**WEEK-4**

**Spring REST using Spring Boot 3**

1. **Create a Spring Web Project using Maven**

**Code:**

SpringLearnApplication.java

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

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

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

**public** **static** **void** main(String[] args) {

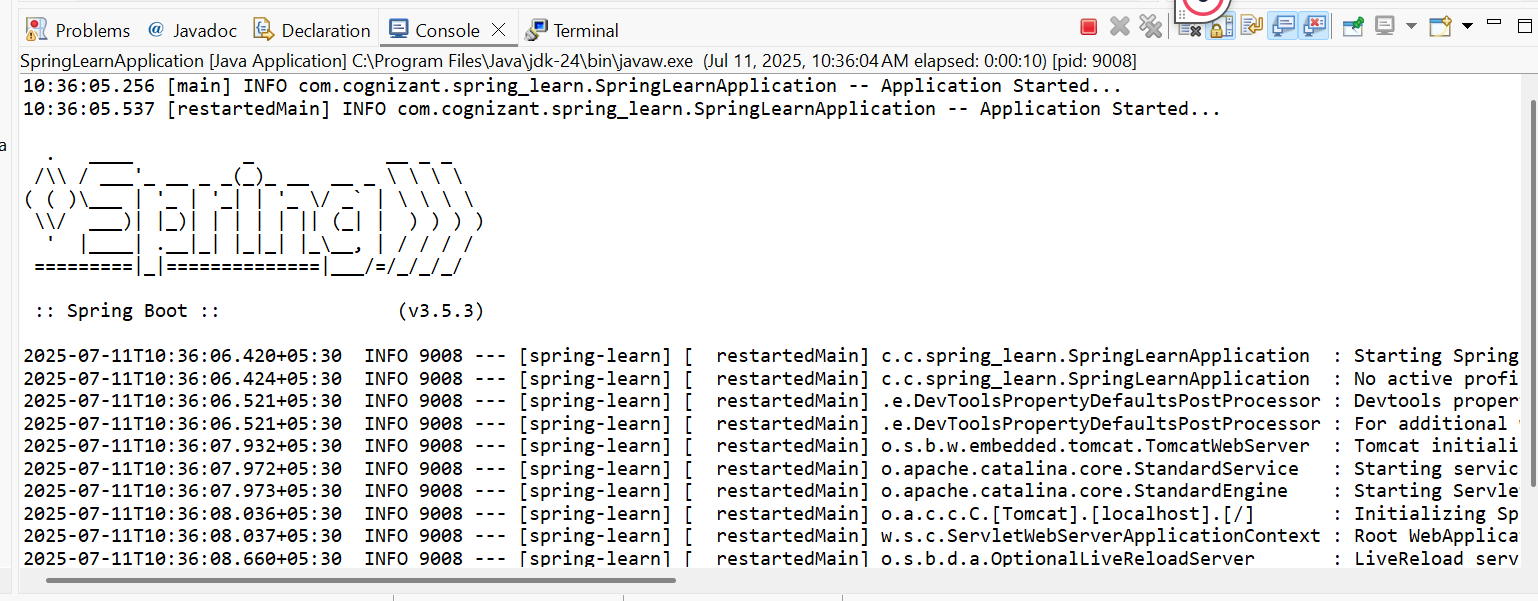
***LOGGER***.info("Application Started...");

SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

}

**Output**:



1. **Spring Core – Load Country from Spring Configuration XML**

**Code:**

Country.java

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

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**public** **class** Country {

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

**private** String code;

**private** String name;

**public** Country() {

***LOGGER***.debug("Inside Country Constructor.");

}

**public** String getCode() {

//LOGGER.debug("Inside getCode");

**return** code;

}

**public** **void** setCode(String code) {

//LOGGER.debug("Inside setCode");

**this**.code = code;

}

**public** String getName() {

//LOGGER.debug("Inside getName");

**return** name;

}

**public** **void** setName(String name) {

//LOGGER.debug("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>">

<!-- Individual beans (optional) -->

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

<!-- List bean -->

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

<constructor-arg>

<list>

<ref bean="countryIN"/>

<ref bean="countryUS"/>

<ref bean="countryDE"/>

<ref bean="countryJP"/>

</list>

</constructor-arg>

</bean>

</beans>

SpringLearnApplication.java

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

**import** java.util.List;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.ApplicationContext;

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

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

@SpringBootApplication

**public** **class** SpringLearnApplication {

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

**public** **static** **void** main(String[] args) {

***LOGGER***.info("Application Started...");

SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

@Bean

**public** CommandLineRunner run() {

**return** args -> {

*displayCountry*();

