**1.Hello World RESTful Web Service**   
  
Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework. Refer details below:  
  
**Method:** GET  
**URL:** /hello  
**Controller:** com.cognizant.spring-learn.controller.HelloController  
**Method Signature:** public String sayHello()  
**Method Implementation:** return hard coded string "Hello World!!"  
**Sample Request**: http://localhost:8083/hello  
**Sample Response:** Hello World!!

**STEP 1: Create HelloController Class**

**HelloController.java:**

**package com.cognizant.spring\_learn.controller;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**import org.springframework.web.bind.annotation.GetMapping;**

**import org.springframework.web.bind.annotation.RestController;**

**@RestController**

**public class HelloController {**

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

**@GetMapping("/hello")**

**public String sayHello() {**

**LOGGER.info("START - sayHello()");**

**String message = "Hello World!!";**

**LOGGER.info("END - sayHello()");**

**return message;**

**}**

**}**

**STEP 2: Set Port to 8083**

**Application.properties:**

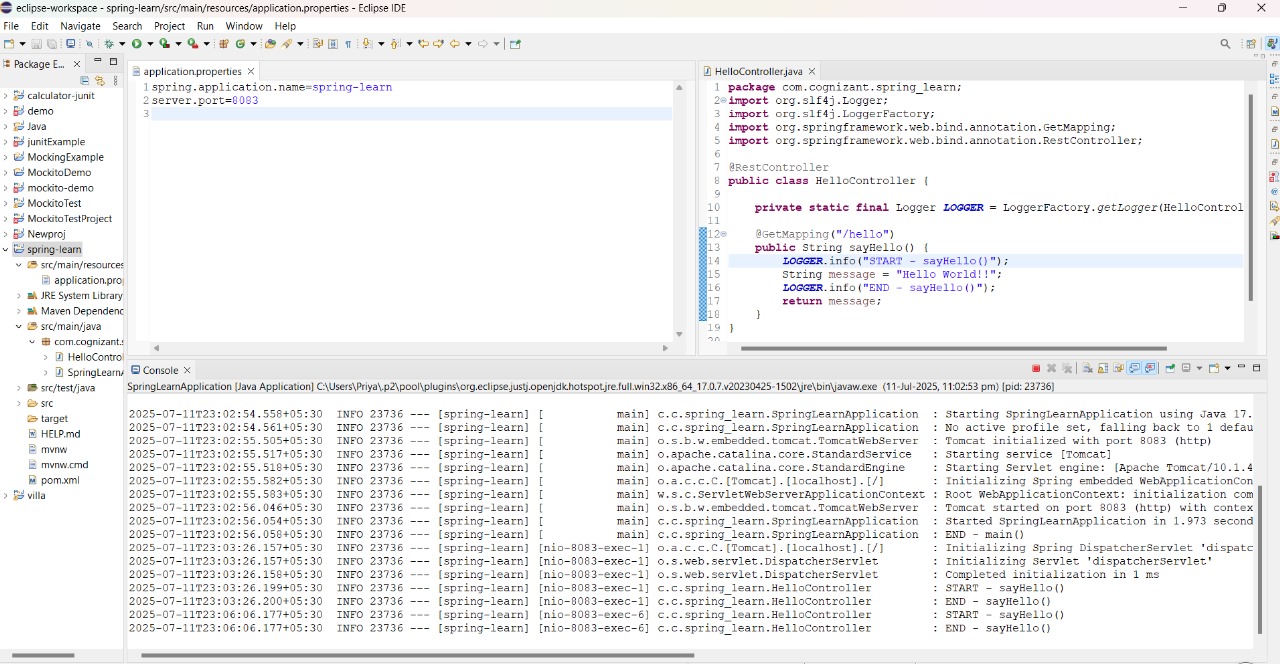
server.port=8083

**STEP 3: Run the Application**

**Right-click on your project in Eclipse or IntelliJ, then:**

**👉 Run as → Spring Boot App**

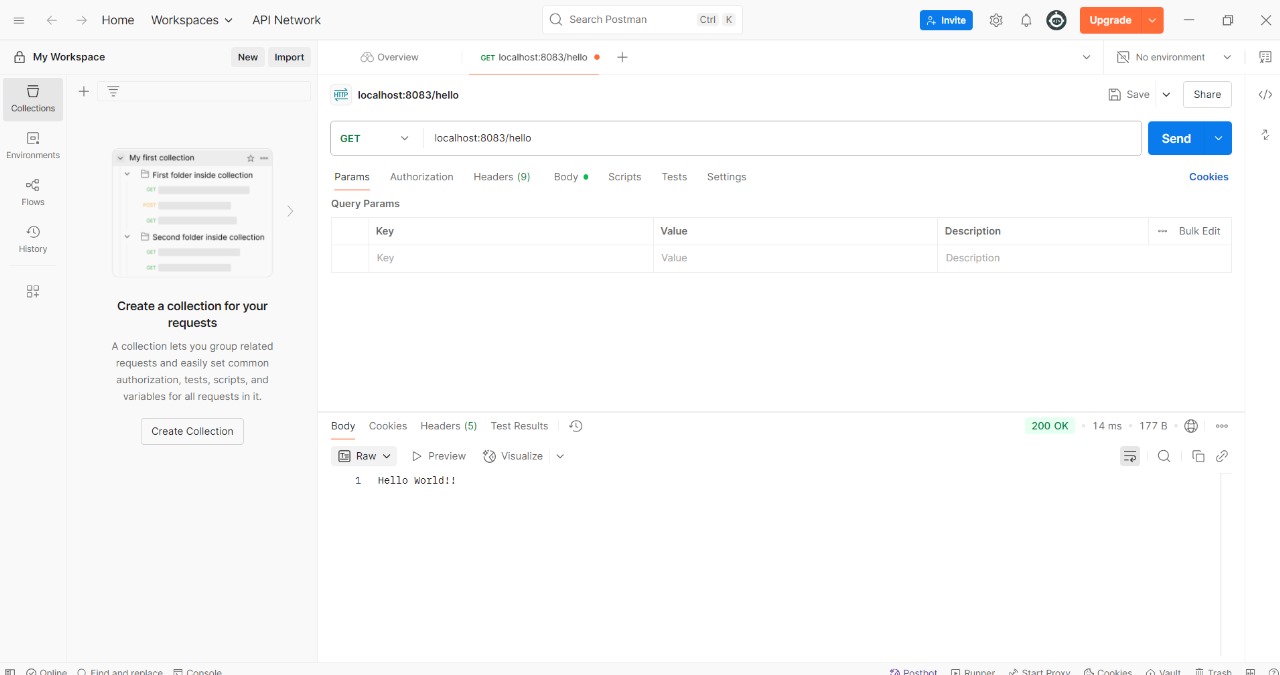
**OUTPUT:**



**Test the Endpoint In Chrome:**



**Test the Endpoint In Postman:**



**2. REST - Country Web Service**   
  
Write a REST service that returns India country details in the earlier created spring learn application.  
  
