**WEEK-3**

**Spring REST using Spring Boot**

**Superset ID: 6419740**

**1.HTTP Request Response**

**Code:**

**HttpDemoApplication.java:**

package com.example.httpdemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class HttpDemoApplication {

public static void main(String[] args) {

SpringApplication.run(HttpDemoApplication.class, args);

}

}

**HelloController.java:**

package com.example.httpdemo.controller;

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

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

@RestController

public class HelloController {

@GetMapping("/hello")

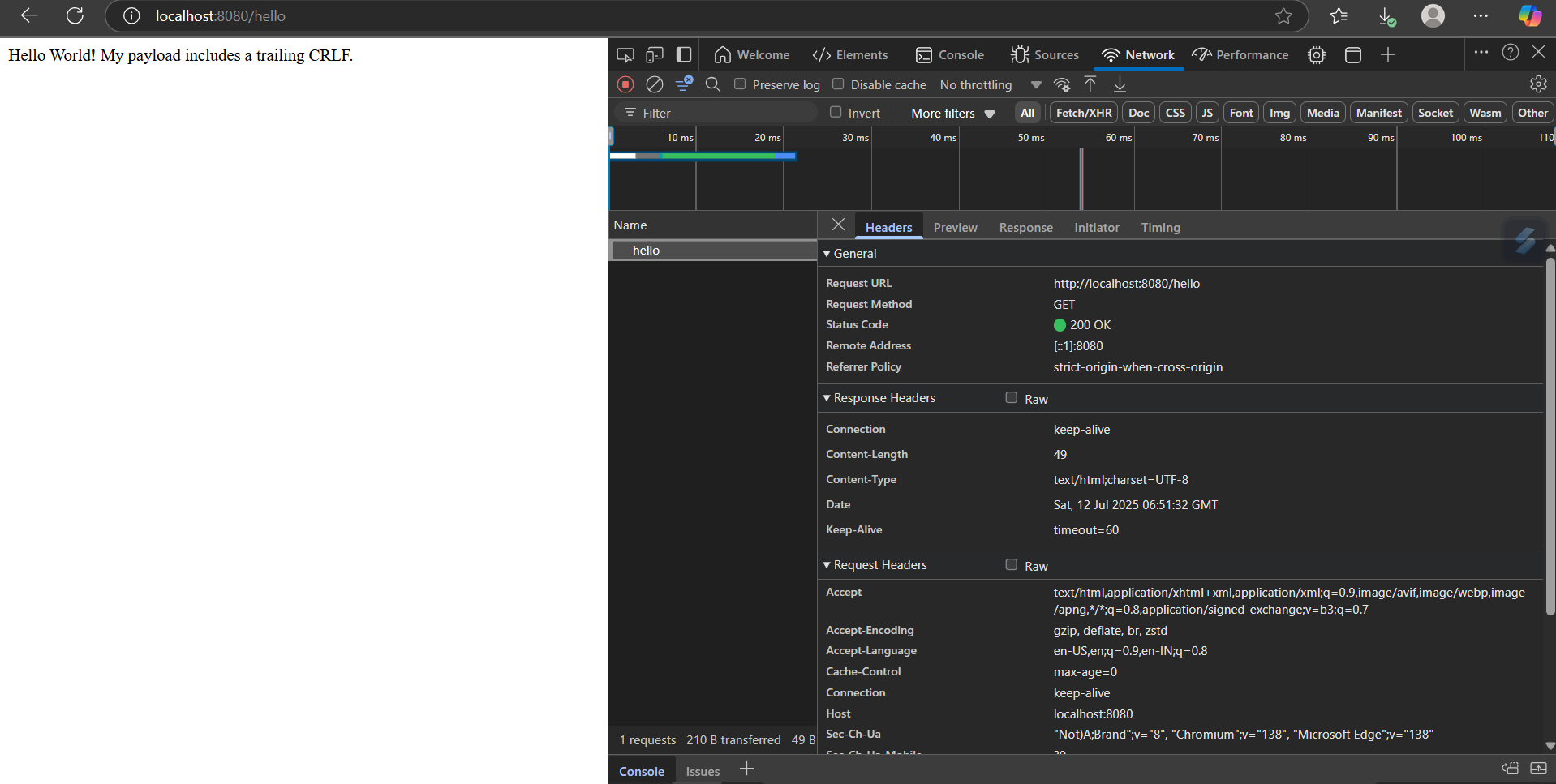
public String sayHello() {

return "Hello World! My payload includes a trailing CRLF.";

