Hands-on 6: Find a Country Based on Country Code

This task demonstrates how to retrieve a country using its country code, handle exceptions when the country is not found, and test the functionality using Spring Boot.

# 1. Create Exception Class

Create a new exception class named CountryNotFoundException in the package:  
`com.cognizant.spring-learn.service.exception`  
  
Code:  
package com.cognizant.spring\_learn.service.exception;  
  
public class CountryNotFoundException extends Exception {  
 public CountryNotFoundException(String message) {  
 super(message);  
 }  
}

# 2. Add Method in CountryService

Add the method `findCountryByCode()` in `CountryService` class.

@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();  
}

# 3. Add Test Method in OrmLearnApplication

private static void getAllCountriesTest() {  
 LOGGER.info("Start");  
 Country country = countryService.findCountryByCode("IN");  
 LOGGER.debug("Country:{}", country);  
 LOGGER.info("End");  
}

# 4. Invoke in Main Method

public static void main(String[] args) {  
 ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);  
 countryService = context.getBean(CountryService.class);  
 getAllCountriesTest();  
}

# 5. Importance of @Transactional Annotation

The `@Transactional` annotation tells Spring to manage the transaction automatically. Spring creates and binds a Hibernate session to the current thread, allowing lazy loading and transactional consistency. It ensures that any exception causes a rollback and no partial data is committed to the database.