**URL**: /country  
**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation**: @RequestMapping  
**Method Name**: getCountryIndia()  
**Method Implementation**: Load India bean from spring xml configuration and return  
**Sample Request**: http://localhost:8083/country  
**Sample Response**:

{

  "code": "IN",

  "name": "India"

}

**Spring.xml**:

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="in" class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="IN" />

<property name="name" value="India" />

</bean>

</beans>

**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;

}

// Getters and setters

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.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class CountryController {

@RequestMapping("/country")

public Country getCountryIndia() {

ApplicationContext context = new ClassPathXmlApplicationContext("spring.xml");

Country country = (Country) context.getBean("in");

return country;

}

}

application.properties:

spring.application.name=spring-learn

server.port=8082

SpringLearnApplication.java:

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource;

@SpringBootApplication

@ImportResource("classpath:spring.xml") // This loads your XML configuration

public class SpringLearnApplication {

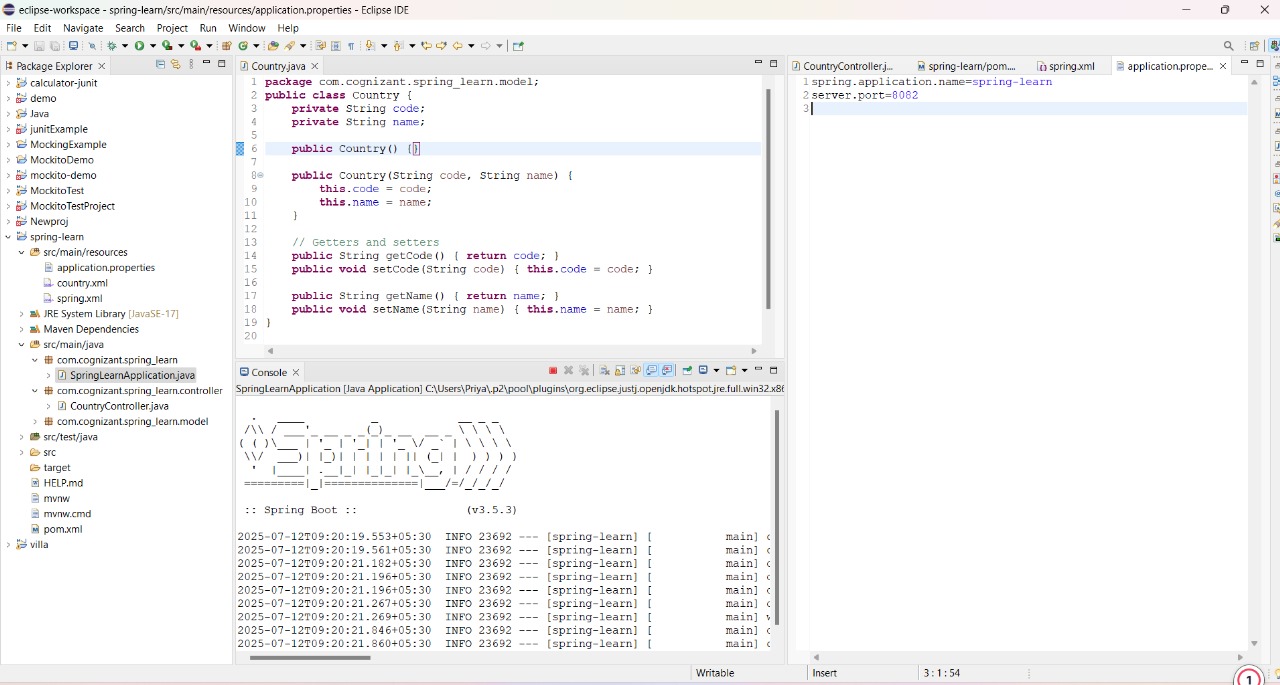
public static void main(String[] args) {

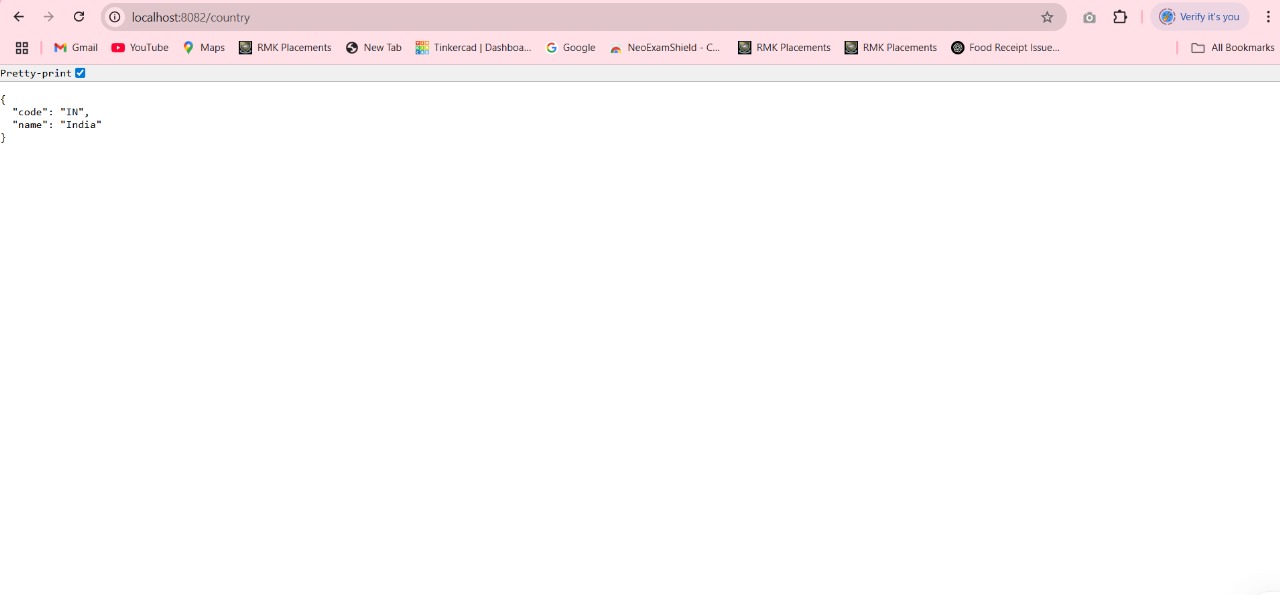
SpringApplication.run(SpringLearnApplication.class, args);

