# Implement services for managing Country:

## Hibernate Configuration:

spring.datasource.url=jdbc:mysql://localhost:3306/your\_db\_name spring.datasource.username=your\_username spring.datasource.password=your\_password spring.jpa.hibernate.ddl-auto=validate

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLD ialect

## Create Entity: Country:

package com.cognizant.springlearn.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 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 + "]";

}

}

### Create Repository: CountryRepository:

package com.cognizant.springlearn.repository; import com.cognizant.springlearn.model.Country;

import org.springframework.data.jpa.repository.JpaRepository; import java.util.List;

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByNameContainingIgnoreCase(String namePart);

}

### Custom Exception: CountryNotFoundException:

package com.cognizant.springlearn.service.exception;

public class CountryNotFoundException extends Exception { public CountryNotFoundException(String message) {

super(message);

}

}

## Service Class: CountryService:

package com.cognizant.springlearn.service; import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.repository.CountryRepository;

import com.cognizant.springlearn.service.exception.CountryNotFoundExcept ion;

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

import org.springframework.transaction.annotation.Transactional; import java.util.List;

import java.util.Optional; @Service

public class CountryService { @Autowired

private CountryRepository countryRepository; @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 Country addCountry(Country country) { return countryRepository.save(country);

}

@Transactional

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

Country existing = findCountryByCode(code); existing.setName(updatedCountry.getName()); return countryRepository.save(existing);

}

@Transactional

public void deleteCountry(String code) throws CountryNotFoundException {

Country country = findCountryByCode(code); countryRepository.delete(country);

}

@Transactional(readOnly = true)

public List<Country> findCountriesByName(String namePart) { return

countryRepository.findByNameContainingIgnoreCase(namePart);

}

}

## Populate Country Table:

DELETE FROM country;

## Test from OrmLearnApplication java Copy Edit:

package com.cognizant.springlearn;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.service.CountryService;

import com.cognizant.springlearn.service.exception.CountryNotFoun dException;

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.SpringBootApplicati on;

import java.util.List; @SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

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

@Autowired

private CountryService countryService; public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) throws Exception { testFindByCode();

testAddCountry(); testUpdateCountry(); testDeleteCountry(); testSearchByName();

}

private void testFindByCode() { LOGGER.info("Find by Code: IN"); try {

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

LOGGER.debug("Result: {}", c);

} catch (CountryNotFoundException e) { LOGGER.error("Not Found: {}", e.getMessage());

} }

private void testAddCountry() { LOGGER.info("Add Country"); Country country = new Country(); country.setCode("ZZ"); country.setName("Zootopia"); countryService.addCountry(country);

}

private void testUpdateCountry() { LOGGER.info("Update Country"); try {

Country update = new Country(); update.setName("New Zootopia"); countryService.updateCountry("ZZ", update);

} catch (CountryNotFoundException e) { LOGGER.error("Update Failed: {}", e.getMessage());

}

}

private void testDeleteCountry() { LOGGER.info("Delete Country"); try {

countryService.deleteCountry("ZZ");

} catch (CountryNotFoundException e) { LOGGER.error("Delete Failed: {}", e.getMessage());

}

}

private void testSearchByName() { LOGGER.info("Search by Name: 'land'"); List<Country> list =

countryService.findCountriesByName("land"); for (Country c : list) {

LOGGER.debug("Found: {}", c);

}

}

}

## OUTPUT:

INFO c.c.s.OrmLearnApplication : Starting OrmLearnApplication on localhost...

INFO c.c.s.OrmLearnApplication : Started OrmLearnApplication in

X.XXX seconds

INFO c.c.s.OrmLearnApplication : Find by Code: IN

DEBUG c.c.s.OrmLearnApplication : Result: Country [code=IN, name=India]

INFO c.c.s.OrmLearnApplication : Add Country

-- (Zootopia inserted into DB)

INFO c.c.s.OrmLearnApplication : Update Country

-- (Zootopia updated to New Zootopia)

INFO c.c.s.OrmLearnApplication : Delete Country

-- (New Zootopia deleted from DB)

INFO c.c.s.OrmLearnApplication : Search by Name: 'land'

DEBUG c.c.s.OrmLearnApplication : Found: Country [code=FI, name=Finland]

DEBUG c.c.s.OrmLearnApplication : Found: Country [code=IS, name=Iceland]

DEBUG c.c.s.OrmLearnApplication : Found: Country [code=NZ, name=New Zealand]

DEBUG c.c.s.OrmLearnApplication : Found: Country [code=CH, name=Switzerland]

DEBUG c.c.s.OrmLearnApplication : Found: Country [code=TH, name=Thailand]

DEBUG c.c.s.OrmLearnApplication : Found: Country [code=PL, name=Poland]

DEBUG c.c.s.OrmLearnApplication : Found: Country [code=IE, name=Ireland]

DEBUG c.c.s.OrmLearnApplication : Found: Country [code=NL, name=Netherlands]

Process finished with exit code 0

# Find a country based on country code:

## Create CountryNotFoundException class:

package com.cognizant.springlearn.service.exception; public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) { super(message);

}

}

## Add findCountryByCode() method in CountryService:

package com.cognizant.springlearn.service; import com.cognizant.springlearn.model.Country; import

com.cognizant.springlearn.service.exception.CountryNotFoundExcept ion;

import com.cognizant.springlearn.repository.CountryRepository; 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();

}

}

## Add test method in OrmLearnApplication.java:

package com.cognizant.springlearn;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.service.CountryService; import

com.cognizant.springlearn.service.exception.CountryNotFoundExcept

ion;

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 OrmLearnApplication implements CommandLineRunner

{

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

@Autowired

private CountryService countryService; public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) throws Exception { testGetCountryByCode(); // invoking the test

}

private void testGetCountryByCode() { LOGGER.info("Start");

try {

Country country = countryService.findCountryByCode("IN"); LOGGER.debug("Country: {}", country);

} catch (CountryNotFoundException e) { LOGGER.error("Exception: {}", e.getMessage());

}

LOGGER.info("End");

}

}

## Country Entity and Repository:

package com.cognizant.springlearn.model; import jakarta.persistence.Entity;

import jakarta.persistence.Id; @Entity

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 + "]";

}

}

## CountryRepository.java:

package com.cognizant.springlearn.repository;

import org.springframework.data.jpa.repository.JpaRepository; import com.cognizant.springlearn.model.Country;

public interface CountryRepository extends JpaRepository<Country, String> {

}

## OUTPUT:

INFO Start

DEBUG Country: Country [code=IN, name=India] INFO End

OR

INFO Start

ERROR Exception: Country not found with code: XX INFO End

# Add a new country:

## Modify CountryService to add addCountry() method:

@Transactional

public void addCountry(Country country) { countryRepository.save(country);

}

## Add testAddCountry() in OrmLearnApplication.java:

private void testAddCountry() { LOGGER.info("Start");

Country newCountry = new Country(); newCountry.setCode("XY"); newCountry.setName("Xylotopia"); countryService.addCountry(newCountry); try {

Country result = countryService.findCountryByCode("XY"); LOGGER.debug("Added Country: {}", result);

} catch (CountryNotFoundException e) {

LOGGER.error("Country not found after add: {}", e.getMessage());

}

LOGGER.info("End");

}

## Call testAddCountry() from run() method:

@Override

public void run(String... args) throws Exception { testAddCountry();

}

## Verify in Database:

SELECT \* FROM country WHERE co\_code = 'XY';

## Sample Console Output:

INFO Start

DEBUG Added Country: Country [code=XY, name=Xylotopia] INFO End