}

}

**Output:**



**2.Hello World RESTful Web Service**   
**Code:**

**HelloController.java:**

package com.cognizant.springlearn.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");

String message = "Hello World!!";

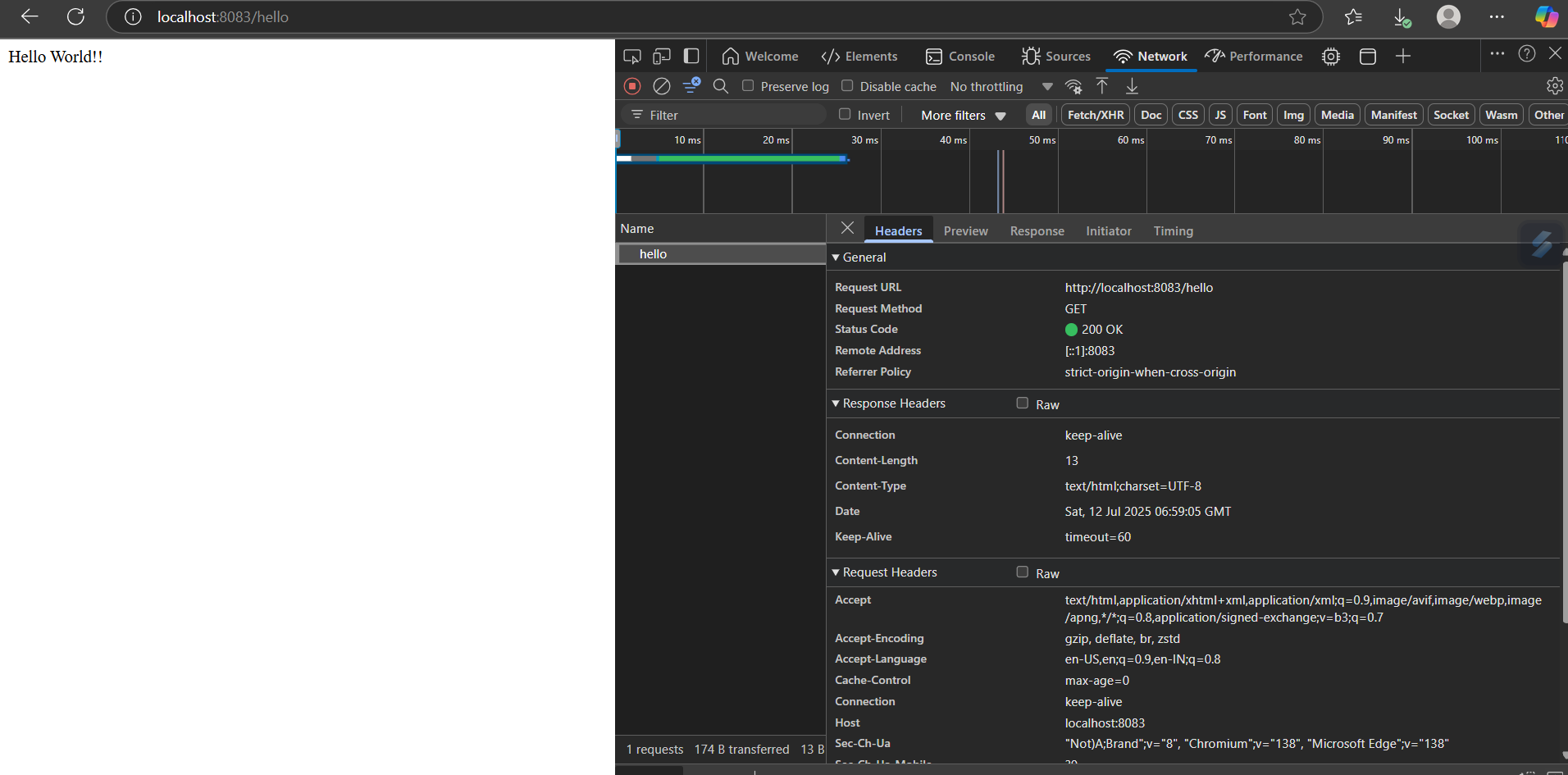
LOGGER.info("END");