*displayCountryList*();

};

}

**public** **static** **void** displayCountry() {

***LOGGER***.info("START displayCountry()");

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

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

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

((ClassPathXmlApplicationContext) context).close();

}

**public** **static** **void** displayCountryList() {

***LOGGER***.info("START displayCountryList()");

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

List<Country> countryList = context.getBean("countryList", java.util.List.**class**);

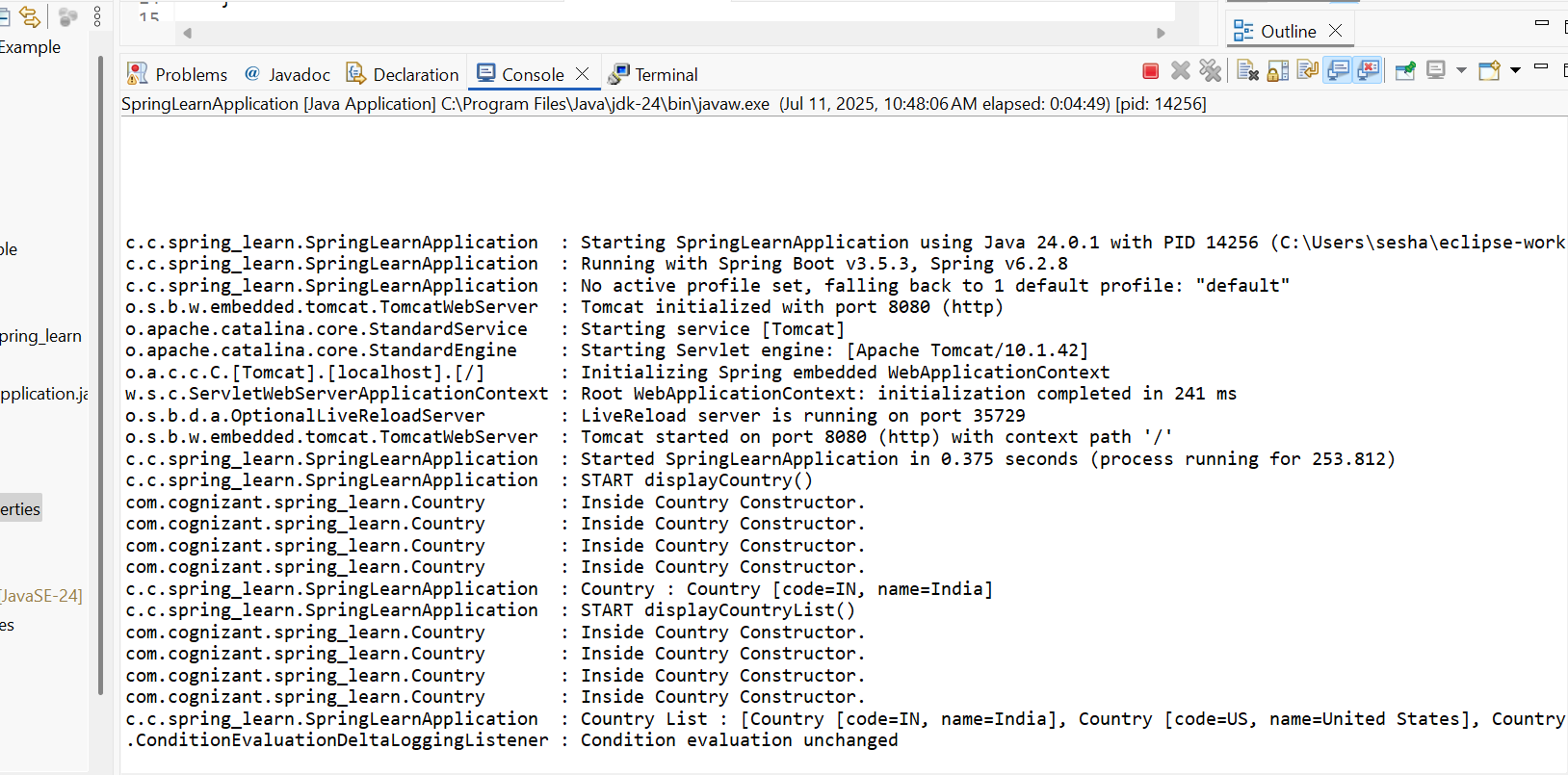
***LOGGER***.debug("Country List : {}", countryList);

((ClassPathXmlApplicationContext) context).close();