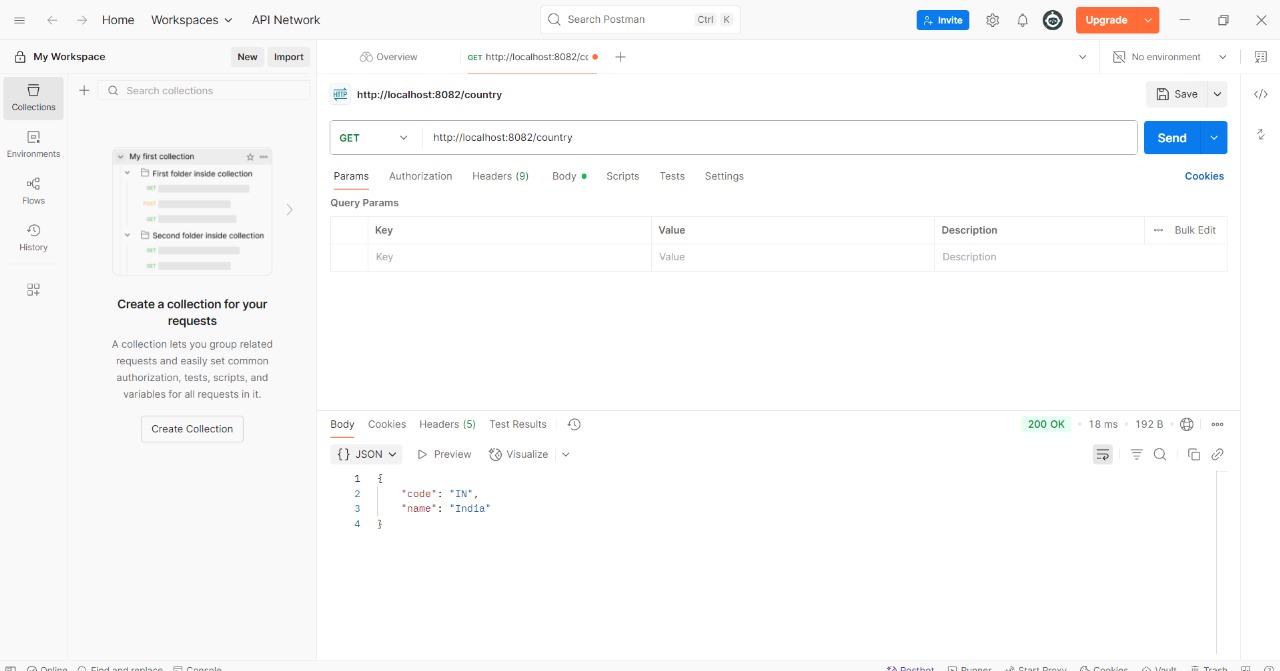
}

}

**OUTPUT:**







**3. REST - Get all countries**   
  
Write a REST service that returns all the countries.  
  
**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation**: @GetMapping("/countries")  
**Method Name**: getAllCountries()  
**Method Implementation**: Load country list from country.xml and return  
  
**Sample Request**: http://localhost:8083/countries  
**Sample Response**:

[

  { "code": "IN", "name": "India"},

  { "code": "US", "name": "United States"},

  { "code": "JP", "name": "Japan"},

  { "code": "DE", "name": "Germany"}

]

**country.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:util="http://www.springframework.org/schema/util"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/util

http://www.springframework.org/schema/util/spring-util.xsd">

<util:list id="countryList">

<bean class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="IN" />

<property name="name" value="India" />

</bean>

<bean class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="US" />

<property name="name" value="United States" />

</bean>

<bean class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="JP" />

<property name="name" value="Japan" />

</bean>

<bean class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="DE" />

<property name="name" value="Germany" />

</bean>

</util:list>

</beans>

**CountryController.java**:

**package** com.cognizant.spring\_learn.controller;

**import** com.cognizant.spring\_learn.model.Country;

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

**import** org.springframework.beans.factory.annotation.Qualifier;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.RestController;

**import** java.util.List;

@RestController

**public** **class** CountryController {

@Autowired

@Qualifier("countryList")

**private** List<Country> countries;

@GetMapping("/countries")

**public** List<Country> getAllCountries() {

**return** countries;

}

}

**AppConfig.java:**

**package** com.cognizant.spring\_learn.config;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.context.annotation.ImportResource;

