**Exercise - 5: Hands on 5 -Implement services for managing 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 + '\'' +  
 '}';  
 }  
}

**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 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;  
  
 public Country findCountryByCode(String code) {  
 return countryRepository.findById(code).orElse(null);  
 }  
  
 @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 org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ConfigurableApplicationContext;  
import java.util.List;  
  
@SpringBootApplication  
public class CountryManagementApplication {  
  
 public static void main(String[] args) {  
 ConfigurableApplicationContext context = SpringApplication.*run*(CountryManagementApplication.class, args);  
 CountryService service = context.getBean(CountryService.class);  
  
 System.*out*.println(service.findCountryByCode("IN"));  
  
 Country newCountry = new Country();  
 newCountry.setCode("XY");  
 newCountry.setName("ExampleLand");  
 service.addCountry(newCountry);  
  
 service.updateCountry("XY", "UpdatedLand");  
  
 List<Country> found = service.findCountriesByPartialName("land");  
 found.forEach(System.*out*::println);  
  
 service.deleteCountry("XY");  
 }  
  
}

**OUTPUT:**

