NAME: AWEYS XUSEEN JIMCAALE

ID: C1221259

CLASS:CA227

package samawade.samawadeSchool;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SamawadeSchoolApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(SamawadeSchoolApplication.class, args);  
 }  
  
}

package samawade.samawadeSchool.countroller;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
import java.net.URI;  
  
@RestController  
public class GreetingController {  
 @GetMapping("/")  
 public String home() {  
 return "Welcome to Aayatiin School API!";  
 }  
  
 @GetMapping("/greet/{name}")  
 public String greet(@PathVariable String name) {  
 return "Hello, " + name + "aweys xusen jimcaale";  
 }  
  
 @GetMapping("/success")  
 public ResponseEntity<String> success() {  
 return ResponseEntity.*ok*()  
 .header("X-Custom-Header", "Aayatiin")  
 .body("Success with custom header!");  
 }  
  
 @GetMapping("/not-found")  
 public ResponseEntity<String> notFound() {  
 return ResponseEntity.*status*(404).body("Error: Not Found");  
 }  
  
 @PostMapping("/create")  
 public ResponseEntity<String> create() {  
 return ResponseEntity.*created*(URI.*create*("/resource/1"))  
 .body("Created successfully");  
 }  
  
}

package samawade.samawadeSchool.model;  
  
public class Staff {  
 private Long id;  
 private String name;  
 private String role;  
  
 // Add these manually  
 public Long getId() {  
 return id;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getRole() {  
 return role;  
 }  
  
 public void setRole(String role) {  
 this.role = role;  
 }  
}

package samawade.samawadeSchool.countroller;  
  
import samawade.samawadeSchool.model.Staff;  
import samawade.samawadeSchool.service.StaffService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/staffs")  
public class StaffControlle {  
 @Autowired  
 private StaffService staffService;  
  
 @GetMapping  
 public List<Staff> getAll() {  
 return staffService.getAllStaff();  
 }  
  
 @GetMapping("/{id}")  
 public ResponseEntity<Staff> getById(@PathVariable Long id) {  
 Staff staff = staffService.getStaffById(id);  
 return (staff != null) ? ResponseEntity.*ok*(staff) : ResponseEntity.*notFound*().build();  
 }  
  
 @PostMapping  
 public ResponseEntity<Staff> create(@RequestBody Staff staff) {  
 return ResponseEntity.*status*(201).body(staffService.createStaff(staff));  
 }  
  
 @PutMapping("/{id}")  
 public ResponseEntity<Staff> update(@PathVariable Long id, @RequestBody Staff staff) {  
 Staff updated = staffService.updateStaff(id, staff);  
 return (updated != null) ? ResponseEntity.*ok*(updated) : ResponseEntity.*notFound*().build();  
 }  
  
 @DeleteMapping("/{id}")  
 public ResponseEntity<Void> delete(@PathVariable Long id) {  
 return (staffService.deleteStaff(id)) ? ResponseEntity.*noContent*().build() : ResponseEntity.*notFound*().build();  
 }  
}

============================================================================== package samawade.samawadeSchool.service;  
import samawade.samawadeSchool.model.Staff;  
import org.springframework.stereotype.Service;  
  
import java.util.\*;  
  
@Service  
public class StaffService {  
 private final Map<Long, Staff> staffMap = new HashMap<>();  
 private long idCounter = 1;  
  
 public List<Staff> getAllStaff() {  
 return new ArrayList<>(staffMap.values());  
 }  
  
 public Staff getStaffById(Long id) {  
 return staffMap.get(id);  
 }  
  
 public Staff createStaff(Staff staff) {  
 staff.setId(idCounter++);  
 staffMap.put(staff.getId(), staff);  
 return staff;  
 }  
  
 public Staff updateStaff(Long id, Staff updatedStaff) {  
 if (staffMap.containsKey(id)) {  
 updatedStaff.setId(id);  
 staffMap.put(id, updatedStaff);  
 return updatedStaff;  
 }  
 return null;  
 }  
  
 public boolean deleteStaff(Long id) {  
 return staffMap.remove(id) != null;  
 }  
}

