**Spring REST using Spring Boot 3**

**Exercise 1: Create a Spring Web Project using Maven**

**Code**

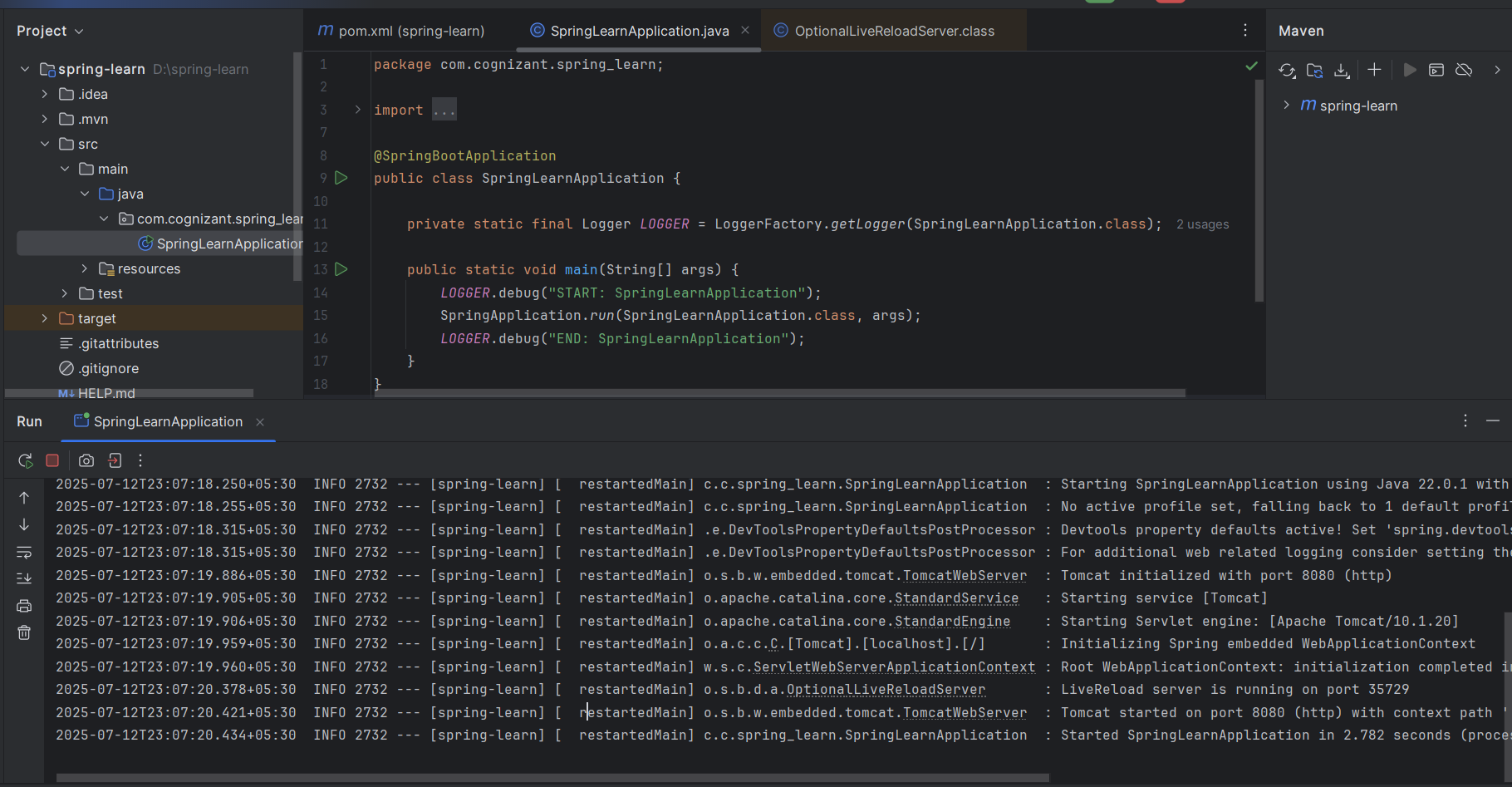
**SpringLearnApplication.java**

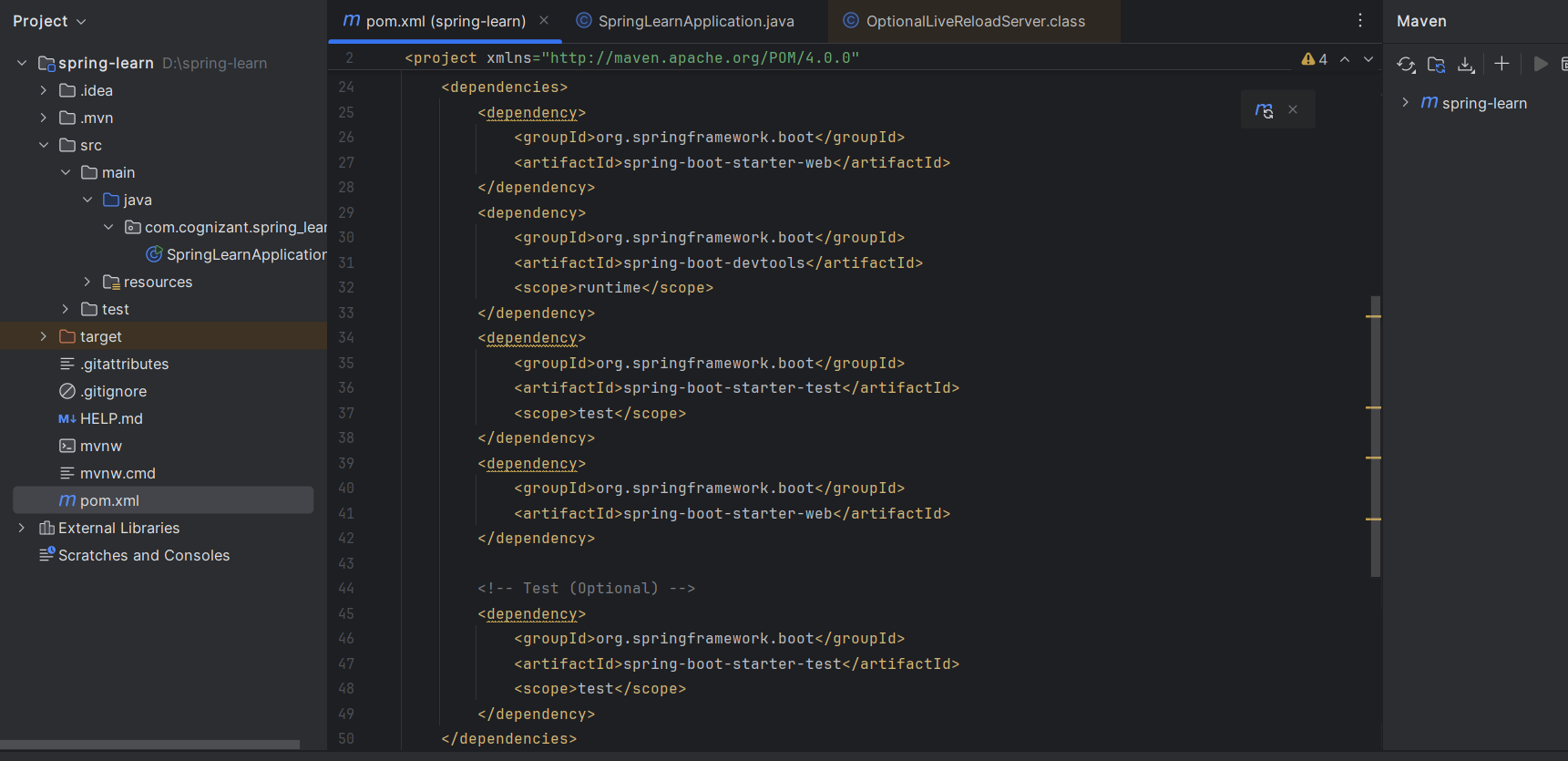
package com.cognizant.spring\_learn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 *LOGGER*.debug("START: SpringLearnApplication");  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 *LOGGER*.debug("END: SpringLearnApplication");  
 }  
}

**pom.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</name>  
 <description>Spring Learn Project</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.5</version>  
 <relativePath/>  
 </parent>  
  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Test (Optional) -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Output**

****

****

**SME WALKTHROUGH:**

**src/main/java - Application code**

* Contains:
  + com.cognizant.spring\_learn.SpringLearnApplication.java (entry point)
  + Any controllers, services, configurations you add in your project.

**src/main/resources - Application configuration**

* Contains:
  + application.properties for defining:
    - Server port
    - Logging level
    - DB configuration if needed
* Other resources like XML configurations or static files go here if needed.

**src/test/java - Testing code**

* Contains test classes for unit testing your application using JUnit.

**SpringLearnApplication.java - main() walkthrough**

* Entry point for Spring Boot.
* Calls SpringApplication.run(), bootstrapping the application:
  + Starts the embedded Tomcat server.
  + Scans for components (@Controller, @Service, etc.).
  + Initializes Spring context.

