**REST - Get country exceptional scenario**   
   
In the previous hands on where we implemented getting country based on country code, what happens if the country code provided in the URL is not present.  
   
**Refer steps below to implement**

* Create a new exception class com.cognizant.springlearn.service.exception.CountryNotFoundException
* Include below specified annotation at the class level in CountryNotFoundException class

@ResponseStatus(value = HttpStatus.NOT\_FOUND, reason = "Country not found")

* In CountryService.getCountry() method include the logic to throw CountryNotFoundException if the country code does not exists in the list.
* In CountryController.getCountry() method include throws clause in method signature. This will respond to the caller of the web service with appropriate error message in JSON format.
* Test the service in postman and using curl command. Refer below for executing curl command.

**Steps to invoke RESTful Web Service using curl command**

* Open Git Bash
* Execute the below command

curl -i <http://localhost:8090/country/az>

**Sample Request**: <http://localhost:8083/country/az>   
**Sample Response**:

{

"timestamp": "2019-10-02T03:27:54.521+0000",

"status": 404,

"error": "Not Found",

"message": "Country not found",

"path": "/country/az"

}

**Solution**

**CountryNotFoundException.java**

package com.cognizant.springapplearn.service.exception;  
  
import org.springframework.http.HttpStatus;  
import org.springframework.web.bind.annotation.ResponseStatus;  
  
@ResponseStatus(value = HttpStatus.NOT\_FOUND, reason = "Country not found")  
public class CountryNotFoundException extends RuntimeException {  
 public CountryNotFoundException(String countryNotFound) {  
 super("Country not found");  
 }  
}

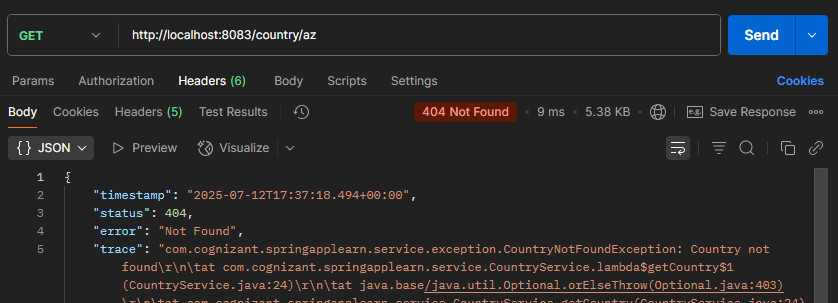
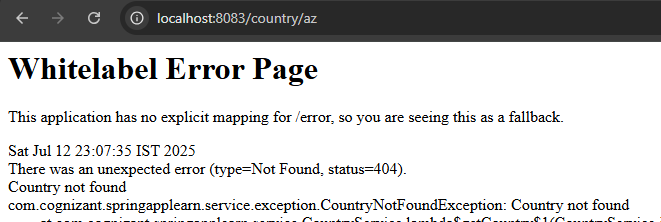
**CountryController.java**

package com.cognizant.springapplearn.controller;  
  
import com.cognizant.springapplearn.model.Country;  
import com.cognizant.springapplearn.service.CountryService;  
import com.cognizant.springapplearn.service.exception.CountryNotFoundException;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import java.util.List;  
  
@RestController  
public class CountryController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
 @Autowired  
 private CountryService countryService;  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 LOGGER.info("START - getCountryIndia()");  
 ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country");  
 LOGGER.info("END - getCountryIndia()");  
 return country;  
 }  
 @GetMapping("/countries")  
 public List<Country> getAllCountries() {  
 LOGGER.info("START - getAllCountries()");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countryList = context.getBean("countryList", List.class);  
  
 LOGGER.info("END - getAllCountries()");  
 return countryList;  
 }  
 **@GetMapping("/country/{code}")**  
 **public Country getCountry(@PathVariable String code) throws CountryNotFoundException {**  
 **return countryService.getCountry(code);**  
 **}**  
  
  
  
}

**CountryService.java**

package com.cognizant.springapplearn.service;  
  
import java.util.List;  
  
import com.cognizant.springapplearn.service.exception.CountryNotFoundException;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.stereotype.Service;  
  
import com.cognizant.springapplearn.model.Country;  
  
@Service  
public class CountryService {  
 @Autowired  
 private ApplicationContext context;  
 public Country getCountry(String code) {  
 List<Country> countryList = (List<Country>) context.getBean("countryList");  
 return countryList.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElseThrow(() -> new CountryNotFoundException("Country not found"));  
 }  
}

**Output**



**MockMVC - Test get country service**   
   
Using MockMVC test the get country service.  
   
Create a test cases to test the following aspects:

* Test is the CountryController is loaded
* Invoke the service to get country and check in the response if it contains code as "IN" and name as "India"

Refer steps below to implement

* **Test loading CountryController**
  + Include CountryController instance variable in SpringLearnApplicationTests.java and autowire the instance variable using annotation.

@Autowired

private CountryController countryController;

* Include assertion in contextLoads() method to check if controller is loaded.

