**Week 4 : Spring REST Using Spring Boot**

**Hello World RESTful Web Service**

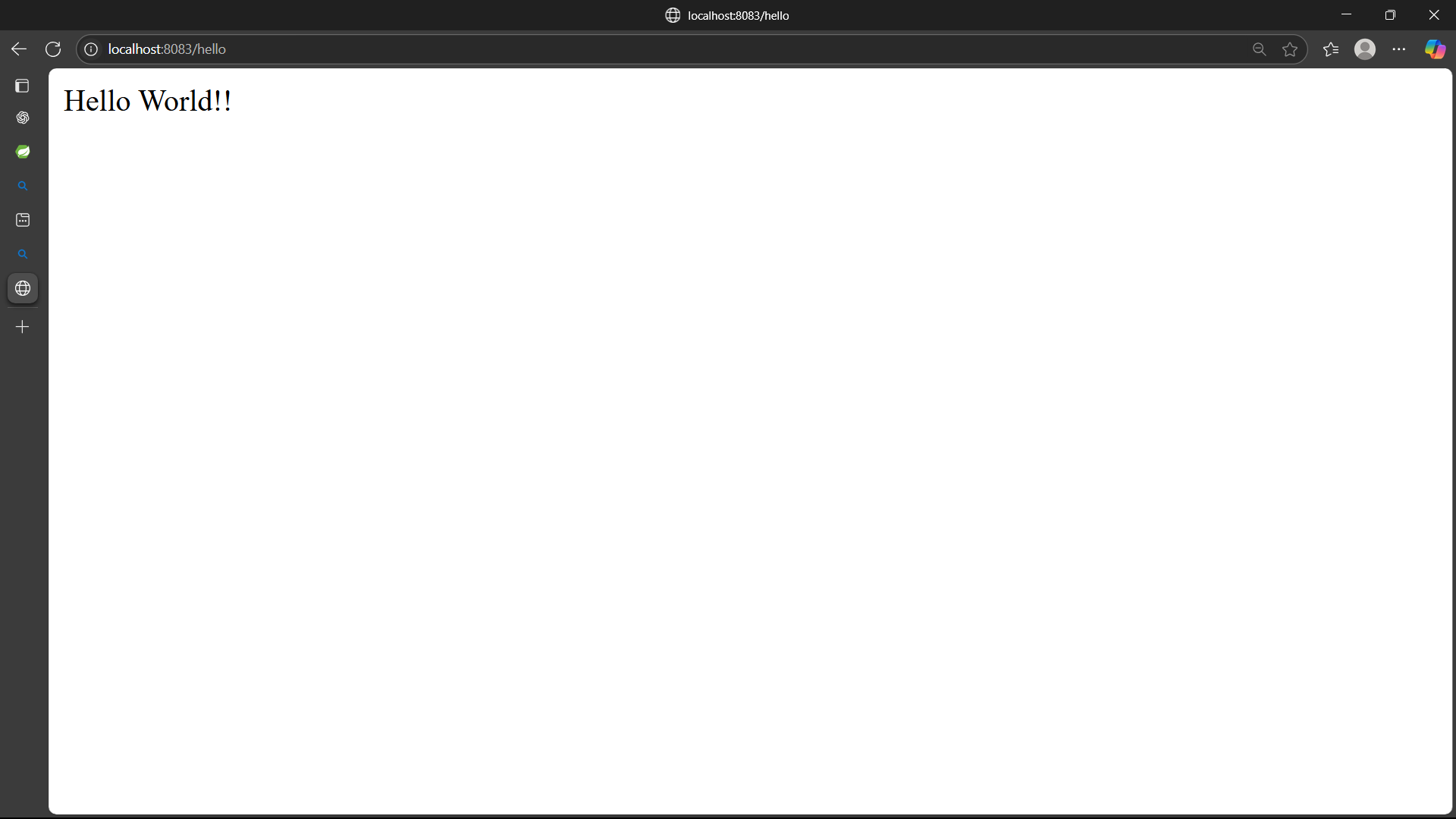
**Code :**

**SpringlearnApplication.java**

package com.cognizant.springlearn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringlearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringlearnApplication.class, args);  
 }  
}

**HelloController.java**

package com.cognizant.springlearn.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloController {  
  