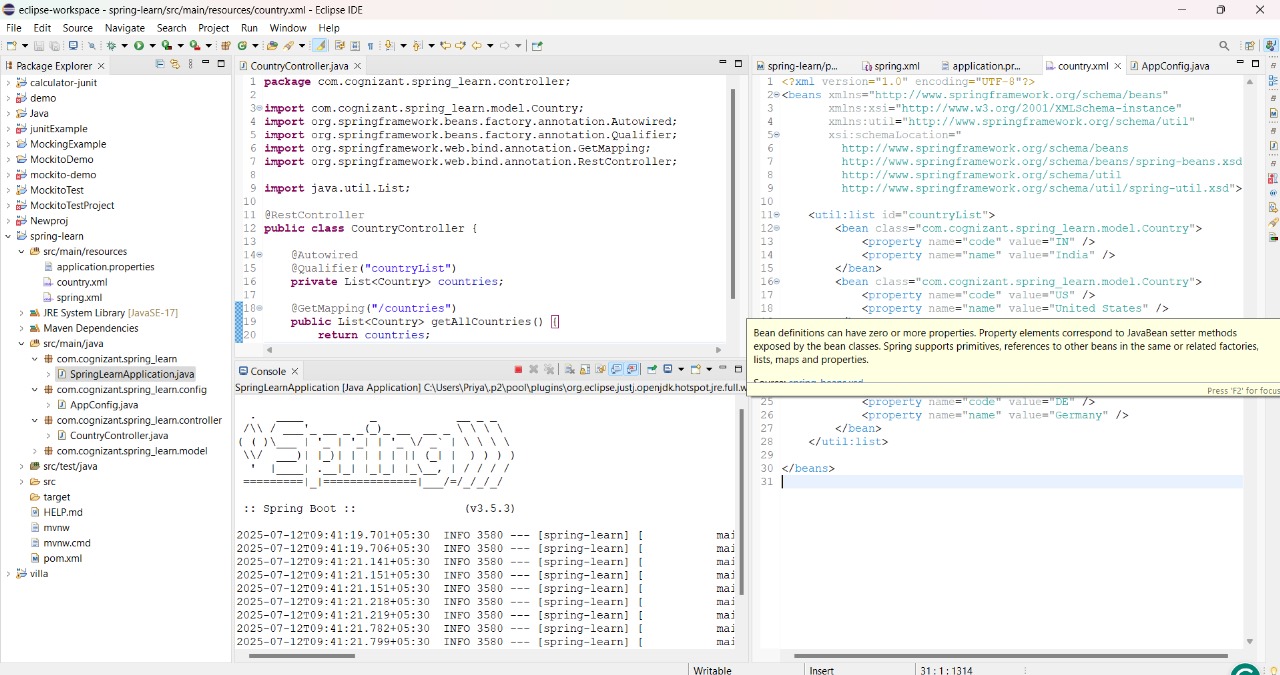
@Configuration

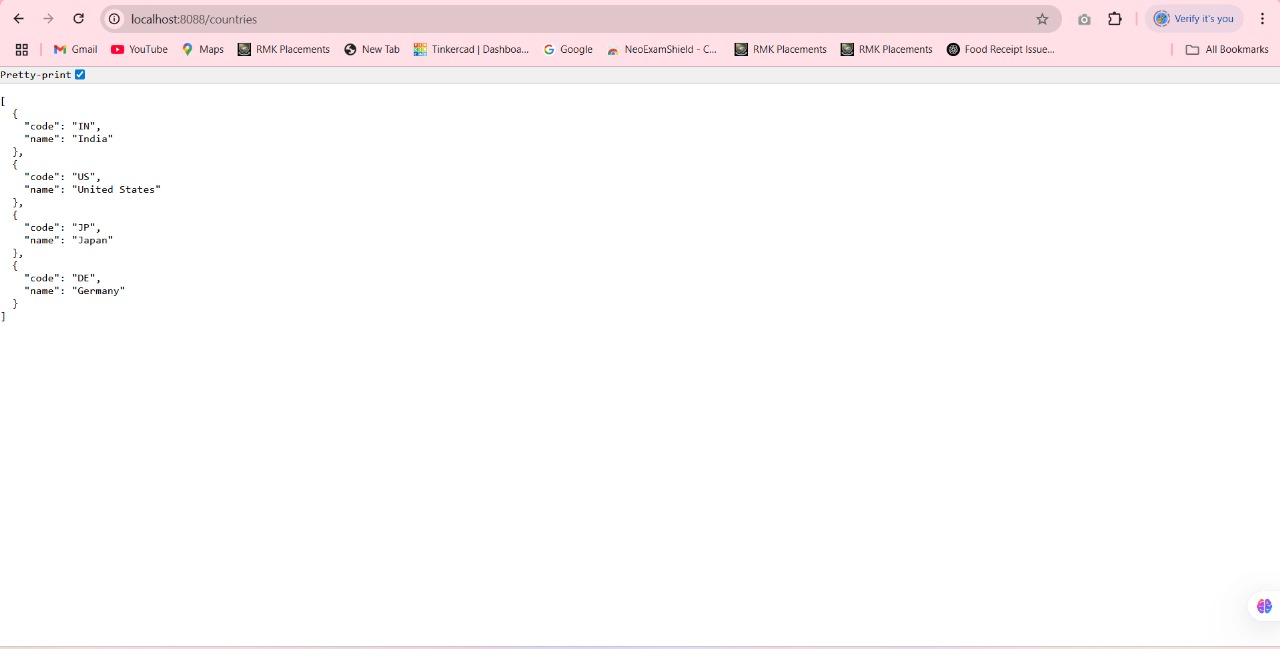
@ImportResource("classpath:country.xml")

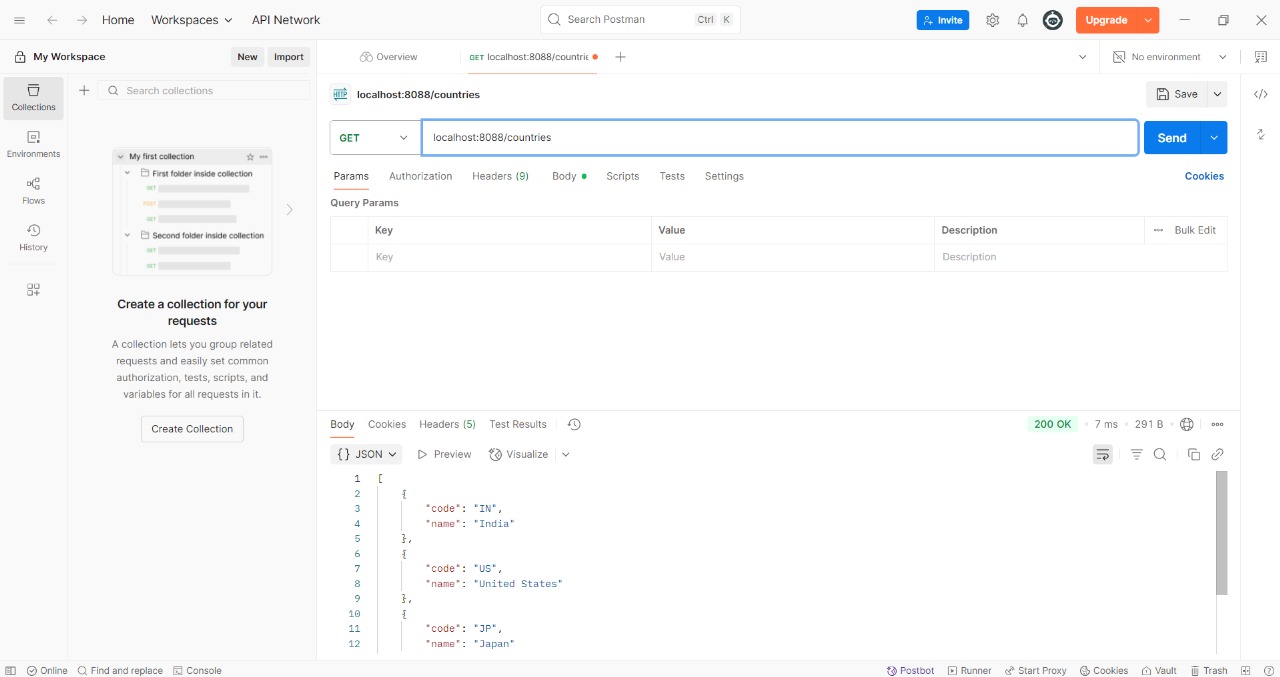
**public** **class** AppConfig {

}

**OUTPUT:**







**4. REST - Get country based on country code**   
  
Write a REST service that returns a specific country based on country code. The country code should be case insensitive.  
  
