**Hands on 6**

**Find a country based on country code** 

* Create new exception class CountryNotFoundException in com.cognizant.spring-learn.service.exception
* Create new method findCountryByCode() in CountryService with @Transactional annotation
* In findCountryByCode() method, perform the following steps:
  + Method signature

@Transactional

public Country findCountryByCode(String countryCode) throws CountryNotFoundException

* Get the country based on findById() built in method

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

* From the result, check if a country is found. If not found, throw CountryNotFoundException

if (!result.isPresent())

* Use get() method to return the country fetched.

Country country = result.get();

* Include new test method in OrmLearnApplication to find a country based on country code and compare the country name to check if it is valid.

    private static void getAllCountriesTest() {

        LOGGER.info("Start");

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

  LOGGER.debug("Country:{}", country);

        LOGGER.info("End");

    }

* Invoke the above method in main() method and test it.

**NOTE:** SME to explain the importance of @Transactional annotation. Spring takes care of creating the Hibernate session and manages the transactionality when executing the service method.

**ANSWER**

**CountryNotFoundException.java**

package com.cognizant.springlearn.service.exception;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

**CountryService.java**

package com.example.country;

import java.util.List;

import com.cognizant.springlearn.service.exception.CountryNotFoundException;

public interface CountryService {

Country findCountryByCode(String code) throws CountryNotFoundException;

Country addCountry(Country country);

Country updateCountry(String code, Country updated);

void deleteCountry(String code);

List<Country> searchCountryByPartialName(String partialName);

}

**CountryServiceImpl.java**

package com.example.country;

import java.util.List;

import java.util.Optional;

import com.cognizant.springlearn.service.exception.CountryNotFoundException;

import jakarta.transaction.Transactional;

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

import org.springframework.stereotype.Service;

@Service

public class CountryServiceImpl implements CountryService {

@Autowired

private CountryRepository countryRepository;

@Override

@Transactional

public Country findCountryByCode(String code) throws CountryNotFoundException {

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

if (!result.isPresent()) {

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

}

return result.get();

}

@Override

public Country addCountry(Country country) {

return countryRepository.save(country);

}

@Override

public Country updateCountry(String code, Country updated) {

Country existing = countryRepository.findById(code)

.orElseThrow(() -> new RuntimeException("Country not found with code: " + code));

existing.setName(updated.getName());

return countryRepository.save(existing);

}

@Override

public void deleteCountry(String code) {

countryRepository.deleteById(code);

}

@Override

public List<Country> searchCountryByPartialName(String partialName) {

return countryRepository.findByNameContainingIgnoreCase(partialName);

}

}

**CountryApplication.java**

package com.example.country;

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

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.cognizant.springlearn.service.exception.CountryNotFoundException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SpringBootApplication

public class CountryApplication {

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

@Autowired

private CountryService countryService;

public static void main(String[] args) {

var context = SpringApplication.run(CountryApplication.class, args);

CountryApplication app = context.getBean(CountryApplication.class);

app.getCountryTest();

}

private void getCountryTest() {

LOGGER.info("Start");

try {

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

LOGGER.debug("Country: {}", country.getName());

} catch (CountryNotFoundException e) {

LOGGER.error("Country not found", e);

}

LOGGER.info("End");

}

}

**OUTPUT:**

INFO com.example.country.CountryApplication - Start

DEBUG com.example.country.CountryApplication - Country: India

INFO com.example.country.CountryApplication - End