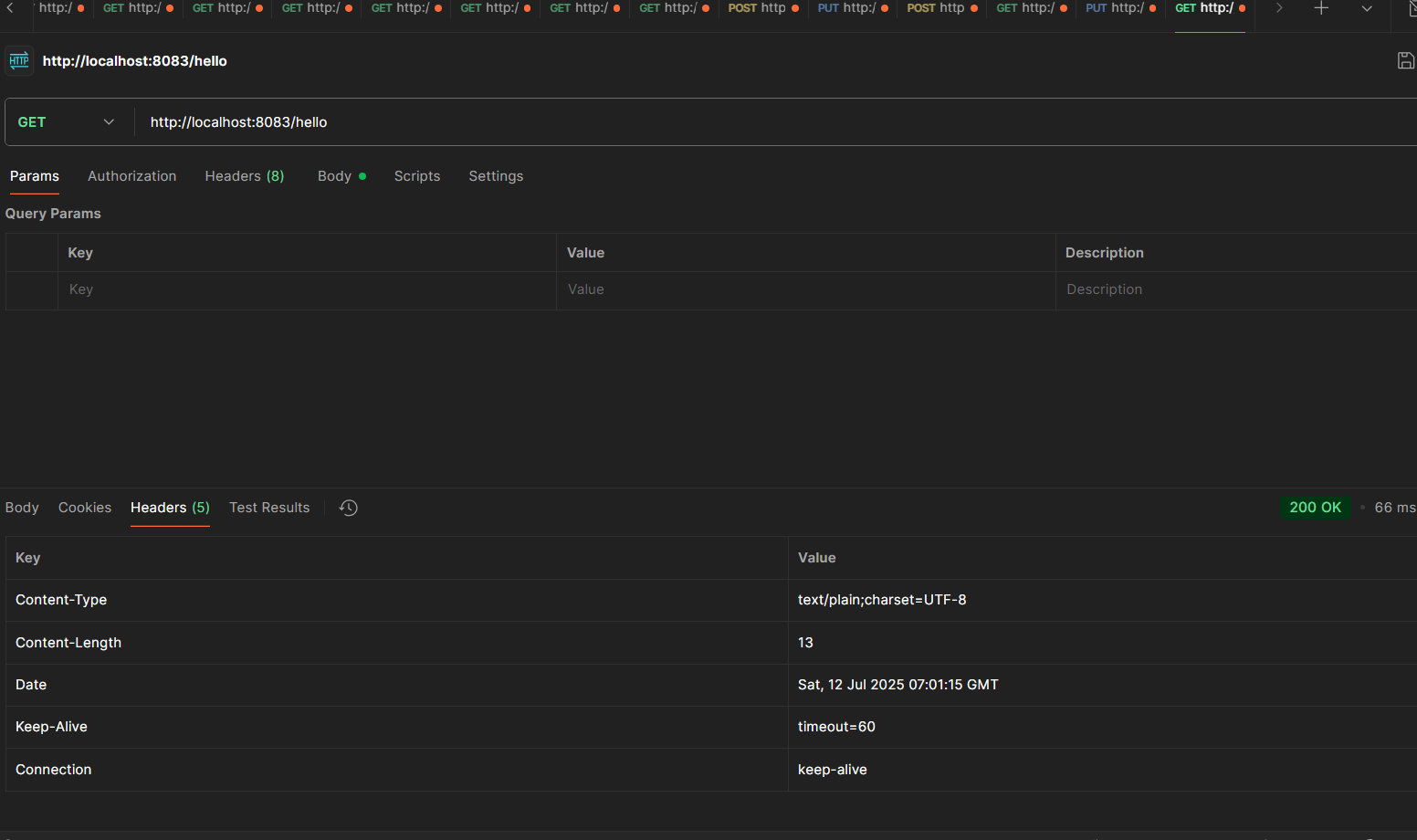
return message;

}

}

**Output:**

****

****

**3. REST - Country Web Service**

**Code:**

**CountryController.java:**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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 {

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

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START");

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

Country country = context.getBean("country", Country.class);