**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation:** @GetMapping("/countries/{code}")  
**Method Name**: getCountry(String code)  
**Method Implemetation**: Invoke countryService.getCountry(code)   
**Service Method:**com.cognizant.spring-learn.service.CountryService.getCountry(String code)  
  
**Service Method Implementation**:

* Get the country code using @PathVariable
* Get country list from country.xml
* Iterate through the country list
* Make a case insensitive matching of country code and return the country.
* Lambda expression can also be used instead of iterating the country list

**Sample Request**: http://localhost:8083/country/in  
  
**Sample Response**:

{

  "code": "IN",

  "name": "India"

}

**CountryController.java:**

**package** com.cognizant.spring\_learn.controller;

**import** com.cognizant.spring\_learn.model.Country;

**import** com.cognizant.spring\_learn.service.CountryService;

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

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

**public** **class** CountryController {

@Autowired

**private** CountryService countryService;

@GetMapping("/countries/{code}")

**public** Country getCountry(@PathVariable String code) {

**return** countryService.getCountry(code);

}

}

CountryService.java (Interface):

**package** com.cognizant.spring\_learn.service;

**import** com.cognizant.spring\_learn.model.Country;

**import** java.util.List;

**public** **interface** CountryService {

Country getCountry(String code);

List<Country> getAllCountries(); // Assuming you already have this for loading from XML

}

**CountryServiceImpl.java (Implementation):**

**package** com.cognizant.spring\_learn.service;

**import** com.cognizant.spring\_learn.model.Country;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** org.springframework.stereotype.Service;

**import** java.util.List;

@Service

**public** **class** CountryServiceImpl **implements** CountryService {

**private** List<Country> countryList;

**public** CountryServiceImpl() {

// Load the country list bean from country.xml on construction

countryList = (List<Country>) **new** ClassPathXmlApplicationContext("country.xml").getBean("countryList");

}

@Override

**public** Country getCountry(String code) {

// Case-insensitive search for country code

**return** countryList.stream()

.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(**null**);

}

@Override

**public** List<Country> getAllCountries() {

**return** countryList;

}

}

**country.xml:**

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="country1" class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

<bean id="country2" class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="US"/>