@Test

public void contextLoads() {

assertNotNull(countryController);

}

* Run the JUnit testing by right clicking on SpringLearnApplicationTests.java > Run As > JUnit Test
* This test can also be executed in command line using the following maven command in the root folder of the project. (Note: don't forget to include proxy details in the below command)

mvn clean test

* Check if the log in the constructor of CountryController is called.

* **Test service to get the country**
  + Include below imports

import static org.junit.Assert.assertNotNull;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.jsonPath;

import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;

import org.springframework.test.web.servlet.MockMvc;

import org.springframework.test.web.servlet.ResultActions;

* Include @AutoConfigureMockMvc annotation for SpringLearnApplicationTests.java
* Autowire mock mvc in SpringLearnApplicationTests.java

@Autowired

private MockMvc mvc;

* Include a new test method in SpringLearnApplicationTests.java

@Test

public void testGetCountry() throws Exception {

}

* Include the following line in the new method that calls the service method. Execute the JUnit test and check if the test case is successful.

@Test

public void testGetCountry() throws Exception {

ResultActions actions = mvc.perform(get("/country"));

}

* Include the following line to check if the HTTP Status is 200, which means the call is successful. Execute JUnit test and check if the test case is successful.

@Test

public void testGetCountry() throws Exception {

ResultActions actions = mvc.perform(get("/country"));

actions.andExpect(status().isOk());

}

* Include the following line to check if the code is available in the reponse

@Test

public void getCountry() throws Exception {

ResultActions actions = mvc.perform(get("/country"));

actions.andExpect(status().isOk());

actions.andExpect(jsonPath("$.code").exists());

}

* Include the following line to check if the value of code is "IN"

@Test

public void getCountry() throws Exception {

ResultActions actions = mvc.perform(get("/country"));

actions.andExpect(status().isOk());

actions.andExpect(jsonPath("$.code").exists());

actions.andExpect(jsonPath("$.code").value("IN"));

}

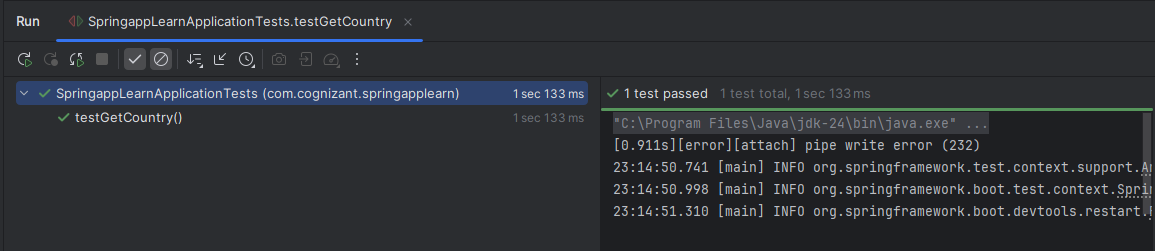
* Using above two steps include checks for "name" attribute and check if it's value is "India"

**Solution**

**SpringappLearnApplicationTests.java**

package com.cognizant.springapplearn;  
  
import com.cognizant.springapplearn.controller.CountryController;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.assertNotNull;  
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;  
import org.springframework.boot.test.context.SpringBootTest;  
import org.springframework.test.web.servlet.MockMvc;  
import org.springframework.test.web.servlet.ResultActions;  
  
@SpringBootTest  
@AutoConfigureMockMvc  
public class SpringappLearnApplicationTests {  
 @Autowired  
 private CountryController countryController;  
 @Autowired  
 private MockMvc mvc;  
 @Test  
 public void contextLoads() {  
 assertNotNull(countryController);  
 }  
 @Test  
 public void testGetCountry() throws Exception {  
 ResultActions actions = mvc.perform(get("/country"));  
 actions.andExpect(status().isOk());  
 actions.andExpect(jsonPath("$.code").exists());  
 actions.andExpect(jsonPath("$.code").value("IN"));  
 actions.andExpect(jsonPath("$.name").exists());  
 actions.andExpect(jsonPath("$.name").value("India"));  
 }  
}

**Output**



**MockMVC - Test get country service for exceptional scenario**   
   
Include MockMVC test that checks if correct response is received when there is an error.  
   
Refer steps below to implement

* Include a new test method testGetCountryException() in SpringLearnApplicationTests.java
* Validate the error response using status(). Refer code below.

actions.andExpect(status().isBadRequest());

actions.andExpect(status().reason("Country Not found"));

**Solution**

**CountryNotFoundException.java**

package com.cognizant.springapplearn.service.exception;  
  
import org.springframework.http.HttpStatus;  
import org.springframework.web.bind.annotation.ResponseStatus;  
@ResponseStatus(value = HttpStatus.NOT\_FOUND, reason = "Country not found")  
public class CountryNotFoundException extends RuntimeException {  
 public CountryNotFoundException(String countryNotFound) {  
 super("Country not found");  
 }  
}

**SpringappLearnApplication.java**

package com.cognizant.springapplearn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.ImportResource;  
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;  
@ImportResource("classpath:country.xml")  
@SpringBootApplication  
@EnableJpaRepositories("com.cognizant.springapplearn.repository")  
public class SpringappLearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(SpringappLearnApplication.class, args);  
 }  
}

**Output**