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

LOGGER.info("END");

return country;

}

}

**Country.java:**

package com.cognizant.springlearn;

public class Country {

private String code;

private String name;

public Country() {

System.out.println("Inside Country Constructor.");

}

public String getCode() {

System.out.println("Inside getCode()");

return code;

}

public void setCode(String code) {

System.out.println("Inside setCode()");

this.code = code;

}

public String getName() {

System.out.println("Inside getName()");

return name;

}

public void setName(String name) {

System.out.println("Inside setName()");

this.name = name;

}

@Override

public String toString() {

return "Country{" +

"code='" + code + '\'' +

", name='" + name + '\'' +

'}';

}

}

**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"

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

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

<bean id="country" class="com.cognizant.springlearn.Country">

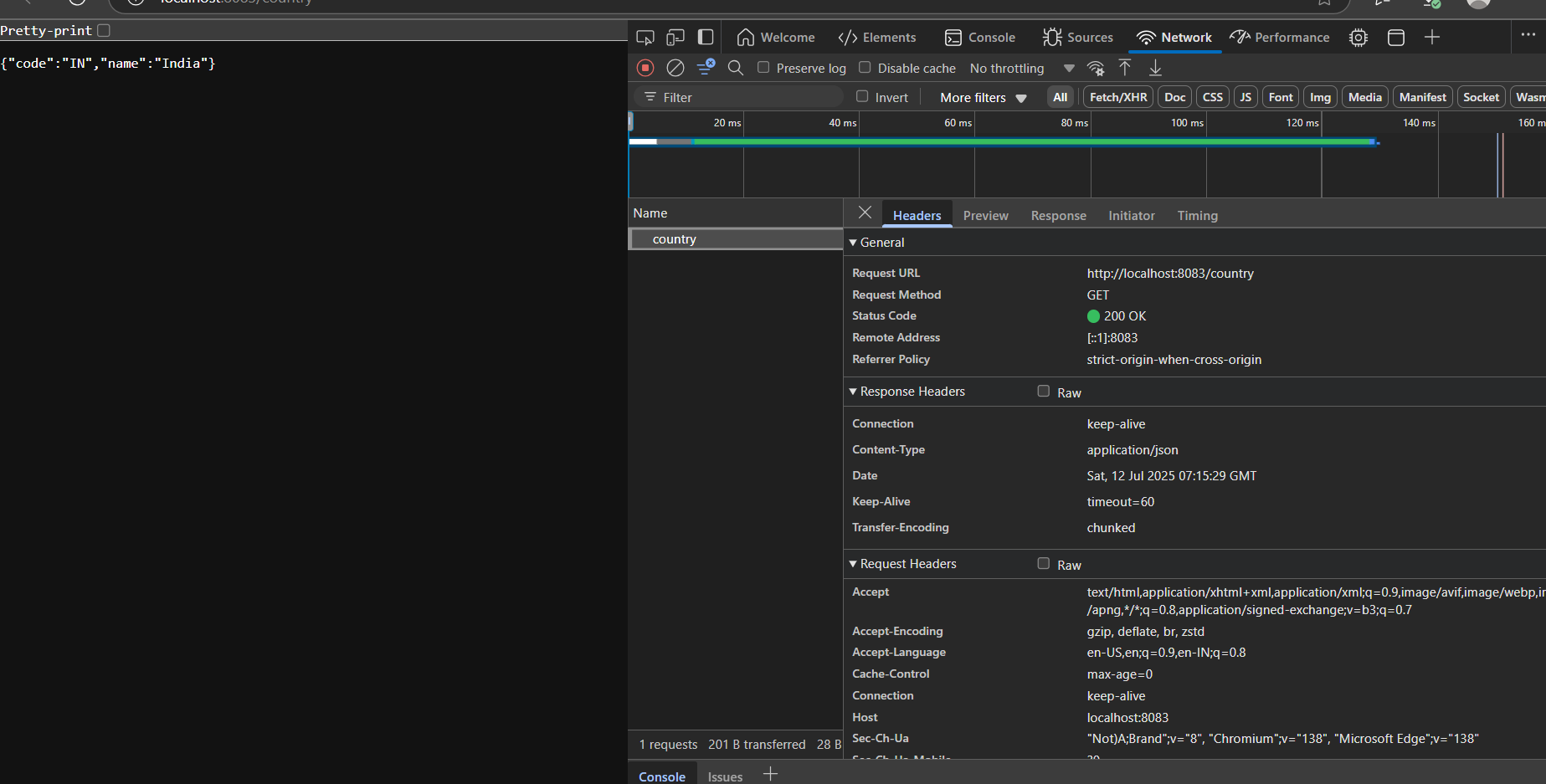
