**SKG\_LAB\_EMPLOYEE:**

**Model:**

**Employee.java**

package com.examly.springapp.model;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Employee {

@Id

private int id;

private String name, designation;

private double salary;

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;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

}

**Repository**

**EmployeeRepo.java**

package com.examly.springapp.repository;

import java.util.List;

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

import com.examly.springapp.model.Employee;

public interface EmployeeRepo extends JpaRepository<Employee,Integer>{

List<Employee> findByName(String name);

List<Employee> findByDesignation(String designation);

List<Employee> findBySalaryBetween(double minSalary, double maxSalary);

}

**Service**

**EmployeeService.java**

package com.examly.springapp.service;

import java.util.\*;

import java.util.stream.Collectors;

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

import org.springframework.stereotype.Service;

import com.examly.springapp.model.Employee;

import com.examly.springapp.repository.EmployeeRepo;

@Service

public class EmployeeService {

@Autowired

EmployeeRepo er;

public Employee add(Employee ee)

{

return er.save(ee);

}

public List<Employee> getAll()

{

return er.findAll();

}

public Employee getById(int id)

{

return er.findById(id).orElse(null);

}

public Map<String, List<Employee>> groupByAttribute(String attribute) {

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

return employees.stream()

.collect(Collectors.groupingBy(emp -> {

switch (attribute) {

case "name":

return emp.getName();

case "designation":

return emp.getDesignation();

// Add more cases for other attributes if needed

default:

return "";

}

}));

}

public List<Employee> findByAttribute(String attribute, String value) {

switch (attribute) {

case "name":

return er.findByName(value);

case "designation":

return er.findByDesignation(value);

default:

return null;

}

}

public List<Employee> findBySalaryRange(double minSalary, double maxSalary) {

return er.findBySalaryBetween(minSalary, maxSalary);

}

}

**Controller**

**EmployeeController.java**

package com.examly.springapp.controller;

import java.util.\*;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RequestParam;

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

import com.examly.springapp.model.Employee;

import com.examly.springapp.service.EmployeeService;

@RestController

public class EmployeeController {

@Autowired

EmployeeService es;

@PostMapping("/employees")

public ResponseEntity<Employee> add(@RequestBody Employee ee)

{

Employee obj = es.add(ee);

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

}

@GetMapping("/employees")

public ResponseEntity<List<Employee>> getAll()

{

List<Employee> obj = es.getAll();

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

}

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

public ResponseEntity<Employee> getById(@PathVariable int id)

{

Employee obj = es.getById(id);

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

}

@GetMapping("/employees/groupBy/{attribute}")

public Map<String, List<Employee>> groupByAttribute(@PathVariable String attribute) {

return es.groupByAttribute(attribute);

}

@GetMapping("/employees/findBy/{attribute}")

public List<Employee> findByAttribute(@PathVariable String attribute, @RequestParam("value") String value) {

return es.findByAttribute(attribute, value);

}

@GetMapping("/employees/salaryRange")

public List<Employee> findBySalaryRange(@RequestParam("minSalary") double minSalary,

@RequestParam("maxSalary") double maxSalary) {

return es.findBySalaryRange(minSalary, maxSalary);

}

}

**Output:**

