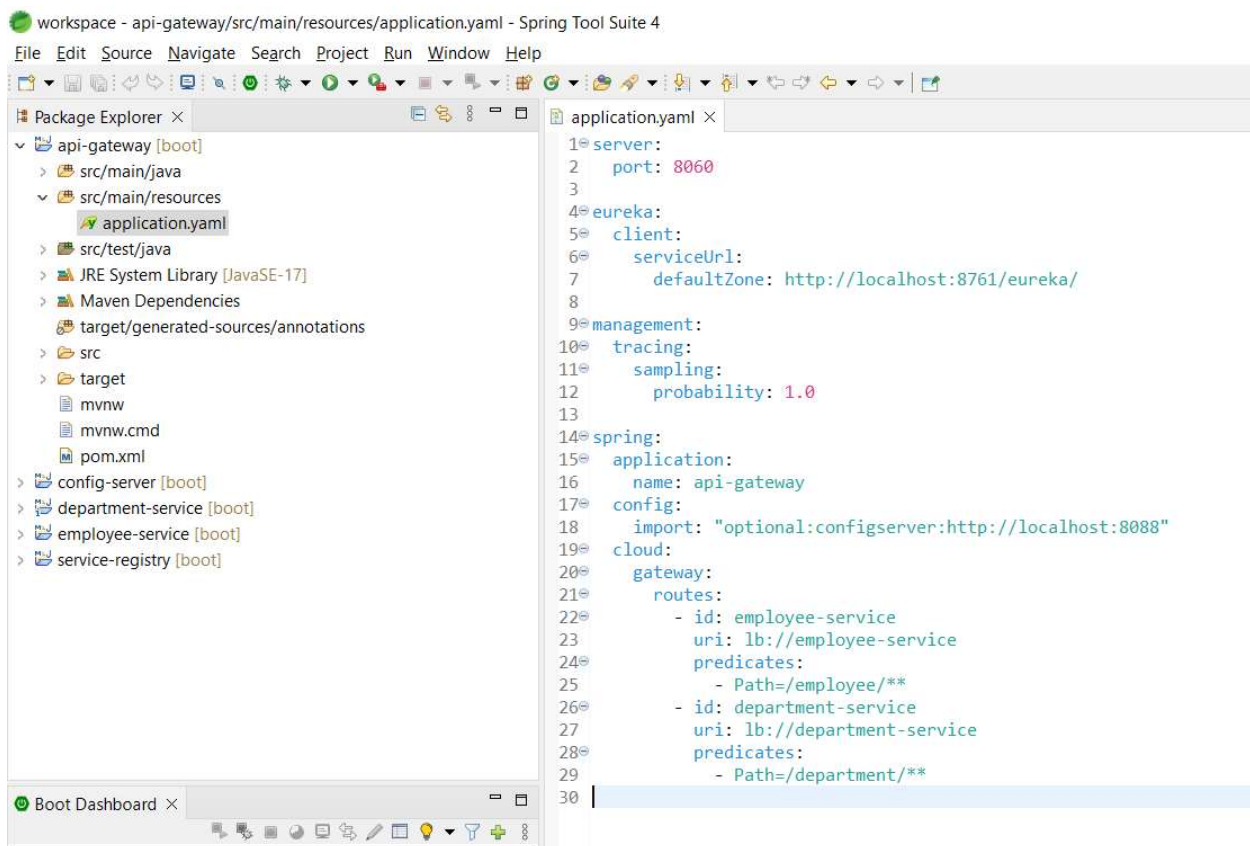
As part of the implementation process for the service-oriented application, I have designed API Gateway which is running on 8060.

Department Service is running on 8081

Employee Service is running on 8082

Eureka Discovery Service is running on 8761

And API GATEWAY is running on 8060, and all routing is defined in API GATEWAY application.yaml file.



workspace - api-gateway/src/main/resources/application.yaml - Spring Tool Suite 4

File Edit Source Navigate Search Project Run Window Help

Package Explorer ×

- api-gateway [boot]
 - src/main/java
 - src/main/resources
 - application.yaml
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - target/generated-sources/annotations
 - src
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml
 - config-server [boot]
 - department-service [boot]
 - employee-service [boot]
 - service-registry [boot]

application.yaml ×

```

1 server:
2   port: 8060
3
4 eureka:
5   client:
6     serviceUrl:
7       defaultZone: http://localhost:8761/eureka/
8
9 management:
10  tracing:
11    sampling:
12      probability: 1.0
13
14 spring:
15   application:
16     name: api-gateway
17   config:
18     import: "optional:configserver:http://localhost:8088"
19   cloud:
20     gateway:
21       routes:
22         - id: employee-service
23           uri: lb://employee-service
24           predicates:
25             - Path=/employee/**
26         - id: department-service
27           uri: lb://department-service
28           predicates:
29             - Path=/department/**
30
  
```

Boot Dashboard ×

workspace - config-server/src/main/resources/config/department-service.yaml - Spring Tool Suite 4

File Edit Source Navigate Search Project Run Window Help

Package Explorer ×

- > api-gateway [boot]
- > config-server [boot]
 - > src/main/java
 - > src/main/resources
 - > config
 - department-service.yaml
 - employee-service.yaml
 - application.yaml
 - > src/test/java
 - > JRE System Library [JavaSE-17]
 - > Maven Dependencies
 - target/generated-sources/annotations
 - > src
 - > target
 - mvnw
 - mvnw.cmd
 - pom.xml
 - > department-service [boot]
 - > employee-service [boot]
 - > service-registry [boot]

department-service.yaml × employee-service.yaml

```

1 server:
2   port: 8081
3
4 eureka:
5   client:
6     serviceUrl:
7       defaultZone: http://localhost:8761/eureka/
8
9 management:
10  tracing:
11    sampling:
12      probability: 1.0
  
```

workspace - config-server/src/main/resources/config/employee-service.yaml - Spring Tool Suite 4

File Edit Source Navigate Search Project Run Window Help

Package Explorer ×

- > api-gateway [boot]
- > config-server [boot]
 - > src/main/java
 - > src/main/resources
 - > config
 - department-service.yaml
 - employee-service.yaml
 - application.yaml
 - > src/test/java
 - > JRE System Library [JavaSE-17]
 - > Maven Dependencies
 - target/generated-sources/annotations
 - > src
 - > target
 - mvnw
 - mvnw.cmd
 - pom.xml
 - > department-service [boot]
 - > employee-service [boot]
 - > service-registry [boot]

department-service.yaml employee-service.yaml ×

```

1 server:
2   port: 8082
3
4 eureka:
5   client:
6     serviceUrl:
7       defaultZone: http://localhost:8761/eureka/
8
9 management:
10  tracing:
11    sampling:
12      probability: 1.0
  
```

Student Name : SHELKE AKSHAY NANDKUMAR
Student ID : 2022mt93331
Student Email : 2022mt93331@wilp.bits-pilani.ac.in
Course : SEZG533 SERVICE ORIENTED COMPUTING
Assignment : DESIGN AND IMPLEMENTATION OF AN SOA APPLICATION



Important Links

Github Link : <https://github.com/ShelkeAkshay-2022mt93331/SOC-Assignment-2022mt93331>

Demo Video Link (Google Drive Folder) : https://drive.google.com/drive/folders/1qwNI56-kG9ihDYcu_jbXeMsG_hKiTbuS?usp=sharing