<property name="code" value="IN" />

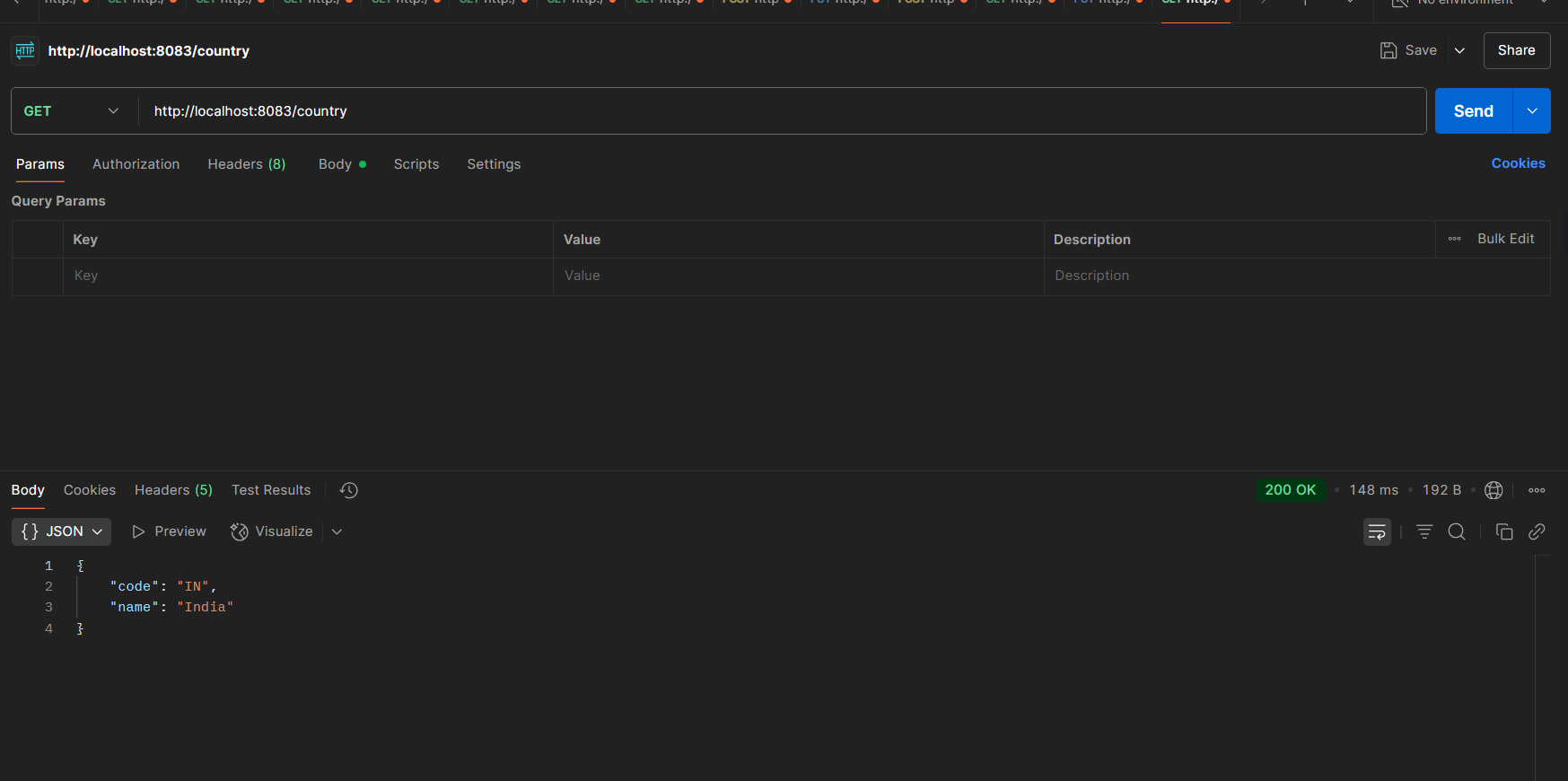
<property name="name" value="India" />

</bean>

</beans>

**Output:**

****

****

**4.REST - Get all countries**

**Code:**

**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"

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

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

<bean id="country" class="com.example.httpdemo.Country">

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

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

</bean>

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

<constructor-arg>

<list>

<bean class="com.example.httpdemo.Country">

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

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

</bean>

<bean class="com.example.httpdemo.Country">

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

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

</bean>

<bean class="com.example.httpdemo.Country">

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