}

}

**Output**:



1. **Hello World RESTful Web Service**

**Code:**

**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!!";

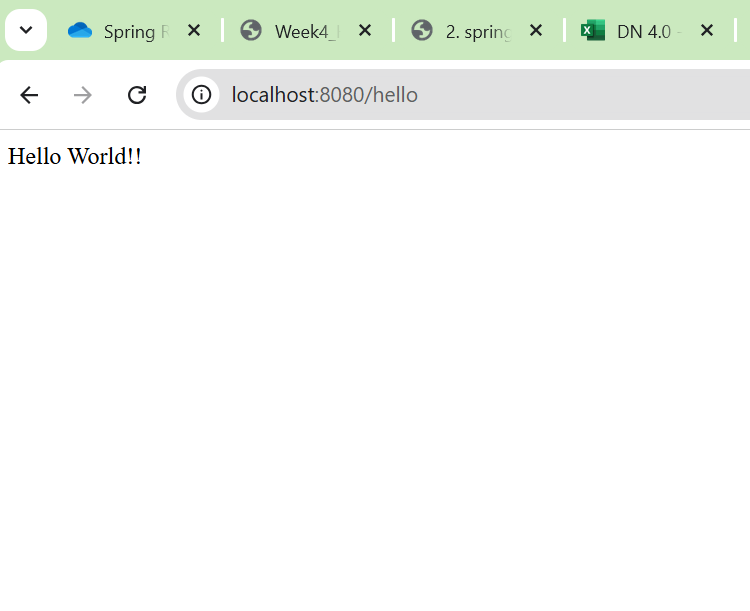
***LOGGER***.info("END sayHello()");

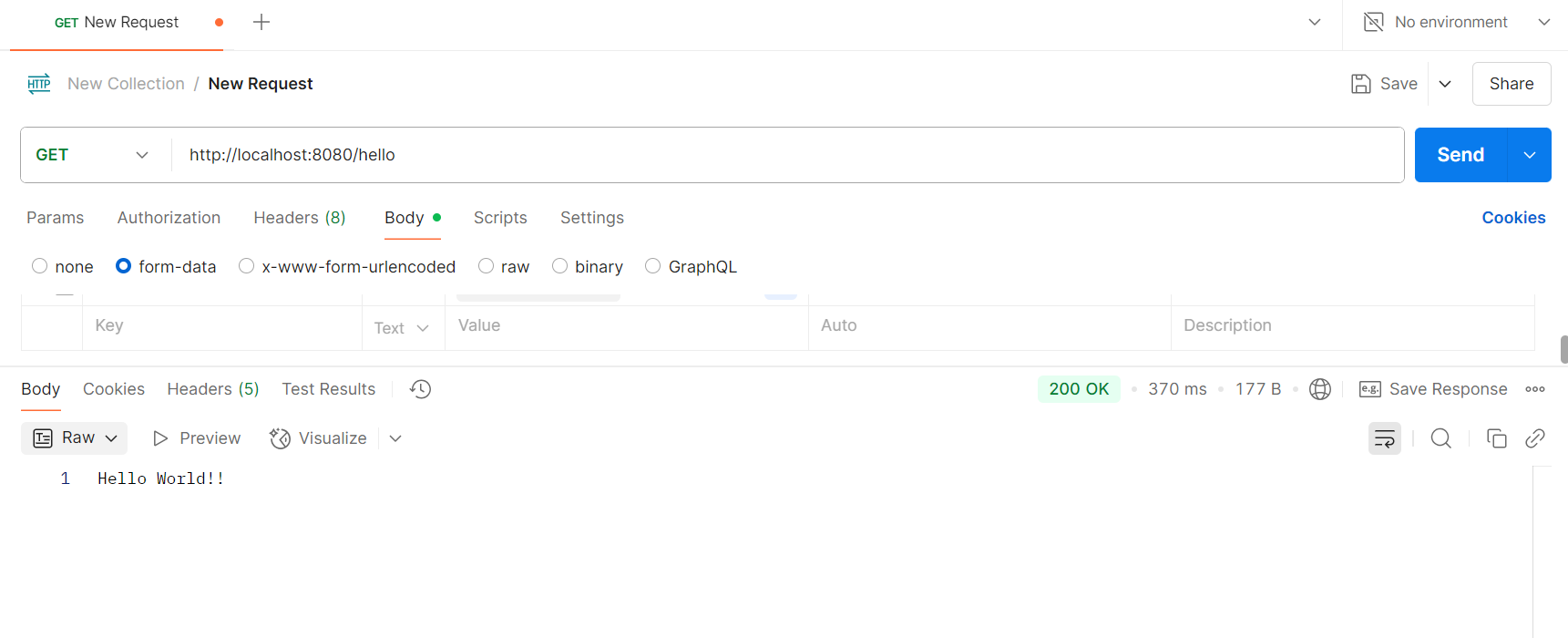
**return** message;

