package com.example.employeemanagementsystem.service;

import com.example.employeemanagementsystem.Employee;

import com.example.employeemanagementsystem.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

// Create or Update an Employee

public Employee saveEmployee(Employee employee) {

return employeeRepository.save(employee);

}

// Read all Employees

public List<Employee> getAllEmployees() {

return employeeRepository.findAll();

}

// Read an Employee by ID

public Optional<Employee> getEmployeeById(Long id) {

return employeeRepository.findById(id);

}

// Delete an Employee by ID

public void deleteEmployee(Long id) {

employeeRepository.deleteById(id);

}

}

package com.example.employeemanagementsystem.service;

import com.example.employeemanagementsystem.Department;

import com.example.employeemanagementsystem.repository.DepartmentRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class DepartmentService {

@Autowired

private DepartmentRepository departmentRepository;

// Create or Update a Department

public Department saveDepartment(Department department) {

return departmentRepository.save(department);

}

// Read all Departments

public List<Department> getAllDepartments() {

return departmentRepository.findAll();

}

// Read a Department by ID

public Optional<Department> getDepartmentById(Long id) {

return departmentRepository.findById(id);

}

// Delete a Department by ID

public void deleteDepartment(Long id) {

departmentRepository.deleteById(id);

}

}

package com.example.employeemanagementsystem.controller;

import com.example.employeemanagementsystem.Employee;

import com.example.employeemanagementsystem.service.EmployeeService;

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

import org.springframework.http.ResponseEntity;

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

import java.util.List;

import java.util.Optional;

@RestController

@RequestMapping("/api/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

// Create a new Employee

@PostMapping

public Employee createEmployee(@RequestBody Employee employee) {

return employeeService.saveEmployee(employee);

}

// Get all Employees

@GetMapping

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

// Get an Employee by ID

@GetMapping("/{id}")

public ResponseEntity<Employee> getEmployeeById(@PathVariable Long id) {

Optional<Employee> employee = employeeService.getEmployeeById(id);

return employee.map(ResponseEntity::ok).orElseGet(() -> ResponseEntity.notFound().build());

}

// Update an Employee

@PutMapping("/{id}")

public ResponseEntity<Employee> updateEmployee(@PathVariable Long id, @RequestBody Employee employeeDetails) {

Optional<Employee> employee = employeeService.getEmployeeById(id);

if (employee.isPresent()) {

employeeDetails.setId(id);

Employee updatedEmployee = employeeService.saveEmployee(employeeDetails);

return ResponseEntity.ok(updatedEmployee);

} else {

return ResponseEntity.notFound().build();

}

}

// Delete an Employee

@DeleteMapping("/{id}")

public ResponseEntity<Void> deleteEmployee(@PathVariable Long id) {

if (employeeService.getEmployeeById(id).isPresent()) {

employeeService.deleteEmployee(id);

return ResponseEntity.ok().build();

} else {

return ResponseEntity.notFound().build();

}

}

}

package com.example.employeemanagementsystem.controller;

import com.example.employeemanagementsystem.Department;

import com.example.employeemanagementsystem.service.DepartmentService;

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

import org.springframework.http.ResponseEntity;

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

import java.util.List;

import java.util.Optional;

@RestController

@RequestMapping("/api/departments")

public class DepartmentController {

@Autowired

private DepartmentService departmentService;

// Create a new Department

@PostMapping

public Department createDepartment(@RequestBody Department department) {

return departmentService.saveDepartment(department);

}

// Get all Departments

@GetMapping

public List<Department> getAllDepartments() {

return departmentService.getAllDepartments();

}

// Get a Department by ID

@GetMapping("/{id}")

public ResponseEntity<Department> getDepartmentById(@PathVariable Long id) {

Optional<Department> department = departmentService.getDepartmentById(id);

return department.map(ResponseEntity::ok).orElseGet(() -> ResponseEntity.notFound().build());

}

// Update a Department

@PutMapping("/{id}")

public ResponseEntity<Department> updateDepartment(@PathVariable Long id, @RequestBody Department departmentDetails) {

Optional<Department> department = departmentService.getDepartmentById(id);

if (department.isPresent()) {

departmentDetails.setId(id);

Department updatedDepartment = departmentService.saveDepartment(departmentDetails);

return ResponseEntity.ok(updatedDepartment);

} else {

return ResponseEntity.notFound().build();

}

}

// Delete a Department

@DeleteMapping("/{id}")

public ResponseEntity<Void> deleteDepartment(@PathVariable Long id) {

if (departmentService.getDepartmentById(id).isPresent()) {

departmentService.deleteDepartment(id);

return ResponseEntity.ok().build();

} else {

return ResponseEntity.notFound().build();

}

}

}