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

</bean>

<bean class="com.example.httpdemo.Country">

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

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

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**CountryController.java:**

package com.example.httpdemo.controller;

import com.example.httpdemo.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.util.List;

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START");

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

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

LOGGER.info("END");

return country;

}

@GetMapping("/countries")

public List<Country> getAllCountries() {

LOGGER.info("START - getAllCountries");

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

List<Country> countries = (List<Country>) context.getBean("countryList");

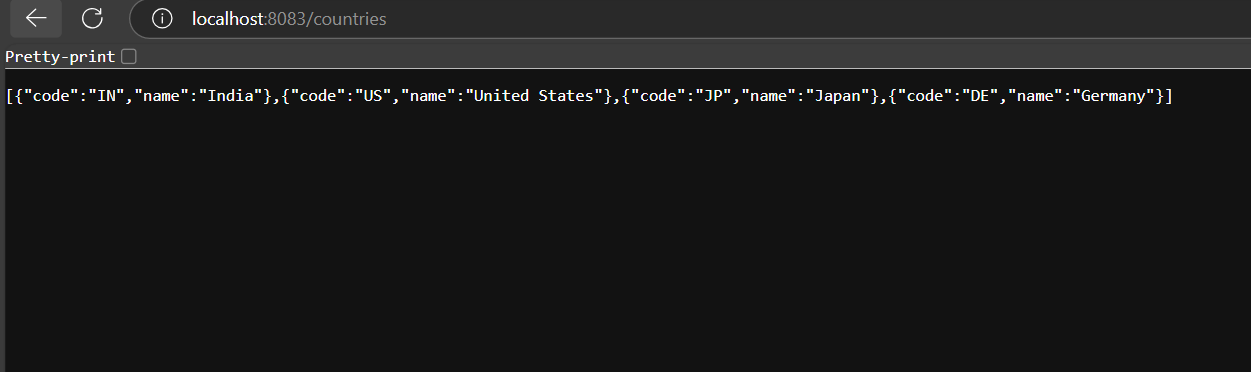
LOGGER.info("END - getAllCountries");

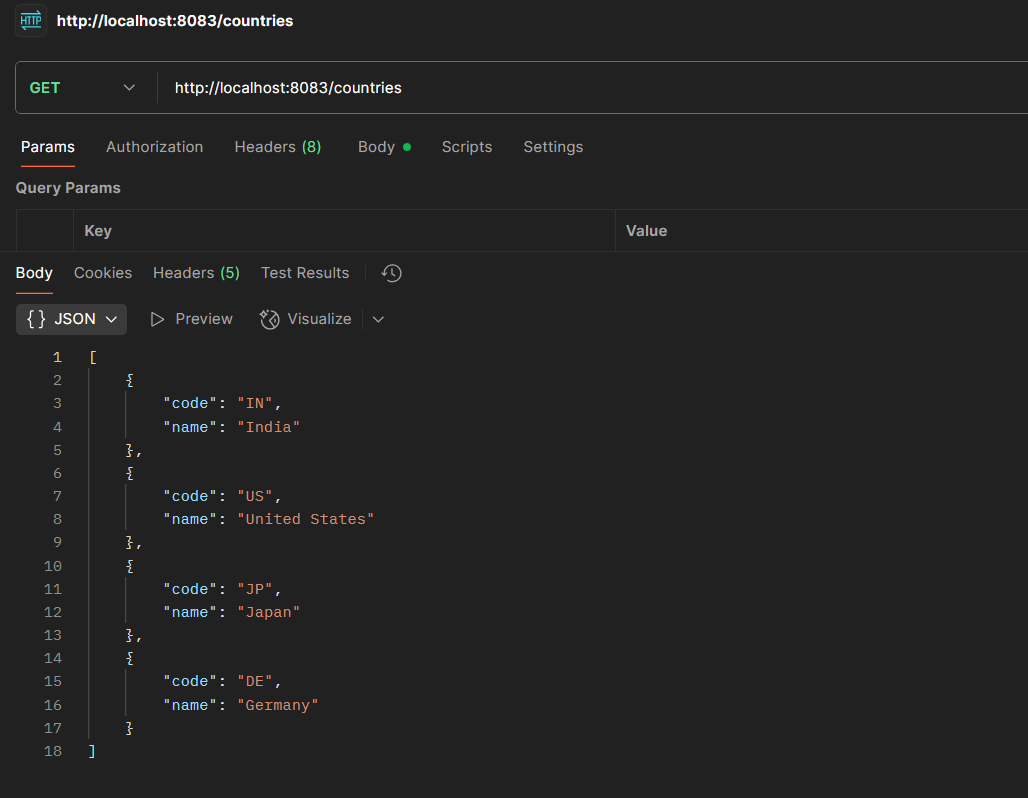
return countries;