 @GetMapping("/hello")  
 public String sayHello() {  
 return "Hello World!!";  
 }  
}

**Output :**  


**REST - Country Web Service**

**Code :**

**SpringlearnApplication.java**

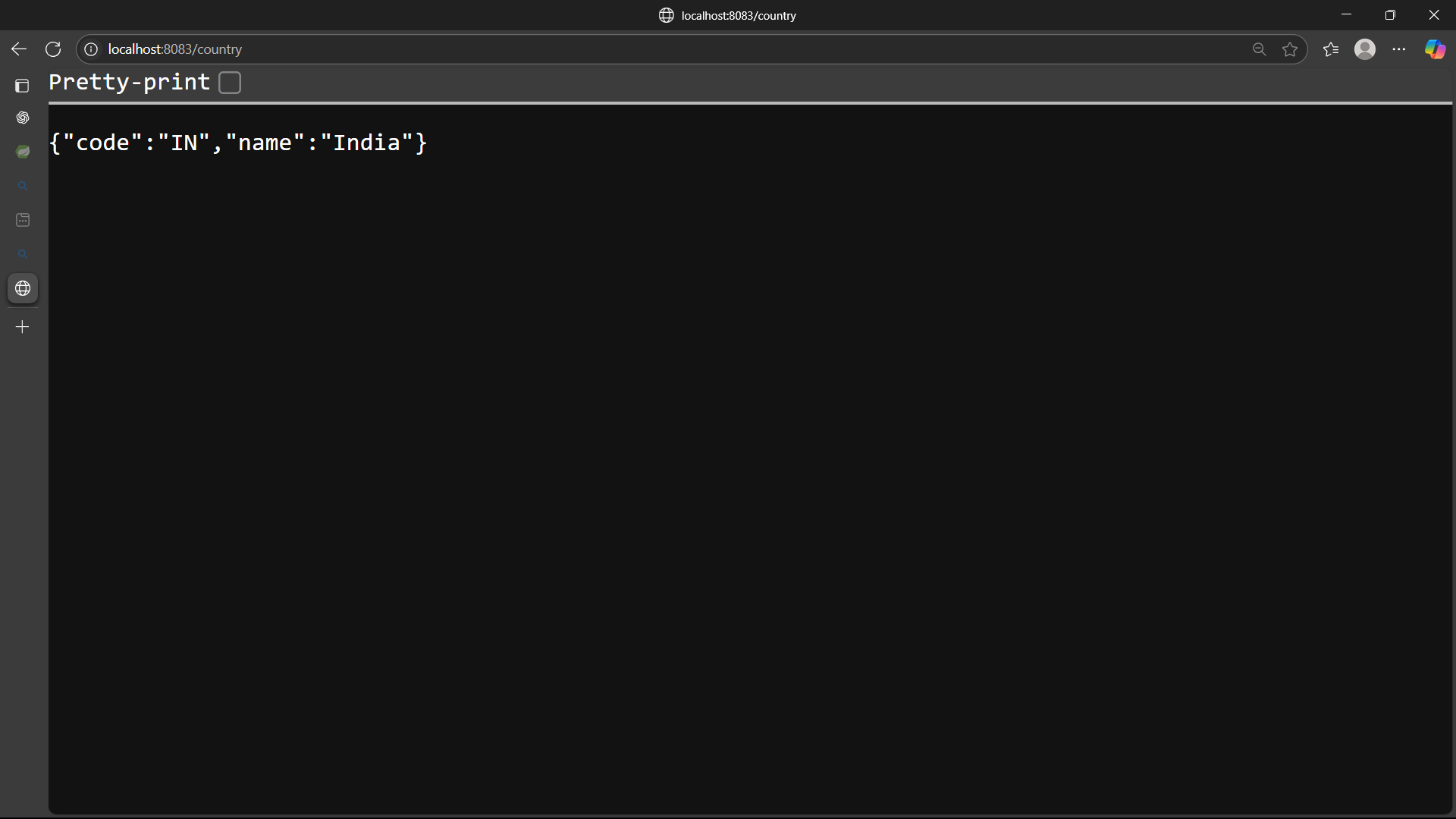
package com.cognizant.springlearn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringlearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringlearnApplication.class, args);  
 }  
}

