**Implement services for managing Country**

**CountryApplication.java**

package com.example.country;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountrytApplication {

public static void main(String[] args) {

SpringApplication.run(CountryManagementApplication.class, args);

}

}

**Country.java**

package com.example.country;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name = "country")

public class Country {

@Id

private String coCode;

private String coName;

public Country() { }

public Country(String coCode, String coName) {

this.coCode = coCode;

this.coName = coName;

}

public String getCoCode() { return coCode; }

public void setCoCode(String coCode) { this.coCode = coCode; }

public String getCoName() { return coName; }

public void setCoName(String coName) { this.coName = coName; }

}

**CountryRepository.java**

package com.example.country;

import java.util.List;

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

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByCoNameContainingIgnoreCase(String coName);

}

**CountryService.java**

package com.example.country;

import java.util.List;

public interface CountryService {

Country get(String code);

Country add(Country country);

Country update(String code, Country country);

void delete(String code);

List<Country> searchByName(String partial);

}

**CountryServiceImpl.java**

package com.example.country;

import java.util.List;

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

import org.springframework.stereotype.Service;

import jakarta.persistence.EntityNotFoundException;

@Service

public class CountryServiceImpl implements CountryService {

private final CountryRepository repo;

@Autowired

public CountryServiceImpl(CountryRepository repo) {

this.repo = repo;

}

@Override

public Country get(String code) {

return repo.findById(code).orElseThrow(EntityNotFoundException::new);

}

@Override

public Country add(Country country) {

return repo.save(country);

}

@Override

public Country update(String code, Country country) {

if (!repo.existsById(code)) throw new EntityNotFoundException();

country.setCoCode(code);

return repo.save(country);

}

@Override

public void delete(String code) {

repo.deleteById(code);

}

@Override

public List<Country> searchByName(String partial) {

return repo.findByCoNameContainingIgnoreCase(partial);

}

}

**CountryController.java**

package com.example.country;

import java.util.List;

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

import org.springframework.http.HttpStatus;

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

@RestController

@RequestMapping("/countries")

public class CountryController {

private final CountryService service;

@Autowired

public CountryController(CountryService service) {

this.service = service;

}

@GetMapping("/{code}")

public Country get(@PathVariable String code) {

return service.get(code);

}

@PostMapping

@ResponseStatus(HttpStatus.CREATED)

public Country add(@RequestBody Country country) {

return service.add(country);

}

@PutMapping("/{code}")

public Country update(@PathVariable String code, @RequestBody Country country) {

return service.update(code, country);

}

@DeleteMapping("/{code}")

@ResponseStatus(HttpStatus.NO\_CONTENT)

public void delete(@PathVariable String code) {

service.delete(code);

}

@GetMapping("/search")

public List<Country> search(@RequestParam String name) {

return service.searchByName(name);

}

}

**application.properties**

spring.datasource.url=jdbc:mysql://localhost:3306/countrydb

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

**pom.xml**

<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>

<groupId>com.example</groupId>

<artifactId>country</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>country</name>

<description>Country Management Service</description>

<packaging>jar</packaging>

<properties>

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

<spring-boot.version>3.3.0</spring-boot.version>

</properties>

<parent>

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

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

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

<relativePath/>

</parent>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**Output:**







