Department service( spring boot starter project)

Src/main/java

1.com.example.demo

DepartmentServiceApplication

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DepartmentServiceApplication {

public static void main(String[] args) {

SpringApplication.run(DepartmentServiceApplication.class, args);

}

}

.controller

DepartmentController

package com.example.demo.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.DeleteMapping;

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.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.example.demo.exceptions.DepartmentNotFound;

import com.example.demo.model.Department;

import com.example.demo.service.DepartmentService;

@RestController

@RequestMapping("/department")

public class DepartmentController {

@Autowired

DepartmentService service;

@PostMapping("/save")

public String saveDepartment(@RequestBody Department department) {

return service.saveDepartment(department);

}

@PutMapping("/update")

public Department updateDepartment(@RequestBody Department department) {

return service.updateDepartment(department);

}

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

public Department getDepartment(@PathVariable ("id") int departmentId) throws DepartmentNotFound {

return service.getDepartment(departmentId);

}

@GetMapping("/fetchAll")

public List<Department> getAllDepartments() {

return service.getAllDepartments();

}

}

.dta

EmpDeptDeleteDTO

package com.example.demo.dto;

import com.example.demo.model.Department;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@AllArgsConstructor

@NoArgsConstructor

@Data

public class EmpDeptDeleteDTO {

private Employee emp;

private Department dept;

}

Employee

package com.example.demo.dto;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Entity

@Table(name = "employees\_info")

@NoArgsConstructor

@AllArgsConstructor

@Data

public class Employee {

@Id

private int employeeId;

private String employeeName;

private int employeeSalary;

private int departmentId;

}

.model

Department

package com.example.demo.model;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import lombok.Data;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Data

@Entity

@Table(name = "department\_info")

public class Department {

@Id

@Column(name = "deptno")

private int departmentId;

private String departmentName;

private String departmentLocation;

}

.repository

DepartmentRepository

package com.example.demo.repository;

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

import com.example.demo.model.Department;

public interface DepartmentRepository extends JpaRepository<Department, Integer> {

}

.service

DepartmentService

package com.example.demo.service;

import java.util.List;

import com.example.demo.dto.EmpDeptDeleteDTO;

import com.example.demo.exceptions.DepartmentNotFound;

import com.example.demo.model.Department;

public interface DepartmentService {

public abstract String saveDepartment(Department department);

public abstract Department updateDepartment(Department department);

public abstract Department getDepartment(int departmentId) throws DepartmentNotFound;

public abstract List<Department> getAllDepartments();

public abstract String deleteDepartment(EmpDeptDeleteDTO employeeDept);

}

DepartmentServiceImp

package com.example.demo.service;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.example.demo.dto.EmpDeptDeleteDTO;

import com.example.demo.exceptions.DepartmentNotFound;

import com.example.demo.model.Department;

import com.example.demo.repository.DepartmentRepository;

@Service

public class DepartmentServiceImp implements DepartmentService {

@Autowired

DepartmentRepository repository;

@Override

public String saveDepartment(Department department) {

repository.save(department);

return "Saved Successfully";

}

@Override

public Department updateDepartment(Department department) {

return repository.save(department);

}

@Override

public Department getDepartment(int departmentId) throws DepartmentNotFound {

Optional<Department> optional = repository.findById(departmentId);

if(optional.isPresent())

return optional.get();

else

throw new DepartmentNotFound("Department id is not valid");

}

@Override

public List<Department> getAllDepartments() {

return repository.findAll();

}

@Override

public String deleteDepartment(EmpDeptDeleteDTO employeeDept) {

return null;

}

}

Pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.4.5</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>DepartmentService</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>DepartmentService</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2024.0.1</spring-cloud.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web-services</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-validation</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-config</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<configuration>

<excludes>

<exclude>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</exclude>

</excludes>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<annotationProcessorPaths>

<path>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</path>

</annotationProcessorPaths>

</configuration>

</plugin>

</plugins>

</build>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

</project>