**Country.java**

package com.cognizant.springlearn.model;  
public class Country {  
 private String code;  
 private String name;  
  
 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;  
 }  
}

**CountryController.java**

package com.cognizant.springlearn.controller;  
  
  
import com.cognizant.springlearn.model.Country;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class CountryController {  
  
 @GetMapping("/country")  
 public Country getCountryIndia() {  
 return new Country("IN", "India");  
 }  
}

**Output :**  


**REST - Get all countries**   
**Code :**

**SpringlearnApplication.java**

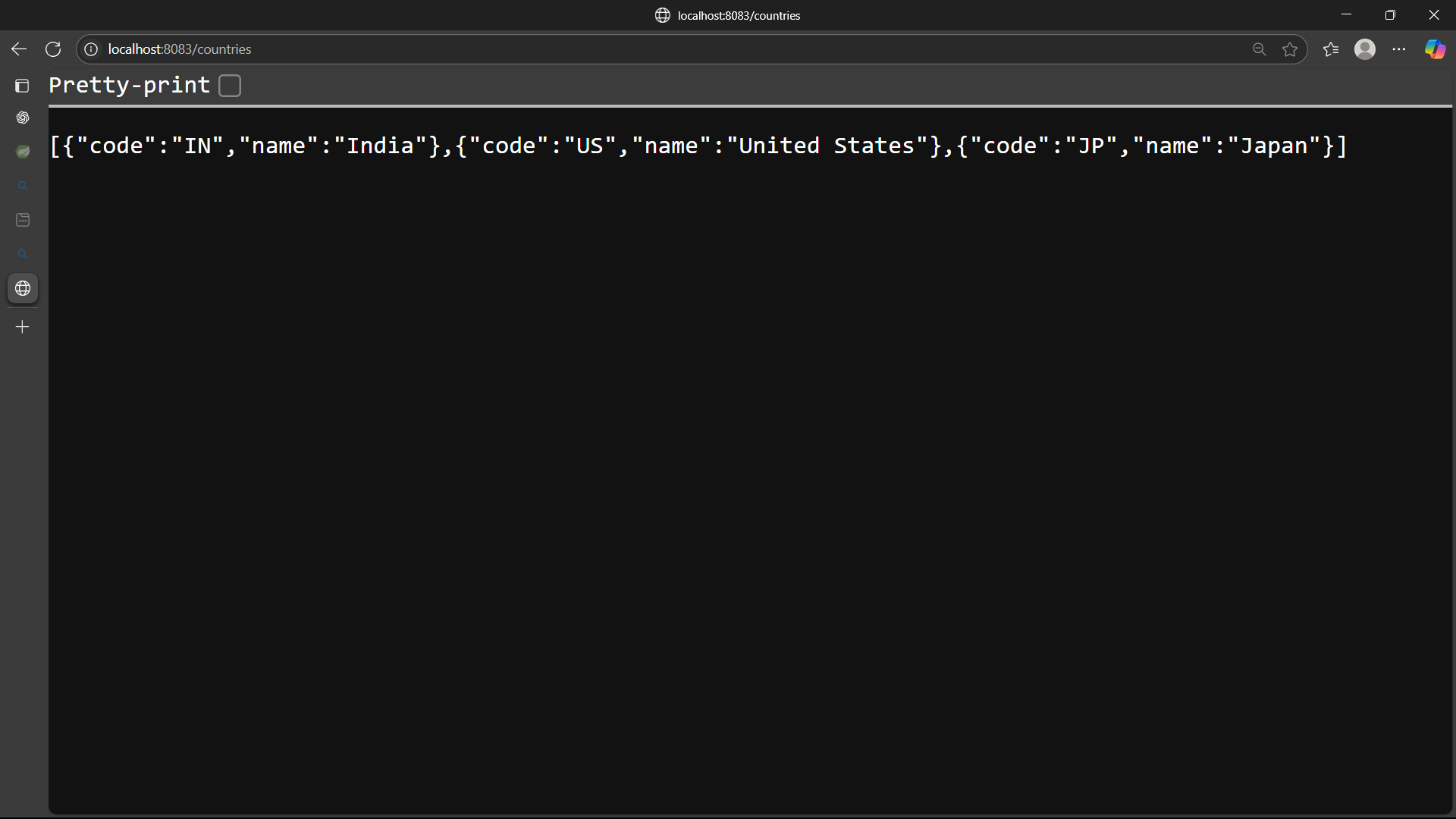
package com.cognizant.springlearn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringlearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringlearnApplication.class, args);  
 }  
}