**Purpose of @SpringBootApplication**

It is equivalent to:

* @Configuration – marks as configuration class.
* @EnableAutoConfiguration – enables auto-configuration based on dependencies.
* @ComponentScan – scans the package and sub-packages for Spring components.

**Exercise 2: Spring Core – Load Country from Spring Configuration XML**

**Code**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
public class SpringLearnApplication {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(SpringLearnApplication.class);  
  
 public static void displayCountry() {  
 *LOGGER*.debug("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
  
 *LOGGER*.debug("Country : {}", country.toString());  
 *LOGGER*.debug("END");  
 }  
  
 public static void main(String[] args) {  
 *displayCountry*();  
 }  
}

**Country.java**

package com.cognizant.spring\_learn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class Country {  
 private String code;  
 private String name;  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(Country.class);  
  
 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 + "]";  
 }  
}

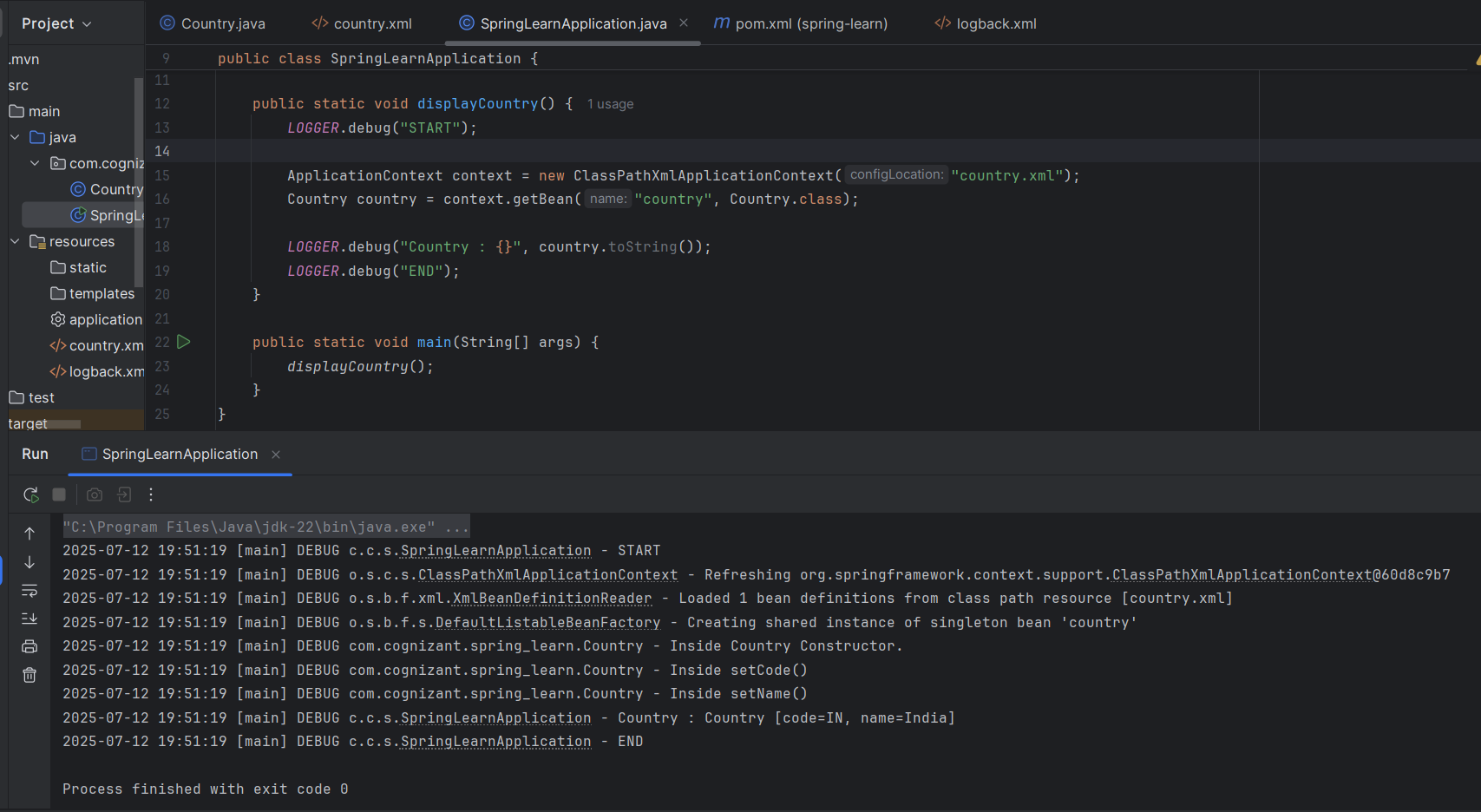
**country.xml**

<?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  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="IN" />  
 <property name="name" value="India" />  
 </bean>  
</beans>