}

}

**Output:**





**5.REST - Get country based on country code**

**Code:**

**CountryService.java:**

package com.example.httpdemo.service;

import com.example.httpdemo.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) {

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

List<Country> countries = (List<Country>) context.getBean("countryList");

return countries.stream()

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

.findFirst()

.orElse(null); // Or throw a custom CountryNotFoundException

}

}

**CountryController.java:**

package com.example.httpdemo.controller;

import com.example.httpdemo.Country;

import com.example.httpdemo.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class CountryController {

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

@Autowired

private CountryService countryService;

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

public Country getCountry(@PathVariable String code) {

LOGGER.info("START - getCountry");

Country country = countryService.getCountry(code);

LOGGER.info("END - getCountry");

return country;

}

}

**Output:**

****

**6.REST - Get country exceptional scenario**

**Code:**

**CountryController.java:**

package com.example.httpdemo.controller;

import com.example.httpdemo.Country;

import com.example.httpdemo.exception.CountryNotFoundException;

import com.example.httpdemo.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class CountryController {

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

@Autowired

private CountryService countryService;

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

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

LOGGER.info("START - getCountry");

Country country = countryService.getCountry(code);

LOGGER.info("END - getCountry");

return country;

}

}

**CountryService.java:**

package com.example.httpdemo.service;

import com.example.httpdemo.Country;

import com.example.httpdemo.exception.CountryNotFoundException;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) {

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

List<Country> countries = (List<Country>) context.getBean("countryList");

return countries.stream()

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

.findFirst()

.orElseThrow(() -> new CountryNotFoundException("Country not found"));

}

}

**CountryNotFoundException.java:**

package com.example.httpdemo.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 {

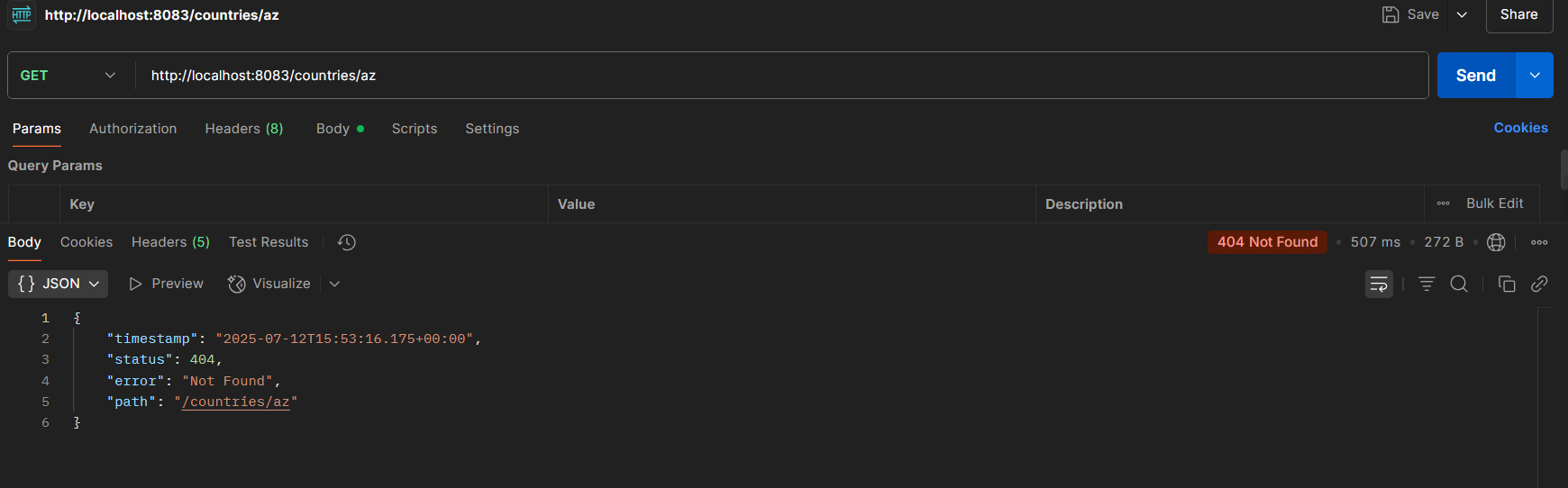
public CountryNotFoundException(String message) {