**Country.java**

package com.cognizant.springlearn.model;  
public class Country {  
 private String code;  
 private String name;  
  
 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;  
 }  
}

**CountryController.java**

package com.cognizant.springlearn.controller;  
  
  
import com.cognizant.springlearn.model.Country;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.ArrayList;  
  
@RestController  
public class CountryController {  
  
 @GetMapping("/countries")  
 public ArrayList<Country> getAllCountries() {  
 ArrayList<Country> countryList = new ArrayList<>();  
 countryList.add(new Country("IN", "India"));  
 countryList.add(new Country("US", "United States"));  
 countryList.add(new Country("JP", "Japan"));  
 return countryList;  
 }  
}

**Output :**  


**REST - Get country based on country code**   
**Code :**

**SpringlearnApplication.java**

package com.cognizant.springlearn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringlearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringlearnApplication.class, args);  
 }  
}

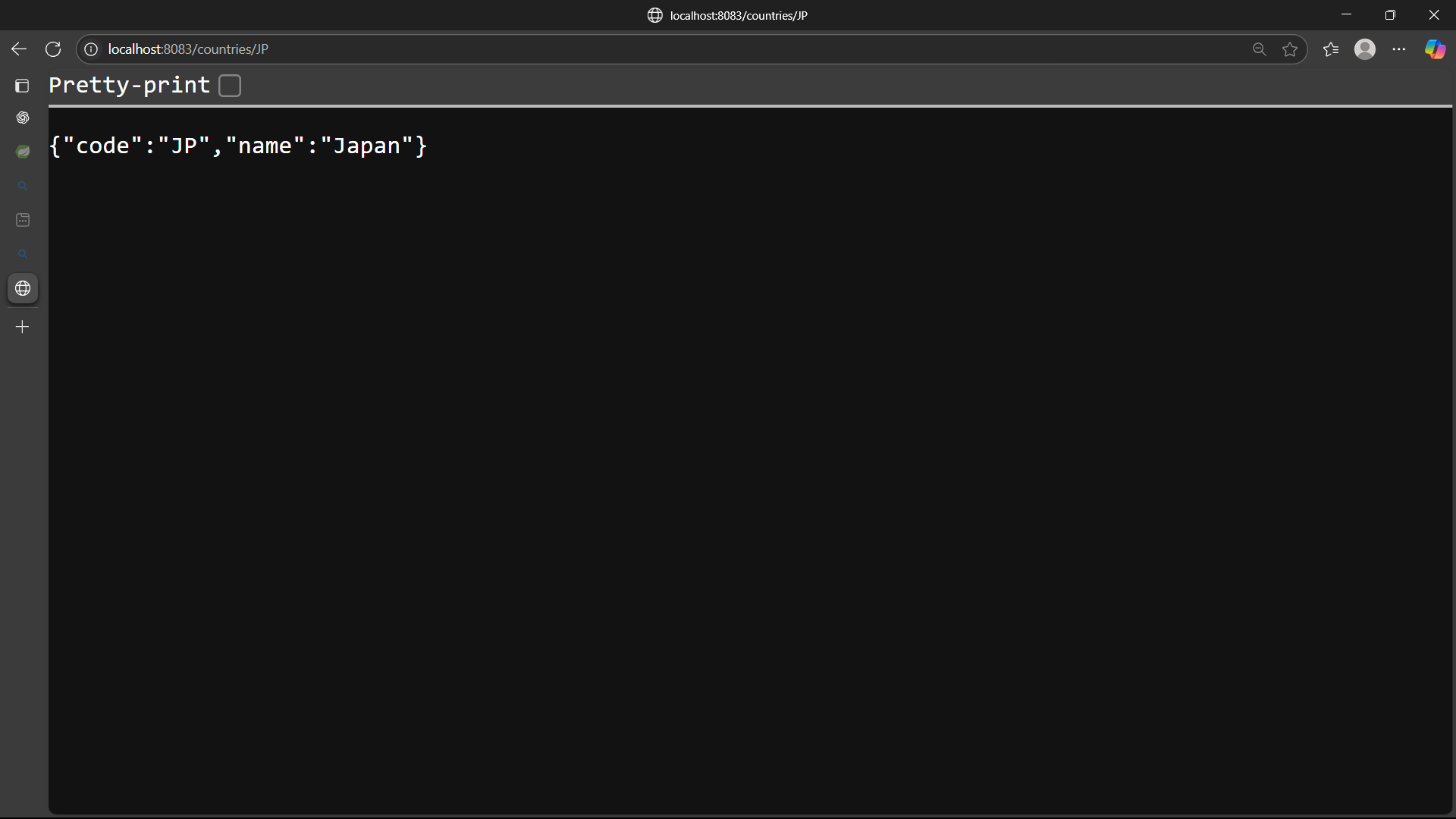
**Country.java**

package com.cognizant.springlearn.model;  
public class Country {  
 private String code;  
 private String name;  
  