**logback.xml**

<configuration>  
 <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">  
 <encoder>  
 <pattern>%d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level %logger{36} - %msg%n</pattern>  
 </encoder>  
 </appender>  
  
 <root level="DEBUG">  
 <appender-ref ref="STDOUT"/>  
 </root>  
</configuration>

**Output**



**SME Explanation:**

**bean tag, id, class, property, name, value:**

* bean tag: Defines a Spring-managed object.
* id: Unique name for the bean in XML.
* class: Fully qualified class name to instantiate.
* property: Used to set variable values in the bean.
* name: Name of the variable to set.
* value: Value to assign to the variable.

**ApplicationContext, ClassPathXmlApplicationContext:**

* ApplicationContext: Spring interface for managing and configuring beans.
* ClassPathXmlApplicationContext: Loads XML configuration from the classpath to initialize beans.

**What happens when context.getBean() is invoked:**

* Spring searches for the bean with the given id.
* It instantiates the class if not already created.
* It injects the property values defined in XML.
* It returns the fully configured object to your program for use.

**Exercise 3: Hello World RESTful Web Service**

**Code**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 }  
}

**HelloController.java**

package com.cognizant.spring\_learn;  
  
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*.debug("Start: sayHello()");  
 String message = "Hello World!!";  
 *LOGGER*.debug("End: sayHello()");  
 return message;  
 }  
}