<property name="name" value="United States"/>

</bean>

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="country1"/>

<ref bean="country2"/>

</list>

</constructor-arg>

</bean>

</beans>

**Country.java:**

**package** com.cognizant.spring\_learn.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;

}

**}**

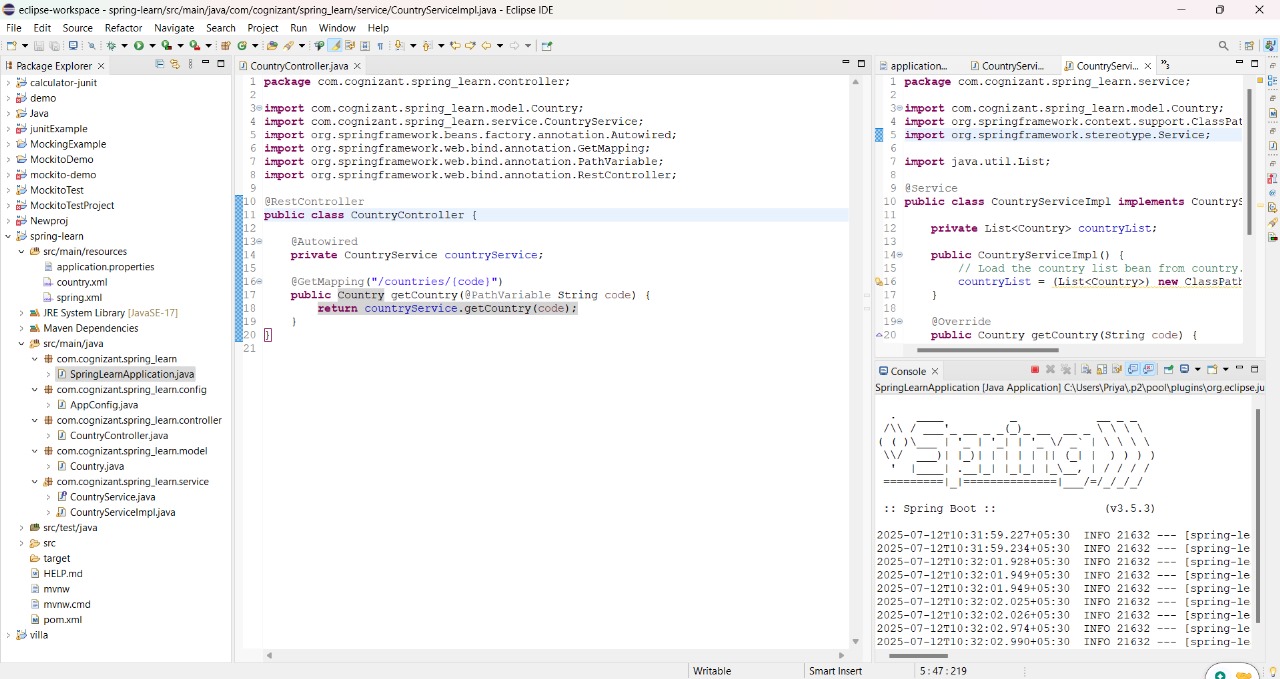
**URL Example (case-insensitive):**

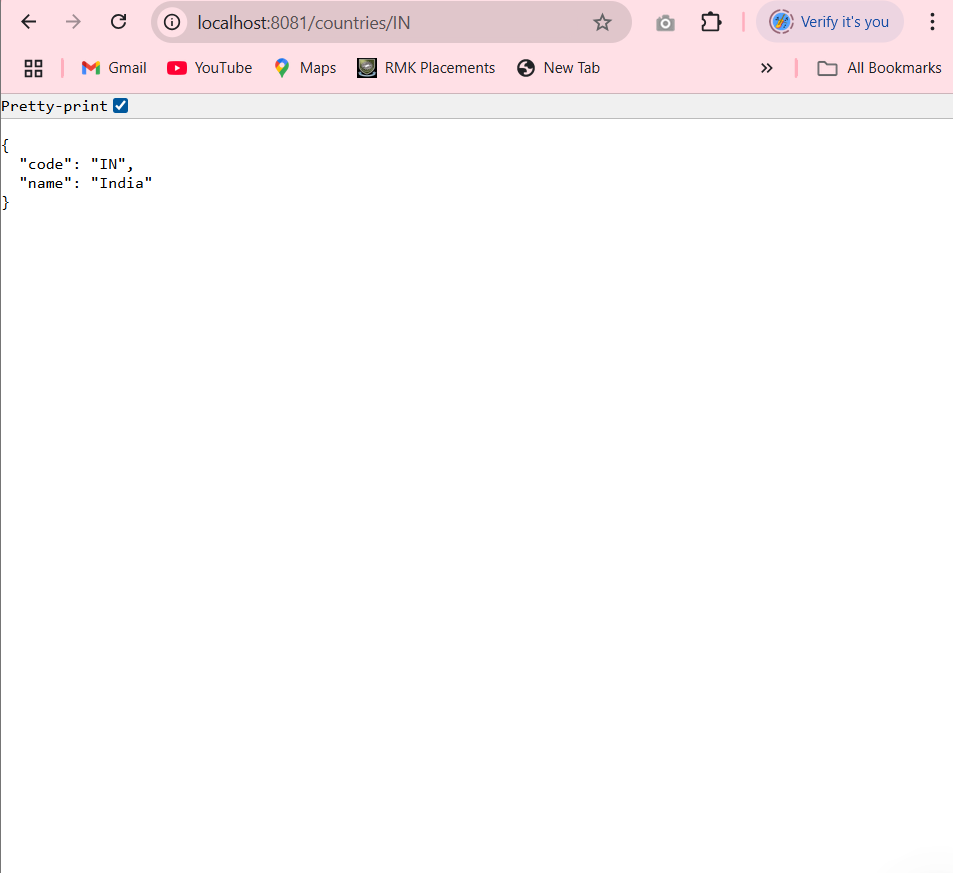
http://localhost:8083/countries/in

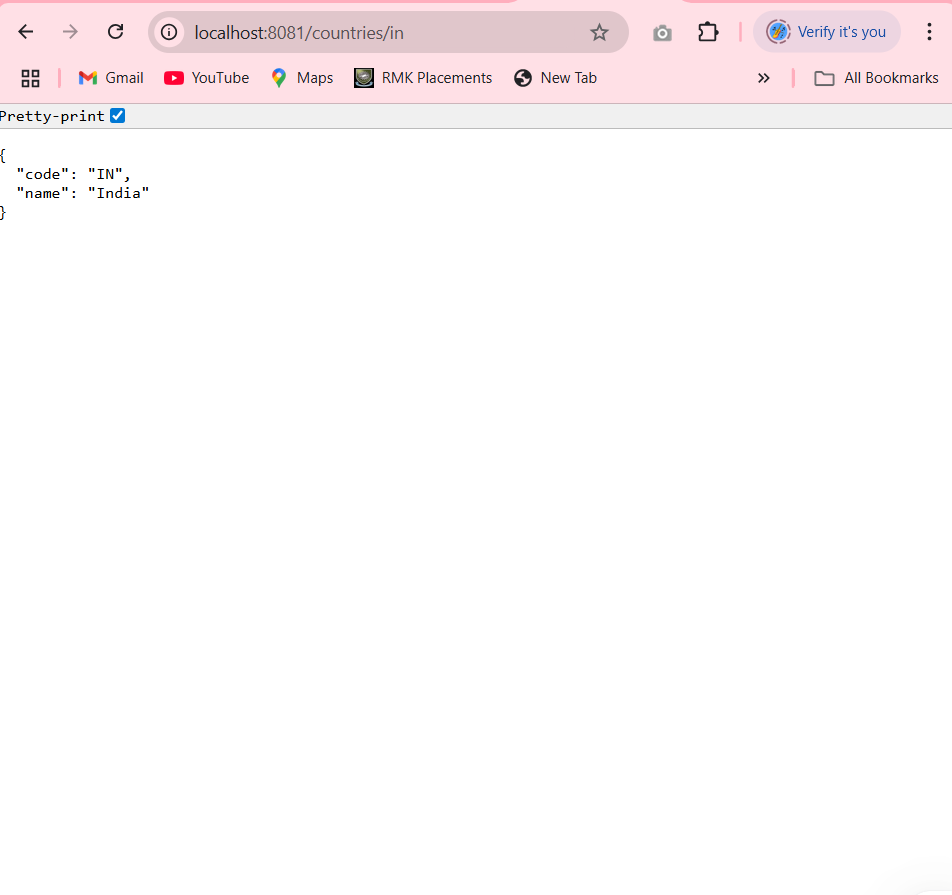
http://localhost:8083/countries/IN

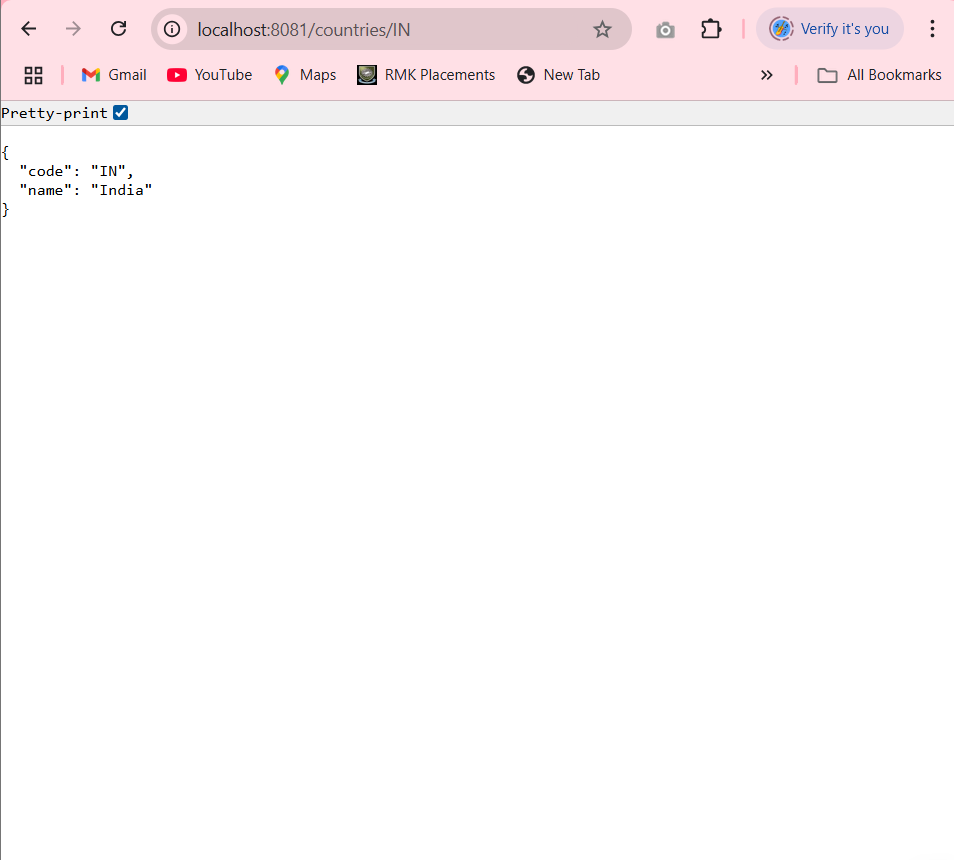
<http://localhost:8083/countries/In>

**OUTPUT:**









**5. 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"

}

**CountryNotFoundException.java**:

**package** com.cognizant.spring\_learn.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** Exception {

**public** CountryNotFoundException() {

**super**("Country not found");

}

**public** CountryNotFoundException(String message) {

**super**(message);

}

}

CountryService.java:

**package** com.cognizant.spring\_learn.service;

**import** com.cognizant.spring\_learn.model.Country;

**import** com.cognizant.spring\_learn.service.exception.CountryNotFoundException;

**public** **interface** CountryService {

Country getCountry(String code) **throws** CountryNotFoundException;

}

**CountryController.java:**

**package** com.cognizant.spring\_learn.controller;

**import** com.cognizant.spring\_learn.model.Country;

**import** com.cognizant.spring\_learn.service.CountryService;

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

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RestController;