 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;  
 }  
}

**CountryController.java**

package com.cognizant.springlearn.controller;  
  
  
import com.cognizant.springlearn.model.Country;  
import org.springframework.http.HttpStatus;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.server.ResponseStatusException;  
  
import java.util.ArrayList;  
@RestController  
public class CountryController {  
  
 @GetMapping("/countries/{code}")  
 public Country getCountryByCode(@PathVariable String code) {  
 ArrayList<Country> countryList = new ArrayList<>();  
 countryList.add(new Country("IN", "India"));  
 countryList.add(new Country("US", "United States"));  
 countryList.add(new Country("JP", "Japan"));  
  
 for (Country country : countryList) {  
 if (country.getCode().equalsIgnoreCase(code)) {  
 return country;  
 }  
 }  
 throw new ResponseStatusException(HttpStatus.*NOT\_FOUND*, "Country not found");  
 }  
}

**Output :**



**REST - Get country exceptional scenario**   
**Code :**

**SpringlearnApplication.java**

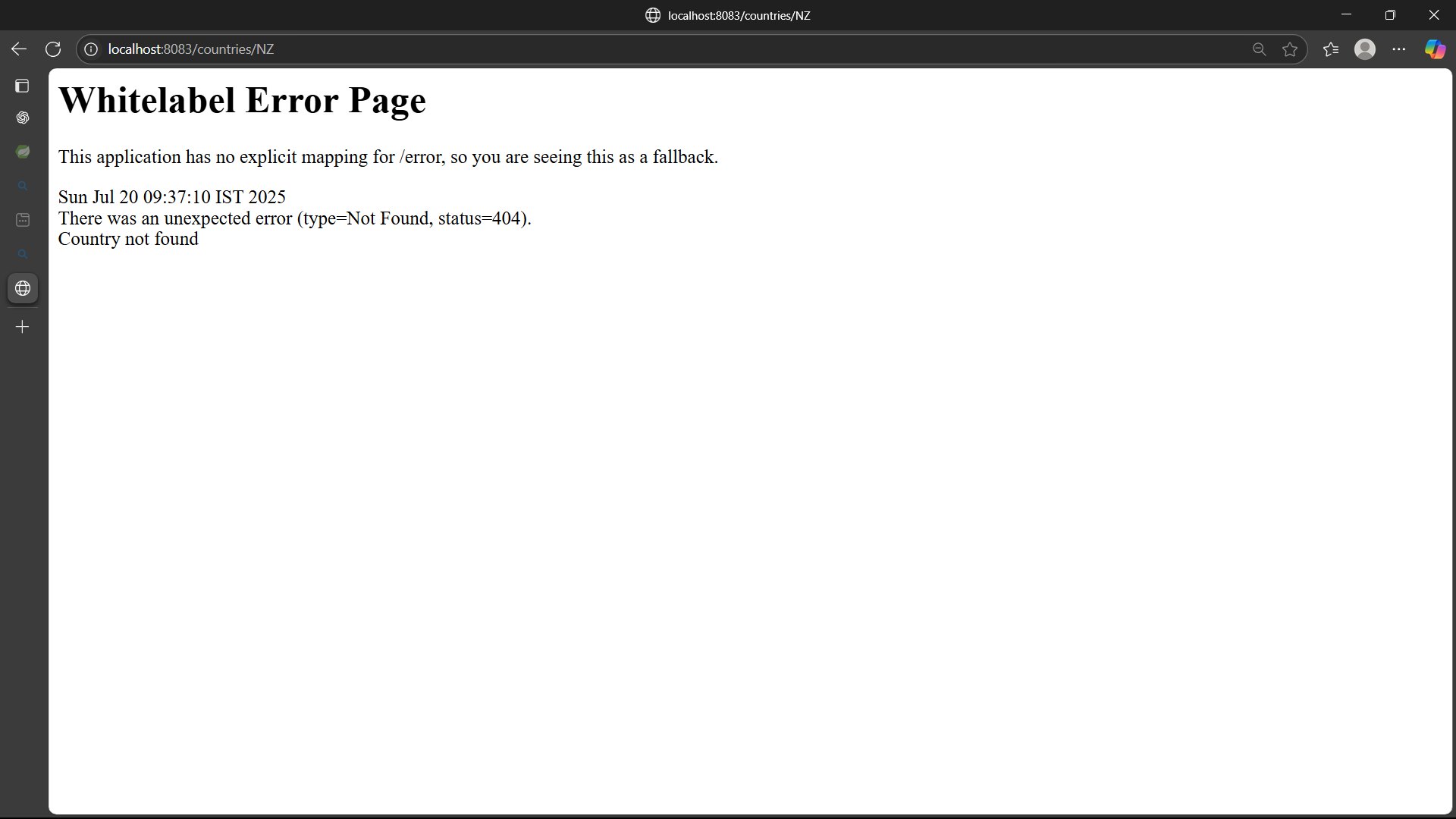
package com.cognizant.springlearn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringlearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringlearnApplication.class, args);  
 }  
}