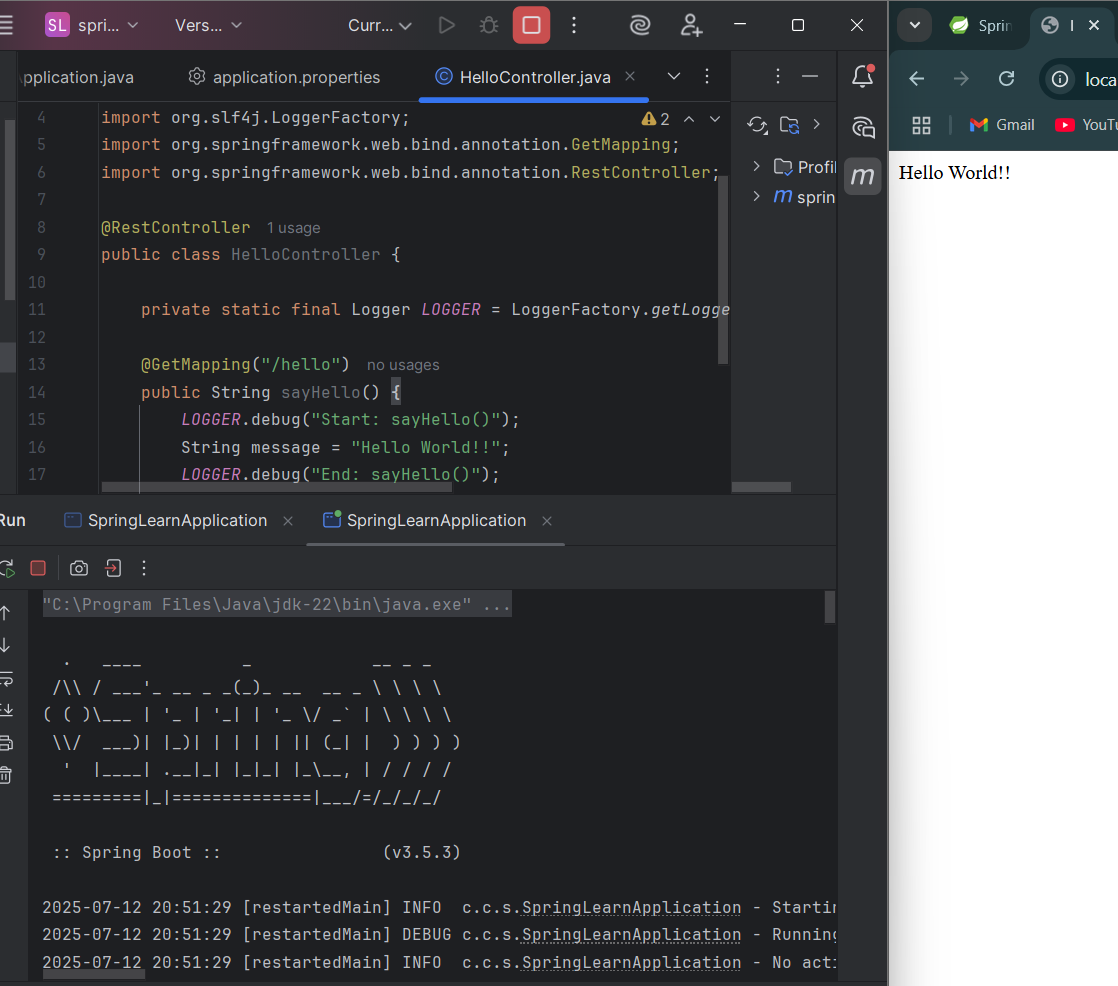
**application.properties**

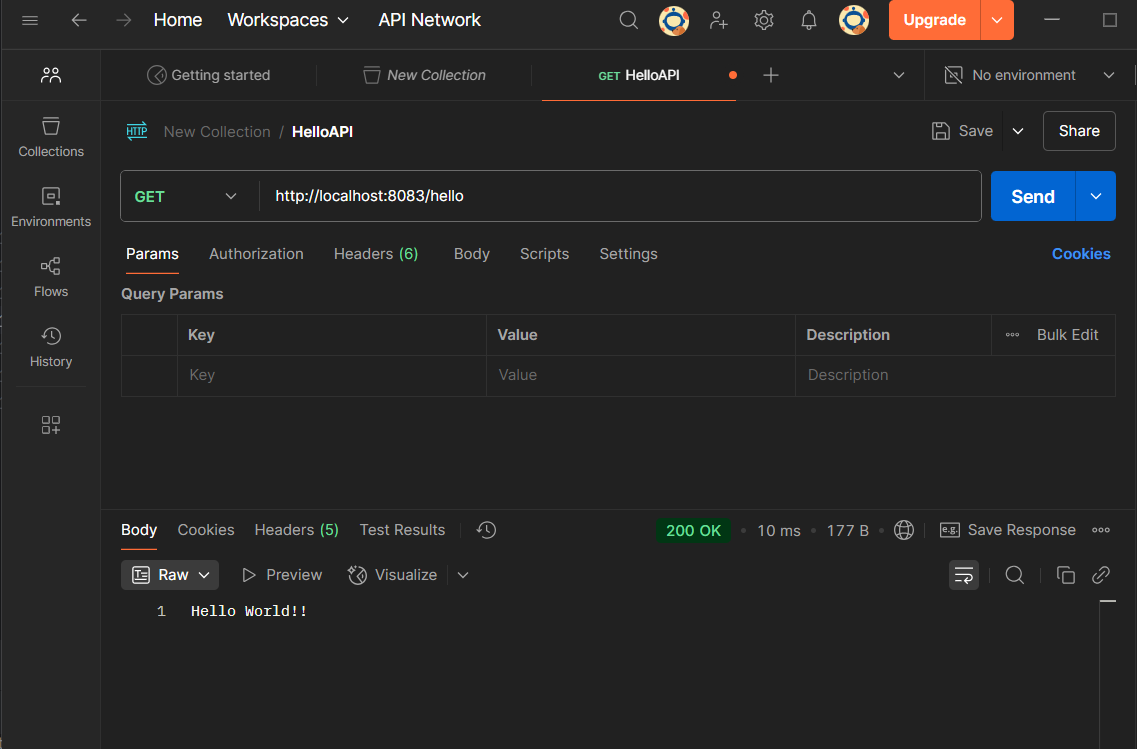
spring.application.name=spring-learn  
server.port=8083