**import** com.cognizant.spring\_learn.service.exception.CountryNotFoundException;

@RestController

**public** **class** CountryController {

@Autowired

**private** CountryService countryService;

@GetMapping("/country/{code}")

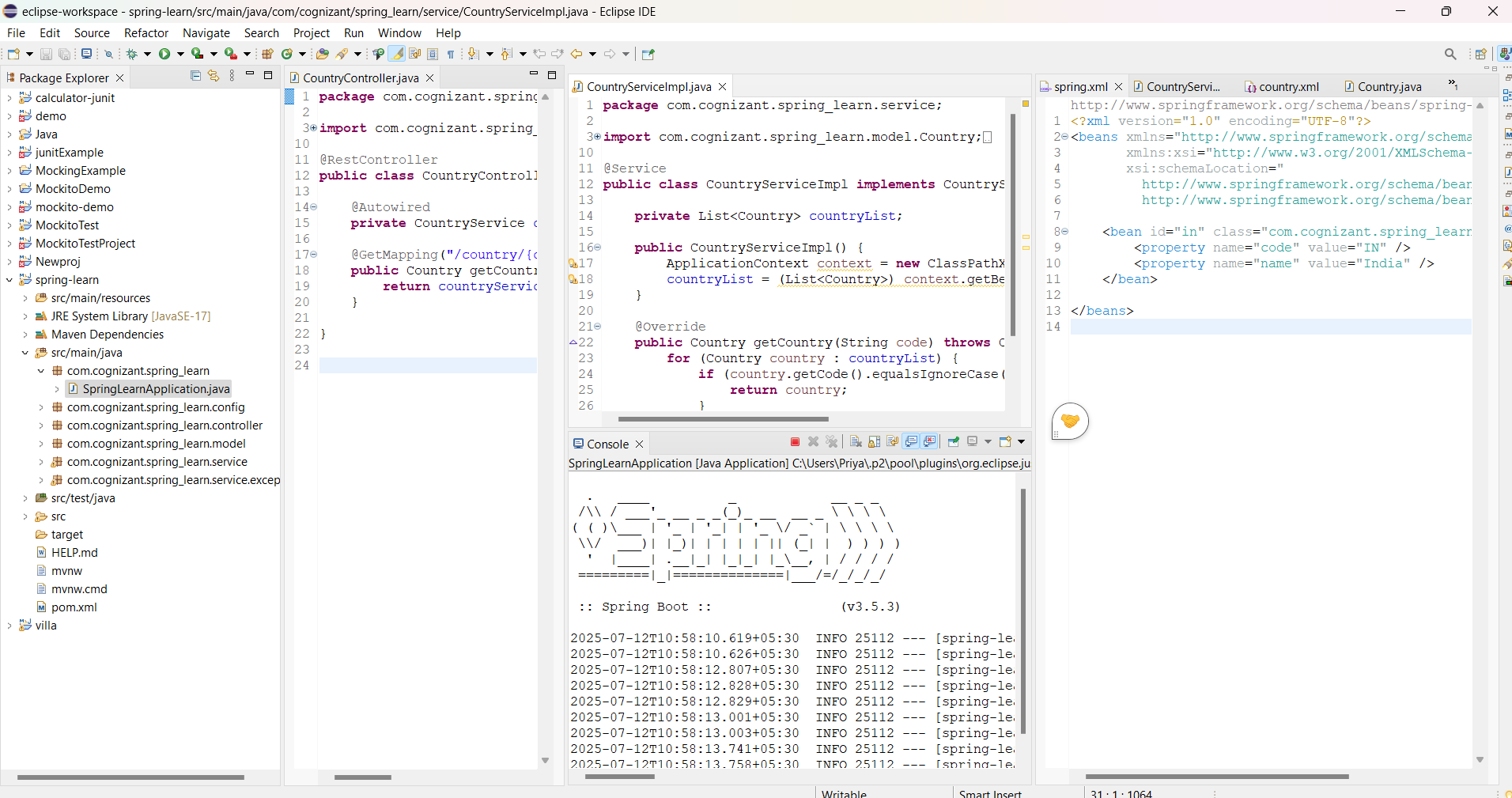
**public** Country getCountry(@PathVariable String code) **throws** CountryNotFoundException {

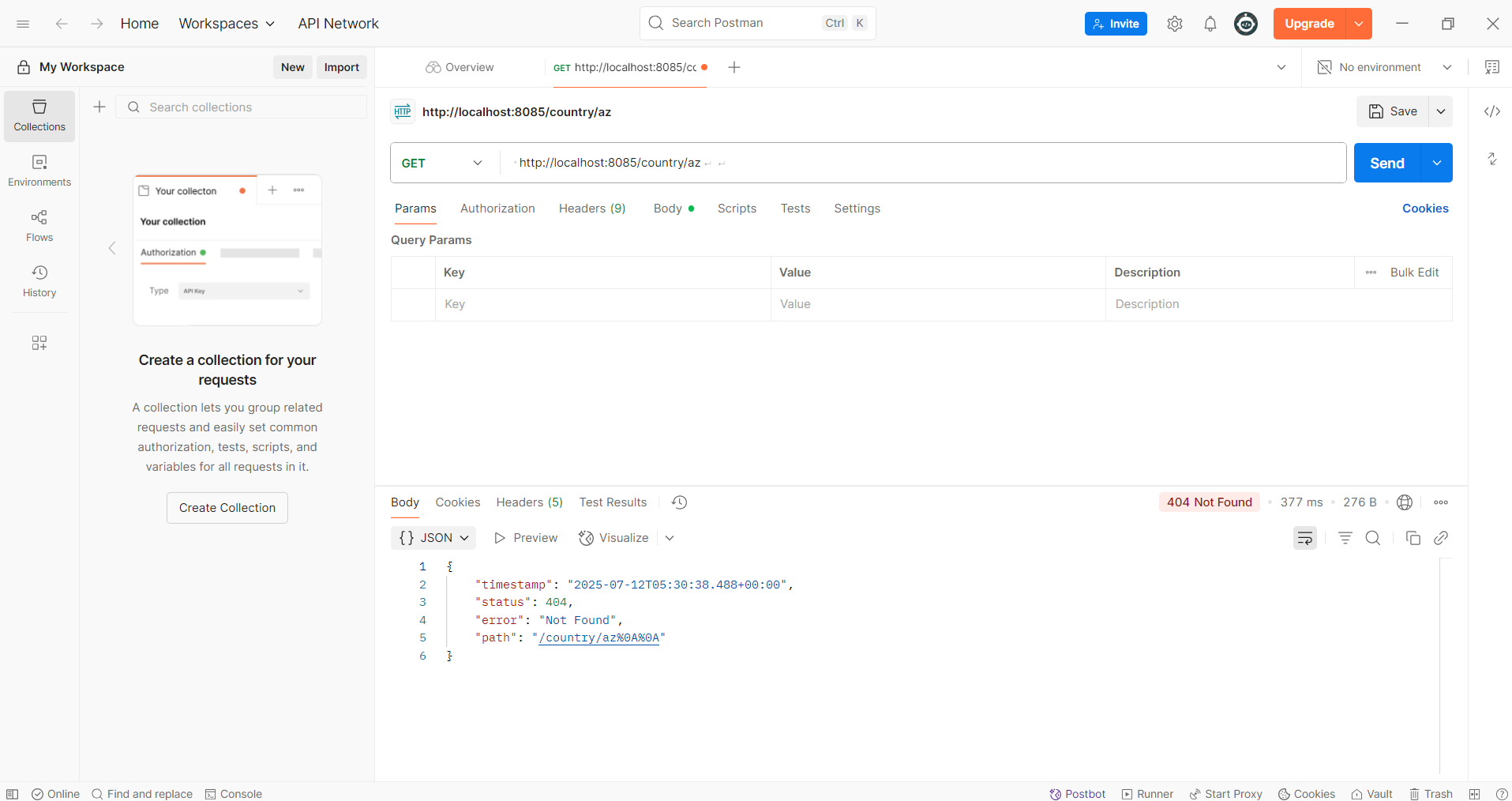
**return** countryService.getCountry(code);