**Country.java**

package com.cognizant.springlearn.model;  
public class Country {  
 private String code;  
 private String name;  
  
 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;  
 }  
}

**CountryController.java**

package com.cognizant.springlearn.controller;  
  
  
import com.cognizant.springlearn.model.Country;  
import org.springframework.http.HttpStatus;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.server.ResponseStatusException;  
  
import java.util.ArrayList;  
@RestController  
public class CountryController {  
  
 @GetMapping("/countries/{code}")  
 public Country getCountryByCode(@PathVariable String code) {  
 ArrayList<Country> countryList = new ArrayList<>();  
 countryList.add(new Country("IN", "India"));  
 countryList.add(new Country("US", "United States"));  
 countryList.add(new Country("JP", "Japan"));  
  
 for (Country country : countryList) {  
 if (country.getCode().equalsIgnoreCase(code)) {  
 return country;  
 }  
 }  
 throw new ResponseStatusException(HttpStatus.*NOT\_FOUND*, "Country not found");  
 }  
}

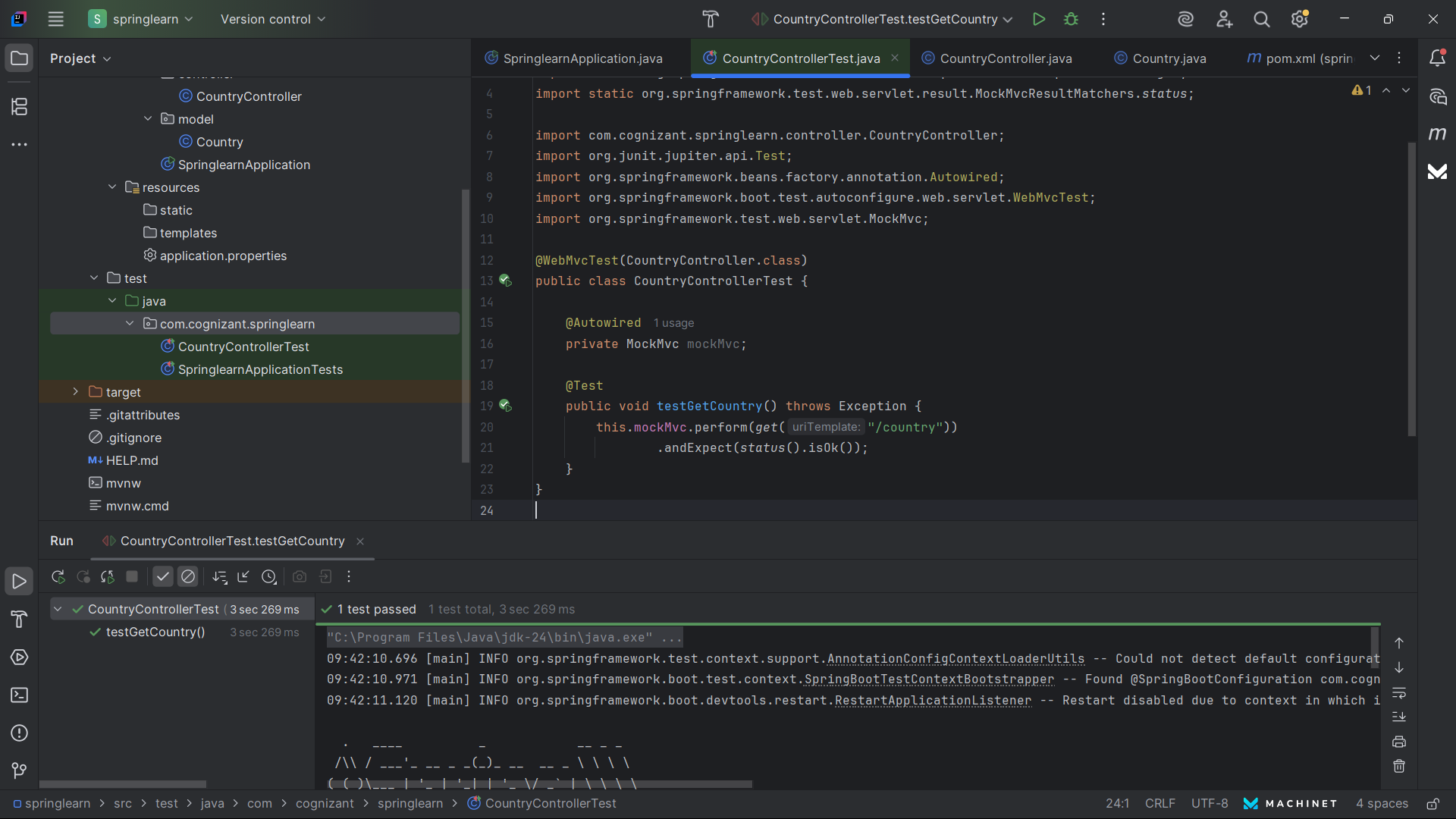
**Output :**  


**MockMVC - Test get country service**   
**Code :**

**CountryControllerTest.java**

package com.cognizant.springlearn;  
  
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*get*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*status*;  
  
import com.cognizant.springlearn.controller.CountryController;  
import org.junit.jupiter.api.Test;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;  