}

}

**Output**:





1. **REST - Country Web Service**

**Code:**

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

**import** com.cognizant.spring\_learn.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** HelloController {

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

@RequestMapping("/country")

**public** Country getCountryIndia() {

***LOGGER***.info("START getCountryIndia()");

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

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

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

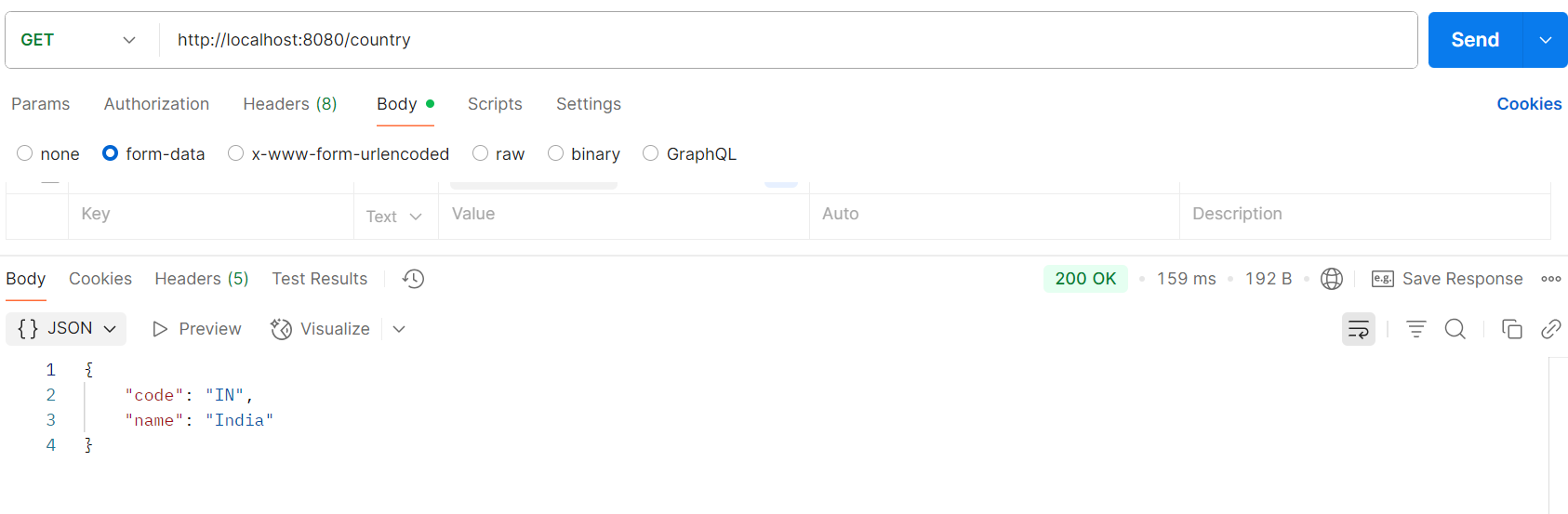
***LOGGER***.info("END getCountryIndia()");

**return** country;

}

}

**Output**:



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

**Code:**

CountryService.java

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.Country;

import org.springframework.stereotype.Service;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) {

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

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

// Using lambda expression for case-insensitive match

return countryList.stream()

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

.findFirst()

.orElse(null);

}

}

CountryController.java

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import com.cognizant.spring\_learn.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() - code: {}", code);

Country country = countryService.getCountry(code);