}

}

**OUTPUT:**





**6. 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"

**CountryControoller.java:**

package com.example.controller;

import com.example.model.Country;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class CountryController {

public CountryController() {

System.out.println("CountryController Constructor Called");

}

@GetMapping("/country")

public Country getCountry() {

return new Country("IN", "India");

}

}

**SpringLearnApplicationTests.java:**

package com.cognizant.spring\_learn;

import com.cognizant.spring\_learn.controller.CountryController;

import org.junit.jupiter.api.Test;

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;

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.\*;

@SpringBootTest

@AutoConfigureMockMvc

public class SpringLearnApplicationTests {

@Autowired

private MockMvc mvc;

@Test

public void testGetCountry() throws Exception {

mvc.perform(get("/country"))

.andExpect(status().isOk())

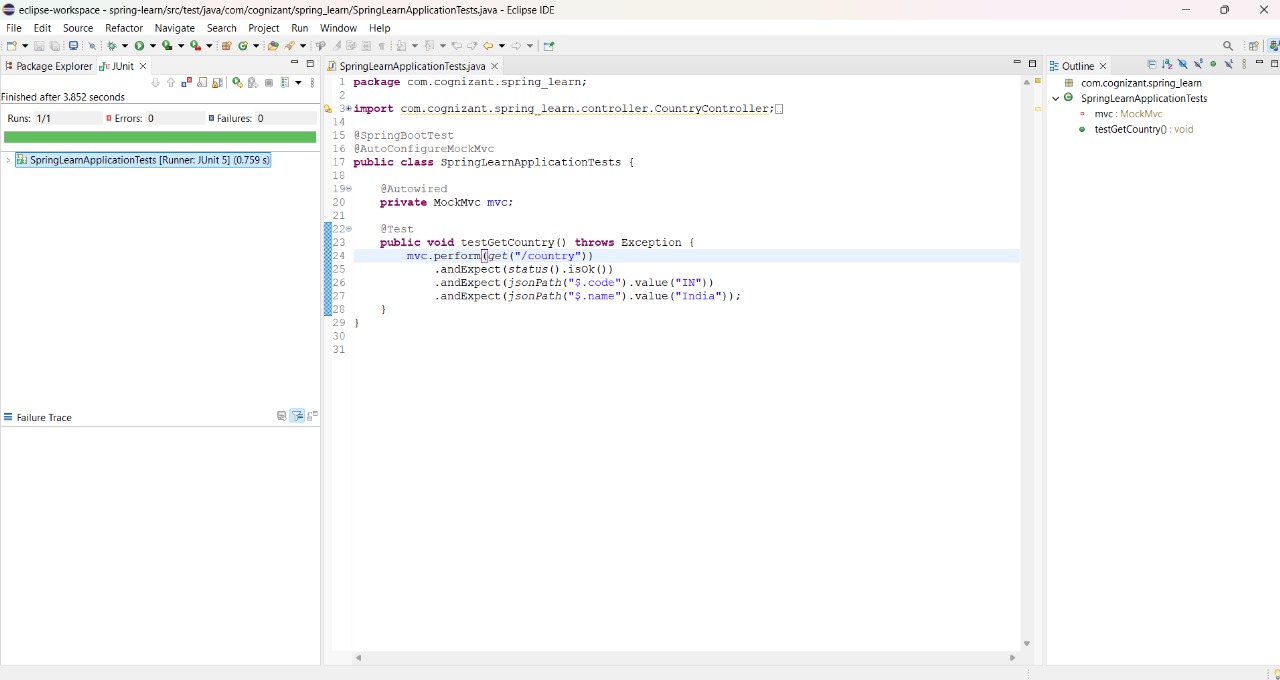
.andExpect(jsonPath("$.code").value("IN"))

.andExpect(jsonPath("$.name").value("India"));

}

}

**OUTPUT:**



**7. 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"));

**Country.java**:

**package** com.cognizant.spring\_learn.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.spring\_learn.controller;

**import** com.cognizant.spring\_learn.model.Country;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.web.bind.annotation.\*;

@RestController

**public** **class** CountryController {

**public** CountryController() {

System.***out***.println("CountryController Constructor Called");

}

@GetMapping("/country")

**public** Country getCountry(@RequestParam(required = **false**) String error) {

**if** ("true".equalsIgnoreCase(error)) {

**throw** **new** CountryNotFoundException();

}

**return** **new** Country("IN", "India");

}

// Custom Exception with status and reason

@ResponseStatus(code = HttpStatus.***BAD\_REQUEST***, reason = "Country Not found")

**public** **static** **class** CountryNotFoundException **extends** RuntimeException {

}

}

**SpringLearnApplicationTests.java:**

**package** com.cognizant.spring\_learn;

**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.*status*;

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

**import** com.cognizant.spring\_learn.controller.CountryController;

**import** org.junit.jupiter.api.Test;

**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** SpringLearnApplicationTests {

@Autowired

**private** CountryController countryController;

@Autowired

**private** MockMvc mvc;

// ✅ Test if controller loads

@Test

**public** **void** contextLoads() {

*assertNotNull*(countryController);

}

// ✅ Test normal success case

@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"));

}

// ✅ Test exceptional case

@Test

**public** **void** testGetCountryException() **throws** Exception {

ResultActions actions = mvc.perform(*get*("/country?error=true"));

actions.andExpect(*status*().isBadRequest()); // 400 Bad Request

actions.andExpect(*status*().reason("Country Not found")); // Custom reason phrase

}

}

**OUTPUT:**