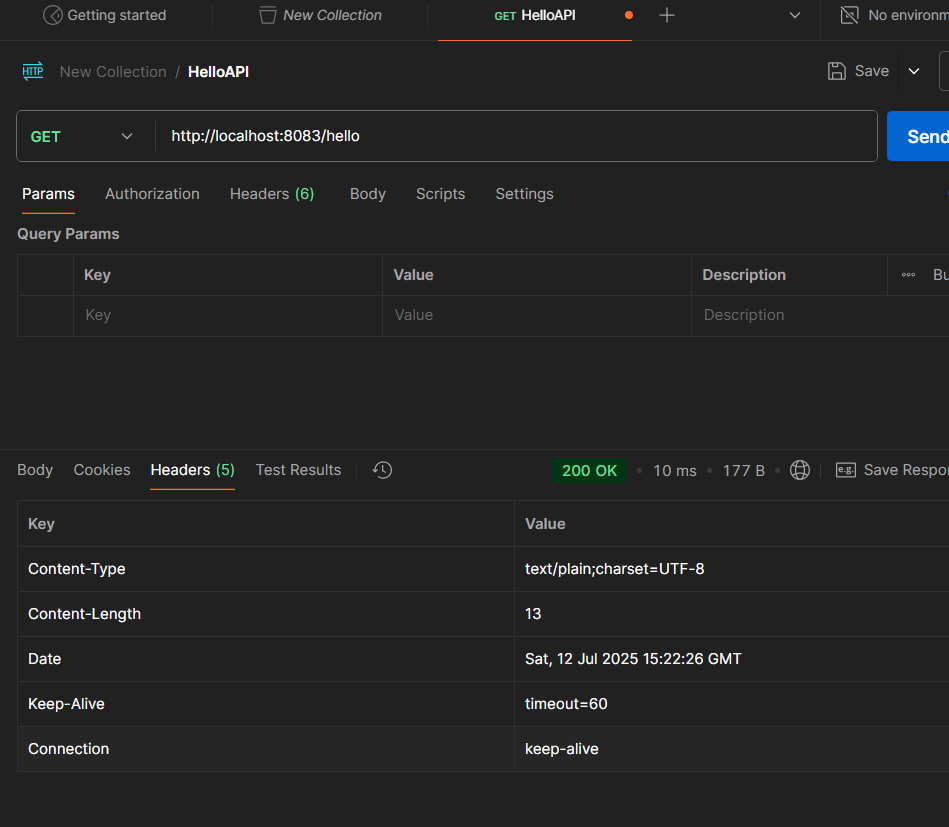
**pom.xml:**

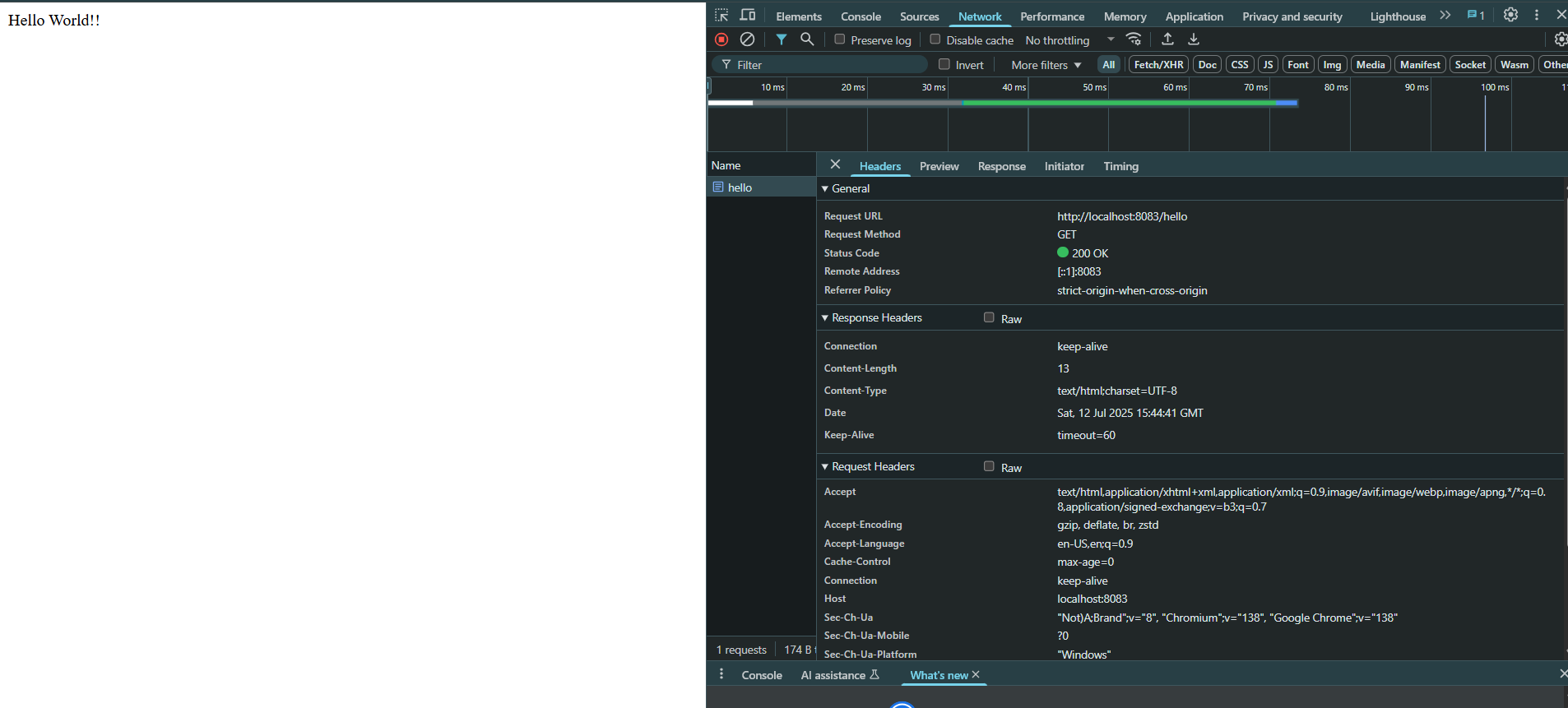
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.5.3</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</name>  
 <description>Demo project for Spring Boot</description>  
  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Web Starter for REST APIs -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Spring Boot DevTools for hot reload during development -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-api</artifactId>  
 <version>2.0.13</version>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Output**

