**Exercise 7 - Hands on 7 - Add a new country**

**CODE:**

**Country Class: -**

package com.cognizant.country\_management.model;  
import jakarta.persistence.Column;  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
import jakarta.persistence.Table;  
  
@Entity  
@Table(name = "country")  
public class Country {  
 @Id  
 @Column(name = "co\_code")  
 private String code;  
 @Column(name = "co\_name")  
 private String 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 + '\'' +  
 '}';  
 }  
}

**CountryNotFoundException Class: -**

package com.cognizant.country\_management.service.exception;  
  
public class CountryNotFoundException extends Exception {  
 public CountryNotFoundException(String message) {  
 super(message);  
 }  
}

**CountryRepository Interface: -**

package com.cognizant.country\_management.repository;  
import com.cognizant.country\_management.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 Class: -**

package com.cognizant.country\_management.service;  
import com.cognizant.country\_management.model.Country;  
import com.cognizant.country\_management.repository.CountryRepository;  
import com.cognizant.country\_management.service.exception.CountryNotFoundException;  
import jakarta.transaction.Transactional;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import java.util.List;  
import java.util.Optional;  
  
@Service  
public class CountryService {  
  
 @Autowired  
 private CountryRepository countryRepository;  
  
  
 @Transactional  
 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();  
 }  
  
 @Transactional  
 public void addCountry(Country country) {  
 countryRepository.save(country);  
 }  
  
 @Transactional  
 public void updateCountry(String code, String newName) {  
 Optional<Country> optional = countryRepository.findById(code);  
 if (optional.isPresent()) {  
 Country country = optional.get();  
 country.setName(newName);  
 countryRepository.save(country);  
 }  
 }  
  
 @Transactional  
 public void deleteCountry(String code) {  
 countryRepository.deleteById(code);  
 }  
  
 public List<Country> findCountriesByPartialName(String partialName) {  
 return countryRepository.findByNameContainingIgnoreCase(partialName);  
 }  
}

**application.properties file: -**

spring.application.name=country-management  
spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn  
spring.datasource.username=root  
spring.datasource.password=Sreddy@11  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true  
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect

**CountryManagementApplication Class(Main Class): -**

package com.cognizant.country\_management;  
  
import com.cognizant.country\_management.model.Country;  
import com.cognizant.country\_management.service.CountryService;  
import com.cognizant.country\_management.service.exception.CountryNotFoundException;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ConfigurableApplicationContext;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
import java.util.List;  
  
@SpringBootApplication  
public class CountryManagementApplication {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryManagementApplication.class);  
 private static CountryService *service*;  
  
 public static void main(String[] args) {  
 ConfigurableApplicationContext context = SpringApplication.*run*(CountryManagementApplication.class, args);  
 *service* = context.getBean(CountryService.class);  
  
 *getCountryTest*();  
 *addCountryTest*();  
 *updateCountryTest*();  
 *searchCountriesTest*();  
 *deleteCountryTest*();  
 *testAddCountry*();  
 }  
  
 private static void getCountryTest() {  
 *LOGGER*.info("START getCountryTest");  
 try {  
 Country country = *service*.findCountryByCode("IN");  
 *LOGGER*.debug("Country: {}", country);  
 } catch (CountryNotFoundException e) {  
 *LOGGER*.error("Exception: {}", e.getMessage());  
 }  
 *LOGGER*.info("END getCountryTest");  
 }  
  
 private static void addCountryTest() {  
 *LOGGER*.info("START addCountryTest");  
 Country newCountry = new Country();  
 newCountry.setCode("XY");  
 newCountry.setName("ExampleLand");  
 *service*.addCountry(newCountry);  
 *LOGGER*.debug("Country added: {}", newCountry);  
 *LOGGER*.info("END addCountryTest");  
 }  
  
 private static void updateCountryTest() {  
 *LOGGER*.info("START updateCountryTest");  
 *service*.updateCountry("XY", "UpdatedLand");  
 *LOGGER*.debug("Country updated");  
 *LOGGER*.info("END updateCountryTest");  
 }  
  
 private static void searchCountriesTest() {  
 *LOGGER*.info("START searchCountriesTest");  
 List<Country> found = *service*.findCountriesByPartialName("land");  
 found.forEach(country -> *LOGGER*.debug("Matched: {}", country));  
 *LOGGER*.info("END searchCountriesTest");  
 }  
  
 private static void deleteCountryTest() {  
 *LOGGER*.info("START deleteCountryTest");  
 *service*.deleteCountry("XY");  
 *LOGGER*.debug("Country XY deleted");  
 *LOGGER*.info("END deleteCountryTest");  
 }  
  
  
 private static void testAddCountry() {  
 *LOGGER*.info("START testAddCountry");  
  
 Country newCountry = new Country();  
 newCountry.setCode("ZZ");  
 newCountry.setName("Zootopia");  
  
 *service*.addCountry(newCountry);  
 *LOGGER*.debug("Country added: {}", newCountry);  
  
 try {  
 Country fetched = *service*.findCountryByCode("ZZ");  
 *LOGGER*.debug("Fetched Country: {}", fetched);  
 } catch (CountryNotFoundException e) {  
 *LOGGER*.error("Exception while fetching added country: {}", e.getMessage());  
 }  
  
 *LOGGER*.info("END testAddCountry");  
 }  
}

**OUTPUT:**

