**Hands On 8: Update a country based on code**   
**Country.java**

package com.example.country.entity;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
  
@Entity  
public class Country {  
 @Id  
 private String code;  
 private String name;  
  
 *// Constructors* 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.country.repository;  
  
import com.example.country.entity.Country;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface CountryRepository extends JpaRepository<Country, String> {  
}

**CountryService.java**

package com.example.country.service;  
  
import com.example.country.entity.Country;  
import com.example.country.repository.CountryRepository;  
import com.example.country.service.exception.CountryNotFoundException;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import org.springframework.transaction.annotation.Transactional;  
  
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 not found with code: " + countryCode);  
 }  
 return result.get();  
 }  
 @Transactional  
 public void addCountry(Country country) {  
 countryRepository.save(country);  
 }

@Transactional  
public void updateCountry(String code, String newName) throws CountryNotFoundException {  
 Country country = countryRepository.findById(code)  
 .orElseThrow(() -> new CountryNotFoundException("Country not found for code " + code));  
 country.setName(newName);  
 countryRepository.save(country);  
}

}

**CountryApplication.java**

package com.example.country;  
  
import com.example.country.entity.Country;  
import com.example.country.repository.CountryRepository;  
import com.example.country.service.exception.CountryNotFoundException;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.CommandLineRunner;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import com.example.country.service.CountryService;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
@SpringBootApplication  
public class CountryApplication implements CommandLineRunner {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryApplication.class);  
  
 @Autowired  
 private CountryRepository countryRepository;  
 @Autowired  
 private CountryService countryService;  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(CountryApplication.class, args);  
 }  
  
 @Override  
 public void run(String... args) {  
 testUpdateCountry();  
 }  
  
 private void testUpdateCountry() {  
 *LOGGER*.info("Start testUpdateCountry");  
  
 try {  
 countryService.updateCountry("IN", "Bharat");  
 Country updated = countryService.findCountryByCode("IN");  
 *LOGGER*.debug("Updated Country: {}", updated);  
 } catch (CountryNotFoundException e) {  
 *LOGGER*.error("Country not found: {}", e.getMessage());  
 }  
  
 *LOGGER*.info("End testUpdateCountry");  
 }  
  
}

**application.properties**

spring.h2.console.enabled=true  
spring.h2.console.path=/h2-console  
  
spring.datasource.url=jdbc:h2:mem:testdb  
spring.datasource.driverClassName=org.h2.Driver  
spring.datasource.username=sa  
spring.datasource.password=  
  
spring.jpa.show-sql=true  
spring.jpa.hibernate.ddl-auto=create  
spring.datasource.initialization-mode=always  
  
logging.level.com.example.country=DEBUG

**Output**