import org.springframework.test.web.servlet.MockMvc;  
  
@WebMvcTest(CountryController.class)  
public class CountryControllerTest {  
  
 @Autowired  
 private MockMvc mockMvc;  
  
 @Test  
 public void testGetCountry() throws Exception {  
 this.mockMvc.perform(*get*("/country"))  
 .andExpect(*status*().isOk());  
 }  
}

**Output :**



**MockMVC - Test get country service for exceptional scenario**   
  
**Code :**

**CountryControllerTest.java**

package com.cognizant.springlearn;  
  
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*get*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*status*;  
  
import com.cognizant.springlearn.exception.CountryNotFoundException;  
import com.cognizant.springlearn.controller.CountryController;  
import com.cognizant.springlearn.service.CountryService;  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
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.boot.test.mock.mockito.MockBean;  
import org.springframework.test.web.servlet.MockMvc;  
  
@SpringBootTest  
@AutoConfigureMockMvc  
public class CountryControllerTest {  
  
 @Autowired  
 private MockMvc mockMvc;  
  
 @MockBean  
 private CountryService countryService;  
  
 @Test  
 public void testGetCountry\_NotFound() throws Exception {  
 String countryCode = "NZ";  
  
 Mockito.*when*(countryService.getCountry(countryCode))  
 .thenThrow(new CountryNotFoundException("Country not found"));  
  
 mockMvc.perform(*get*("/countries/" + countryCode))  
 .andExpect(*status*().isNotFound());  
 }  
}

**Output :**

