

Skanda Prasad K S
1DT20IS093
BATCH-01
SPRINGS FINAL ASSESSMENT

1.Employee-Registry

application .properties

```
spring.datasource.url=jdbc:postgresql://localhost:5432/Employee_Database
spring.datasource.username=postgres
spring.datasource.password=Candy648
spring.datasource.driver-class-name=org.postgresql.Driver
```

```
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto = update
spring.datasource.initialization-mode=always
```

```
server.port = 8080
```

```
eureka.client.service-url.default-zone=http://localhost:8761/eureka/
```

```
management.tracing.sampling.probability=1.0
```

```
spring.application.name=EMPLOYEE-REGISTRY
```

Main.class

```
package com.infosys;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
import org.springframework.cloud.openfeign.EnableFeignClients;

@SpringBootApplication
@EnableDiscoveryClient
@EnableFeignClients
public class EmployeeServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(EmployeeServiceApplication.class, args);
    }
}
```

```
}
```

EmployeeEntity.java

```
package com.infosys.entity;
```

```
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;
```

```
@Data
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
@ToString
```

```
@Entity
```

```
@Table(name="employee_service")
```

```
public class Employee {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Long Employee_id;
```

```
    @Column
```

```
    private String Employee_name;
```

```
    @Column
```

```
    private String Job;
```

```
    @Column
```

```
    private String Data_of_joining;
```

```
    @Column
```

```
    private Long salary;
```

```
}
```

Project entity.java

```
package com.infosys.entity;
```

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;
```

```
@Data
@NoArgsConstructor
@AllArgsConstructor
@ToString
public class Project {
```

```
    private Long project_id;

    private String project_title;

    private String project_language;

    private Long employee_id;
```

```
}
```

Resource handling

```
package com.infosys.exception;
```

```
public class ResourceNotFoundException extends RuntimeException {
```

```
    public ResourceNotFoundException(String s)
    {
        super(s);
    }
    public ResourceNotFoundException()
    {
        super("Resource not found");
    }
}
```

```
}
```

Controller.java

```
package com.infosys.controller;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
import com.infosys.entity.Employee;
import com.infosys.entity.Project;
import com.infosys.service.EmployeeService;
import com.infosys.service.ProjectClient;
```

```
@RestController
```

```
@RequestMapping("/employee")
```

```
public class EmployeeController {
```

```
    @Autowired
```

```
    private EmployeeService ES;
```

```
    @Autowired
```

```
    private ProjectClient PC;
```

```
    @PostMapping
```

```
    public void save(@RequestBody Employee e)
```

```
    {
```

```
        ES.createEmployee(e);
```

```
    }
```

```
    @GetMapping
```

```
    public List<Employee> getAll()
```

```
    {
```

```
        return ES.getAll();
```

```
    }
```

```
    @GetMapping("/{id}")
```

```
    public Employee getById(@PathVariable Long id)
```

```
    {
```

```
        return ES.getEmployeeById(id);
```

```
    }
```

```
    @GetMapping("/{date}")
```

```
    public List<Employee> getByDate(@PathVariable String date)
```

```
    {
```

```
        return ES.getByDate(date);
```

```
    }
```

```

        @GetMapping("/project/{id}")
        public List<Project> getProjectById(@PathVariable Long id)
        {
            return PC.getAllProjectById(id);
        }
    }
}

```

Service.java

```

package com.infosys.service;

import java.util.List;

import com.infosys.entity.Employee;

public interface EmployeeService {

    public void createEmployee(Employee e);

    public List<Employee> getAll();

    public Employee getEmployeeById(Long id);

    public List<Employee> getByDate(String date);

}

```

ServiceImpl.java

```

package com.infosys.service;

import java.util.List;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.infosys.entity.Employee;
import com.infosys.repository.EmployeeRepository;

@Service
public class EmployeeServiceImpl implements EmployeeService{

    @Autowired
    private EmployeeRepository ER;

    @Override

```

```

    public void createEmployee(Employee e) {
        ER.save(e);
    }

    @Override
    public List<Employee> getAll() {

        return ER.findAll();
    }

    @Override
    public Employee getEmployeeById(Long id) {
        return ER.findById(id).orElseThrow(()->new RuntimeException("Employee not
found"));
    }

    @Override
    public List<Employee> getByDate(String date) {

        List<Employee> employees=ER.findAll();

        return
employees.stream().filter(employee->employee.getData_of_joining()==date).collect(Collectors.toList());
    }

}

```

Projectclient.java

```

package com.infosys.service;

import java.util.List;

import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

import com.infosys.entity.Project;
@FeignClient(url="http://localhost:8081",name="PROJECR-REGISTRY")
public interface ProjectClient {

    @GetMapping("/employee/project/{id}")
    List<Project> getAllProjectById(@PathVariable Long id);
}

```

