**model/Country.java**

package model;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "country")

public class Country {

@Id

private String code;

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 + "]";

}

}

**repository/CountryRepository.java**

package repository;

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

import model.Country;

public interface CountryRepository extends JpaRepository<Country, String> {}

**service/exception/CountryNotFoundException.java**

package service.exception;

public class CountryNotFoundException extends Exception {

private static final long serialVersionUID = 1L;

public CountryNotFoundException(String message) {

super(message);

}

}

**service/CountryService.java**

package service;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import model.Country;

import repository.CountryRepository;

import service.exception.CountryNotFoundException;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

@Transactional

public Country findCountryByCode(String code) throws CountryNotFoundException {

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

if (!result.isPresent()) {

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

}

return result.get();

}

@Transactional

public void addCountry(Country country) {

countryRepository.save(country);

}

@Transactional

public void updateCountry(String code, String name) throws CountryNotFoundException {

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

if (!result.isPresent()) {

throw new CountryNotFoundException("Country not found to update.");

}

Country country = result.get();

country.setName(name);

countryRepository.save(country);

}

@Transactional

public void deleteCountry(String code) {

countryRepository.deleteById(code);

}

}

**EXERCISE 5 Implement services for managing Country**

**CountryApp.java**

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import model.Country;

import service.CountryService;

@SpringBootApplication

public class CountryApp {

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(CountryApp.class, args);

CountryService countryService = context.getBean(CountryService.class);

try {

**// 6. Find country by code**

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

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

**// 7. Add country**

Country newCountry = new Country();

newCountry.setCode("JP");

newCountry.setName("Japan");

countryService.addCountry(newCountry);

System.out.println("Added country: " + countryService.findCountryByCode("JP"));

**// 8. Update country**

countryService.updateCountry("JP", "Nippon");

System.out.println("Updated country: " + countryService.findCountryByCode("JP"));

**// 9. Delete country**

countryService.deleteCountry("JP");

System.out.println("Deleted country with code: JP");

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

}