**Hands on 6**

**Find a country based on country code**

**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

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>countryapp</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

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

<spring.boot.version>3.2.5</spring.boot.version>

</properties>

<dependencies>

<dependency>

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

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

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

</dependency>

<dependency>

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

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

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

<version>8.3.0</version>

</dependency>

<dependency>

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

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

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

</plugin>

</plugins>

</build>

</project>

**CountryController.java**

package com.example.countryapp.controller;

import com.example.countryapp.model.Country;

import com.example.countryapp.exception.CountryNotFoundException;

import com.example.countryapp.service.CountryService;

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

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

import java.util.List;

@RestController

@RequestMapping("/countries")

public class CountryController {

@Autowired

private CountryService countryService;

@GetMapping("/{code}")

public Country getCountryByCode(@PathVariable String code) throws CountryNotFoundException {

return countryService.findCountryByCode(code);

}

@PostMapping

public Country addCountry(@RequestBody Country country) {

return countryService.addCountry(country);

}

@PutMapping("/{code}")

public Country updateCountry(@PathVariable String code, @RequestBody Country country) throws CountryNotFoundException {

return countryService.updateCountry(code, country);

}

@DeleteMapping("/{code}")

public String deleteCountry(@PathVariable String code) throws CountryNotFoundException {

return countryService.deleteCountry(code);

}

@GetMapping("/search")

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

return countryService.searchCountriesByName(name);

}

}

**CountryNotFoundException.java**

package com.example.countryapp.exception;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

**Country.java**

package com.example.countryapp.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name = "country")

public class Country {

@Id

private String code;

private String name;

public Country() {

}

public Country(String code, String name) {

this.code = code;

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Country{" +

"code='" + code + '\'' +

", name='" + name + '\'' +

'}';

}

}

**CountryRepository.java**

package com.example.countryapp.repository;

import com.example.countryapp.model.Country;

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

import java.util.List;

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByNameContainingIgnoreCase(String name);

}

**CountryService.java**

package com.example.countryapp.service;

import com.example.countryapp.model.Country;

import com.example.countryapp.exception.CountryNotFoundException;

import java.util.List;

public interface CountryService {

Country findCountryByCode(String countryCode) throws CountryNotFoundException;

Country addCountry(Country country);

Country updateCountry(String code, Country country) throws CountryNotFoundException;

String deleteCountry(String code) throws CountryNotFoundException;

List<Country> searchCountriesByName(String name);

}

**CountryServiceImpl.java**

package com.example.countryapp.service;

import com.example.countryapp.model.Country;

import com.example.countryapp.exception.CountryNotFoundException;

import com.example.countryapp.repository.CountryRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class CountryServiceImpl implements CountryService {

@Autowired

private CountryRepository countryRepository;

@Override

public Country findCountryByCode(String countryCode) throws CountryNotFoundException {

Optional<Country> result = countryRepository.findById(countryCode);

if (!result.isPresent()) {

throw new CountryNotFoundException("Country with code " + countryCode + " not found.");

}

return result.get();

}

@Override

public Country addCountry(Country country) {

return countryRepository.save(country);

}

@Override

public Country updateCountry(String code, Country country) throws CountryNotFoundException {

if (!countryRepository.existsById(code)) {

throw new CountryNotFoundException("Cannot update. Country with code " + code + " not found.");

}

country.setCode(code);

return countryRepository.save(country);

}

@Override

public String deleteCountry(String code) throws CountryNotFoundException {

if (!countryRepository.existsById(code)) {

throw new CountryNotFoundException("Cannot delete. Country with code " + code + " not found.");

}

countryRepository.deleteById(code);

return "Deleted country with code " + code;

}

@Override

public List<Country> searchCountriesByName(String name) {

return countryRepository.findByNameContainingIgnoreCase(name);

}

}

**CountryAppApplication.java**

package com.example.countryapp;

import com.example.countryapp.model.Country;

import com.example.countryapp.exception.CountryNotFoundException;

import com.example.countryapp.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryAppApplication implements CommandLineRunner {

private static final Logger LOGGER = LoggerFactory.getLogger(CountryAppApplication.class);

@Autowired

private CountryService countryService;

public static void main(String[] args) {

SpringApplication.run(CountryAppApplication.class, args);

}

@Override

public void run(String... args) {

try {

getCountryTest();

} catch (CountryNotFoundException e) {

System.out.println("Exception: " + e.getMessage());

}

}

private void getCountryTest() throws CountryNotFoundException {

System.out.println("Start");

Country country = countryService.findCountryByCode("IN");

System.out.println("Country Code: " + country.getCode());

System.out.println("Country Name: " + country.getName());

System.out.println("End");

}

}