```
}
```

EmpRepository.java

```
package com.infosys.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import com.infosys.entity.Employee;
```

```
public interface EmployeeRepository extends JpaRepository<Employee, Long>{
```

```
}
```

2.Project-registry

project-entity.java

```
package com.infosys.Entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;

@Data
@NoArgsConstructor
@AllArgsConstructor
@ToString
@Entity
@Table(name="project_service")
public class Project {

    @Id
    private Long project_id;
    @Column
    private String project_title;
    @Column
    private String project_language;
    @Column
    private Long employee_id;

}
```

Main.java

```
package com.infosys;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication
@EnableDiscoveryClient
public class ProjectServiceApplication {
```



```

        public static void main(String[] args) {
            SpringApplication.run(ProjectServiceApplication.class, args);
        }
    }
}

```

Controller.java

```

package com.infosys.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.infosys.Entity.Project;
import com.infosys.service.ProjectService;

@RestController
@RequestMapping("/project")
public class ProjectController {

    @Autowired
    public ProjectService PS;

    @PostMapping
    public void CreateProject(@RequestBody Project p)
    {
        PS.Create(p);
    }

    @GetMapping
    public List<Project> getAll()
    {
        return PS.getAll();
    }

    @GetMapping("/{id}")
    public Project Geybyid(@PathVariable Long id)
    {
        return PS.getProjectById(id);
    }
}

```

```

    @GetMapping("/employee/{Eid}")
    public List<Project> getProEmplId(@PathVariable Long Eid)
    {
        return PS.getProjectByEmployeeId(Eid);
    }
}

```

ExceptionHandler.java

```

package com.infosys.exception;

public class ResourceNotFoundException extends RuntimeException {

    public ResourceNotFoundException(String s)
    {
        super(s);
    }
    public ResourceNotFoundException()
    {
        super("Resource not found");
    }

}

```

Repository.java

```

package com.infosys.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.infosys.Entity.Project;

public interface ProjectRepository extends JpaRepository<Project,Long> {

}

```

Service.java

```

package com.infosys.service;

import java.util.List;

import com.infosys.Entity.Project;

public interface ProjectService {

```

```

        public void Create(Project p);

        public List<Project> getAll();

        public Project getProjectById(Long id);

        public List<Project> getprojectByEmployeeId(Long id);

    }

```

ServiceImpl.java

```

package com.infosys.service;

import java.util.List;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.infosys.Entity.Project;
import com.infosys.repository.ProjectRepository;

@Service
public class ProjectServiceImpl implements ProjectService{

    @Autowired
    private ProjectRepository PR;

    @Override
    public void Create(Project p) {
        PR.save(p);
    }

    @Override
    public List<Project> getAll() {
        return PR.findAll();
    }

    @Override
    public Project getProjectById(Long id) {
        if(PR.findById(id).isEmpty())
        {
            return null;
        }
        else

```

```

        {
            return PR.findById(id).get();
        }
    }

    @Override
    public List<Project> getprojectByEmployeeId(Long id) {
        List<Project> pro=PR.findAll();
        return
        pro.stream().filter(project->project.getEmployee_id()==id).collect(Collectors.toList());
    }
}

```

Application.properties

```

spring.datasource.url=jdbc:postgresql://localhost:5432/Employee_Database
spring.datasource.username=postgres
spring.datasource.password=Candy648
spring.datasource.driver-class-name=org.postgresql.Driver

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto = update
spring.datasource.initialization-mode=always

server.port = 8081

eureka.client.service-url.default-zone=http://localhost:8761/eureka/

management.tracing.sampling.probability=1.0

spring.application.name=PROJECR-REGISTRY

```

3.service-Registry

Main.java

```
package com.infosys;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication
@EnableEurekaServer
public class ServiceRegistryApplication {

    public static void main(String[] args) {
        SpringApplication.run(ServiceRegistryApplication.class, args);
    }

}
```

Application.props

server.port=8761

spring.application.name=SERVICE-REGISTRY

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

4.API-GATEWAY

```
package com.infosys;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication
@EnableDiscoveryClient
public class ApiGatewayApplication {

    public static void main(String[] args) {
        SpringApplication.run(ApiGatewayApplication.class, args);
    }

}

Application.props

server.port=8083

spring.application.name=API_GATEWAY

#routes configuration

spring.cloud.gateway.routes[0].id=EMPLOYEE-REGISTRY
spring.cloud.gateway.routes[0].uri=lb://EMPLOYEE-REGISTRY
spring.cloud.gateway.routes[0].predicates[0]=Path=/employee/**

spring.cloud.gateway.routes[1].id=PROJECR-REGISTRY
spring.cloud.gateway.routes[1].uri=lb://PROJECR-REGISTRY
spring.cloud.gateway.routes[1].predicates[0]=Path=/project/**

eureka.instance.client.serverUrl.defaultZone=http://localhost:8761/eureka/
```