**SKG\_LAB\_EMPLOYEE\_CONTAINS**

**Controller:**

package com.examly.springapp.controller;

import com.examly.springapp.model.Employee;

import com.examly.springapp.service.EmployeeService;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/employees")

public class EmployeeController {

private final EmployeeService employeeService;

@Autowired

public EmployeeController(EmployeeService employeeService) {

this.employeeService = employeeService;

}

@PostMapping

public ResponseEntity<Employee> createEmployee(@RequestBody Employee employee) {

Employee createdEmployee = employeeService.createEmployee(employee);

return new ResponseEntity<>(createdEmployee, HttpStatus.CREATED);

}

@GetMapping("/containing/{searchTerm}")

public ResponseEntity<List<Employee>> getEmployeesContaining(@PathVariable String searchTerm) {

List<Employee> employees = employeeService.getEmployeesContaining(searchTerm);

return new ResponseEntity<>(employees, HttpStatus.OK);

}

@GetMapping("/startsWith/{name}")

public ResponseEntity<List<Employee>> getEmployeesStartsWith(@PathVariable String name) {

List<Employee> employees = employeeService.getEmployeesStartsWith(name);

return new ResponseEntity<>(employees, HttpStatus.OK);

}

@GetMapping("/endsWith/{name}")

public ResponseEntity<List<Employee>> getEmployeesEndsWith(@PathVariable String name) {

List<Employee> employees = employeeService.getEmployeesEndsWith(name);

return new ResponseEntity<>(employees, HttpStatus.OK);

}

@GetMapping("/contains/{designation}")

public ResponseEntity<List<Employee>> getEmployeesByDesignation(@PathVariable String designation) {

List<Employee> employees = employeeService.getEmployeesByDesignation(designation);

return new ResponseEntity<>(employees, HttpStatus.OK);

}

@GetMapping("/isContaining/{name}")

public ResponseEntity<List<Employee>> getEmployeesByName(@PathVariable String name) {

List<Employee> employees = employeeService.getEmployeesByName(name);

return new ResponseEntity<>(employees, HttpStatus.OK);

}

}

**Model:**

package com.examly.springapp.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String designation;

public Employee() {}

public Employee(String name, String designation) {

this.name = name;

this.designation = designation;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDesignation() {

return designation;

}

public void setDesignation(String designation) {

this.designation = designation;

}

}

**Repository:**

package com.examly.springapp.repository;

import com.examly.springapp.model.Employee;

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

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

List<Employee> findByNameContainingOrDesignationContaining(String name, String designation);

List<Employee> findByNameStartingWith(String name);

List<Employee> findByNameEndingWith(String name);

List<Employee> findByDesignation(String designation);

List<Employee> findByNameContaining(String name);

}

**Service:**

package com.examly.springapp.service;

import com.examly.springapp.model.Employee;

import com.examly.springapp.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class EmployeeService {

private final EmployeeRepository employeeRepository;

@Autowired

public EmployeeService(EmployeeRepository employeeRepository) {

this.employeeRepository = employeeRepository;

}

public Employee createEmployee(Employee employee) {

return employeeRepository.save(employee);

}

public List<Employee> getEmployeesContaining(String searchTerm) {

return employeeRepository.findByNameContainingOrDesignationContaining(searchTerm, searchTerm);

}

public List<Employee> getEmployeesStartsWith(String name) {

return employeeRepository.findByNameStartingWith(name);

}

public List<Employee> getEmployeesEndsWith(String name) {

return employeeRepository.findByNameEndingWith(name);

}

public List<Employee> getEmployeesByDesignation(String designation) {

return employeeRepository.findByDesignation(designation);

}

public List<Employee> getEmployeesByName(String name) {

return employeeRepository.findByNameContaining(name);

}

}

**OUTPUT:**