super(message);

}

}

**Output:**

****

**7.MockMVC - Test get country service**

**Code:**

**SpringLearnApplicationTests.java:**

package com.example.httpdemo;

import com.example.httpdemo.controller.CountryController;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertNotNull;

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

import org.springframework.boot.test.context.SpringBootTest;

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

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

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

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 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"));

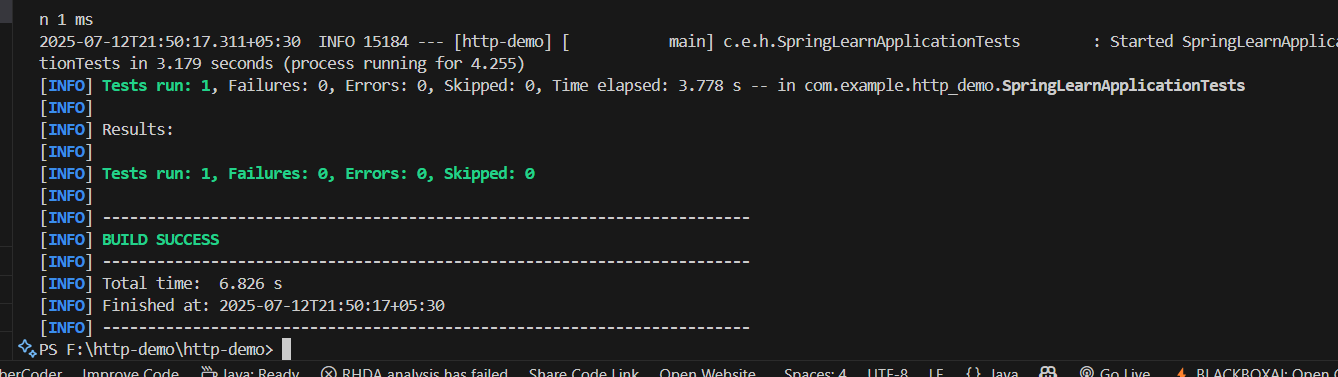
actions.andExpect(jsonPath("$.name").exists());

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

}

}

**Output:**



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

**GlobalExceptionHandler.java:**

package com.example.http\_demo.exception;

import org.springframework.http.HttpStatus;

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

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

@ControllerAdvice

public class GlobalExceptionHandler {

    @ExceptionHandler(CountryNotFoundException.class)

    public void handleCountryNotFoundException(HttpServletResponse response) throws IOException {

        response.sendError(HttpStatus.BAD\_REQUEST.value(), "Country Not found");

    }

}

**SpringLearnApplicationTests.java:**

package com.example.http\_demo.exception;

import org.springframework.http.HttpStatus;

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

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

@ControllerAdvice

public class GlobalExceptionHandler {

    @ExceptionHandler(CountryNotFoundException.class)

    public void handleCountryNotFoundException(HttpServletResponse response) throws IOException {

        response.sendError(HttpStatus.BAD\_REQUEST.value(), "Country Not found");

    }

}

**Output:**