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

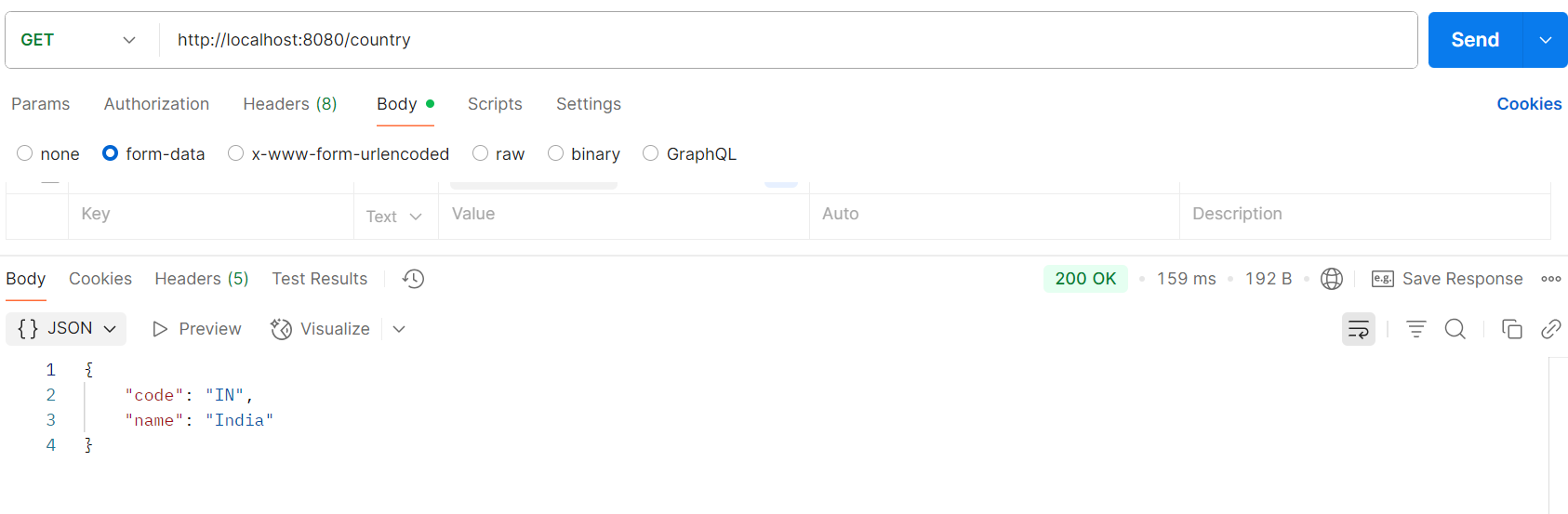
LOGGER.info("END getCountry()");

return country;

}

}

**Output**:



1. **Create authentication service that returns JWT**

**Code:**

AuthenticationController.java

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

**import** io.jsonwebtoken.Jwts;

**import** io.jsonwebtoken.SignatureAlgorithm;

**import** jakarta.servlet.http.HttpServletRequest;

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

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

**import** java.util.Base64;

**import** java.util.Date;

@RestController

**public** **class** AuthenticationController {

@GetMapping("/authenticate")

**public** String authenticate(HttpServletRequest request) {

String authHeader = request.getHeader("Authorization");

**if** (authHeader == **null** || !authHeader.startsWith("Basic ")) {

**throw** **new** RuntimeException("Missing or invalid Authorization header");

}

String base64Credentials = authHeader.substring("Basic ".length());

**byte**[] decoded = Base64.*getDecoder*().decode(base64Credentials);

String credentials = **new** String(decoded);

String[] values = credentials.split(":", 2);

String username = values[0];

String password = values[1];

**if** (!username.equals("user") || !password.equals("pwd")) {

**throw** **new** RuntimeException("Invalid credentials");

}

String token = Jwts.*builder*()

.setSubject(username)

.setIssuedAt(**new** Date())

.setExpiration(**new** Date(System.*currentTimeMillis*() + 60 \* 60 \* 1000)) // 1 hour

.signWith(SignatureAlgorithm.***HS256***, "secret")

.compact();

**return** "{ \"token\": \"" + token + "\" }";

}

}

SecurityConfig.java

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

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

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

**import** org.springframework.security.config.annotation.web.builders.HttpSecurity;

**import** org.springframework.security.core.userdetails.User;

**import** org.springframework.security.core.userdetails.UserDetails;

**import** org.springframework.security.core.userdetails.UserDetailsService;

**import** org.springframework.security.provisioning.InMemoryUserDetailsManager;

**import** org.springframework.security.web.SecurityFilterChain;

@Configuration

**public** **class** SecurityConfig {

@Bean

**public** SecurityFilterChain filterChain(HttpSecurity http) **throws** Exception {

http

.csrf(csrf -> csrf.disable())

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").permitAll()

.anyRequest().authenticated())

.~~httpBasic~~();

**return** http.build();

}

@Bean

**public** UserDetailsService userDetailsService() {

UserDetails user = User.*withUsername*("user")

.password("{noop}pwd") // no encoding for demo

.roles("USER")

.build();

**return** **new** InMemoryUserDetailsManager(user);

}

}

**Output**:

{ "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzI2Mzg1MjAwLCJleHAiOjE3MjYzODg4MDB9.\_YAfN9pHMyiZ1f7YlZ\_cHvppYBkSuFcBtUkRQn1u4JA" }