****

****

****



**Exercise 4: REST - Country Web Service**

**Code**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 }  
}

**Country.java**

package com.cognizant.spring\_learn;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
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 + "]";  
 }  
}

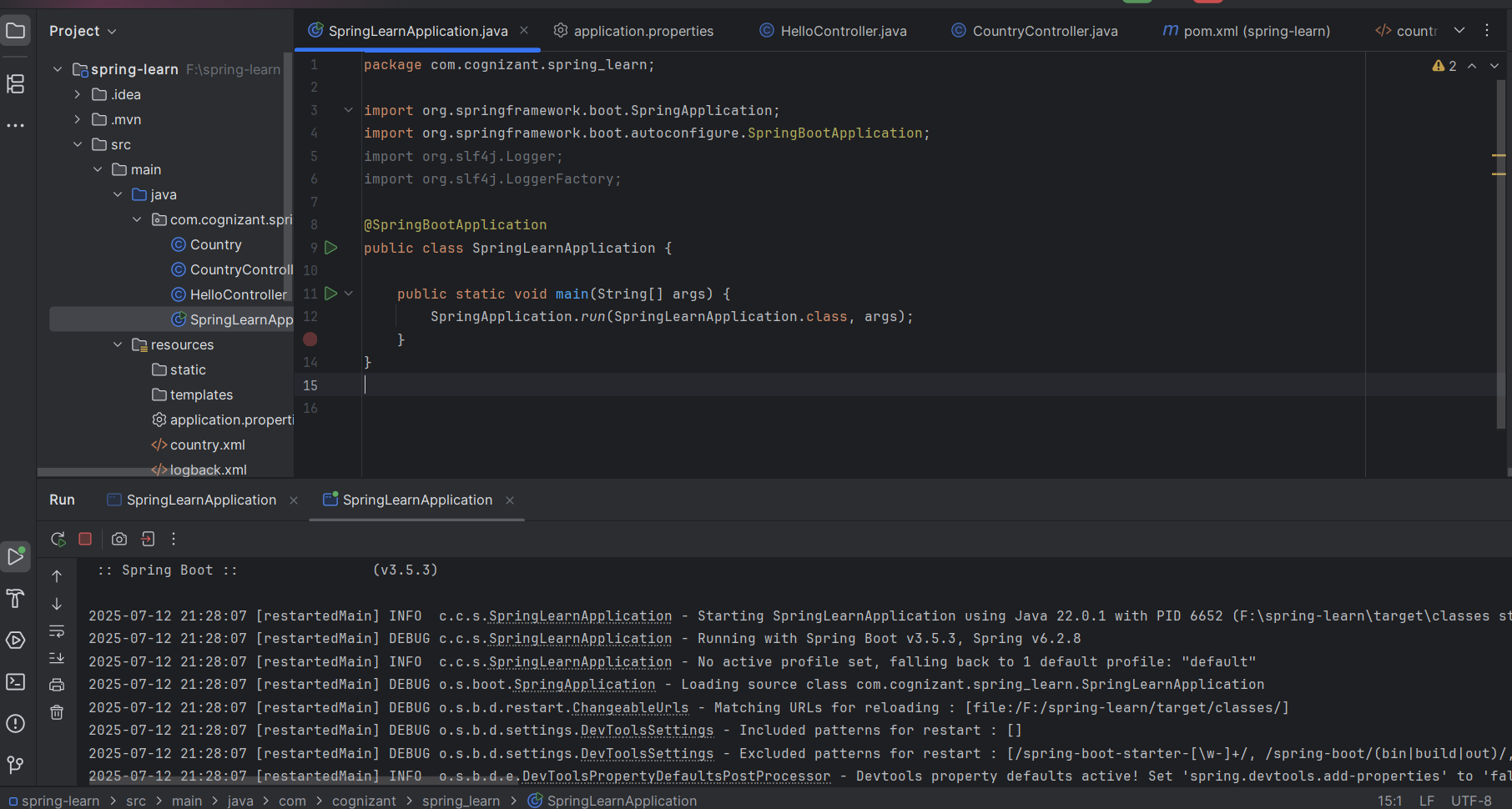
**CountryController.java**

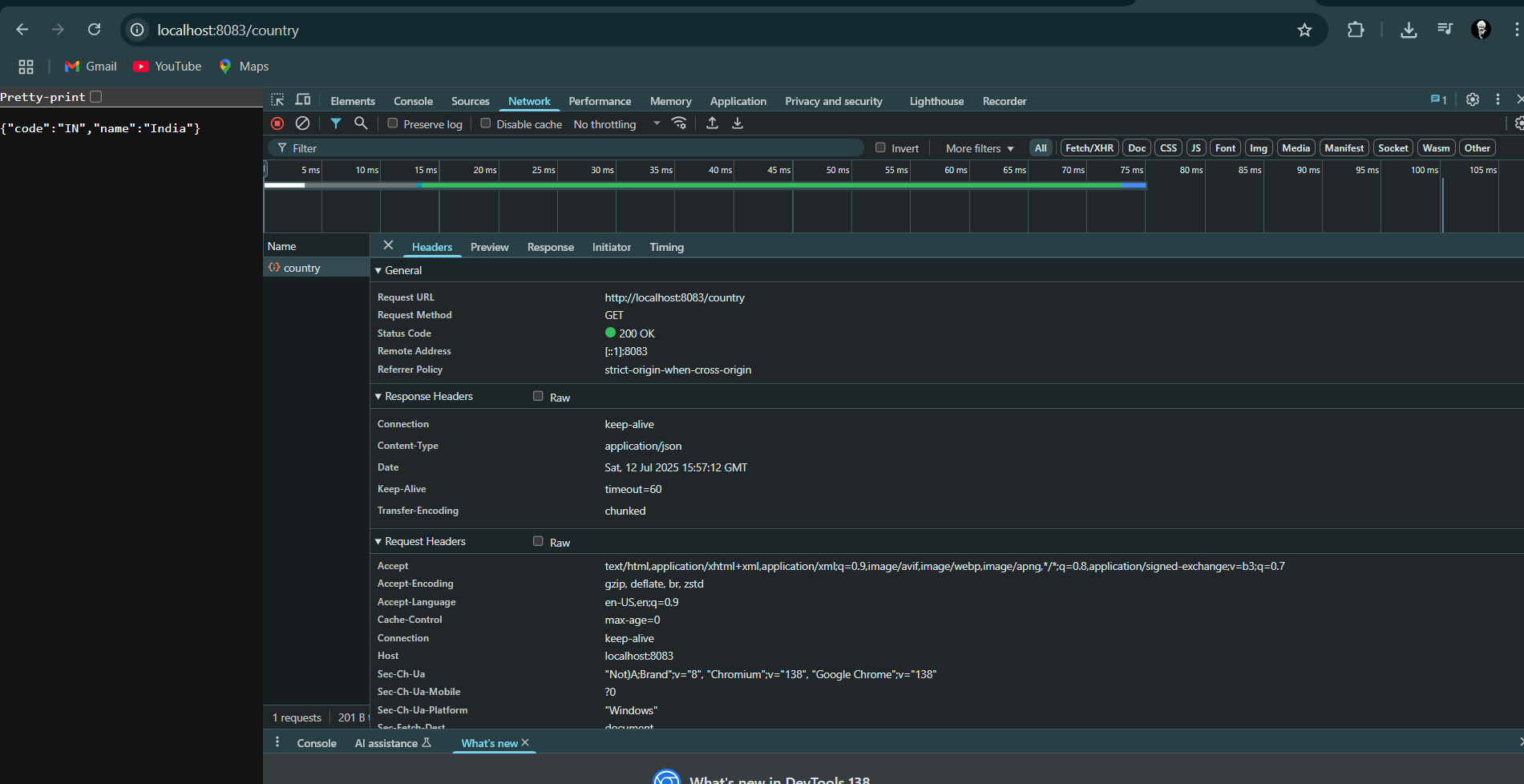
package com.cognizant.spring\_learn;  
  
import com.cognizant.spring\_learn.Country;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class CountryController {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryController.class);  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 *LOGGER*.debug("START");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country", Country.class);  
 *LOGGER*.debug("Country: {}", country);  
 *LOGGER*.debug("END");  
 return country;  
 }  
}

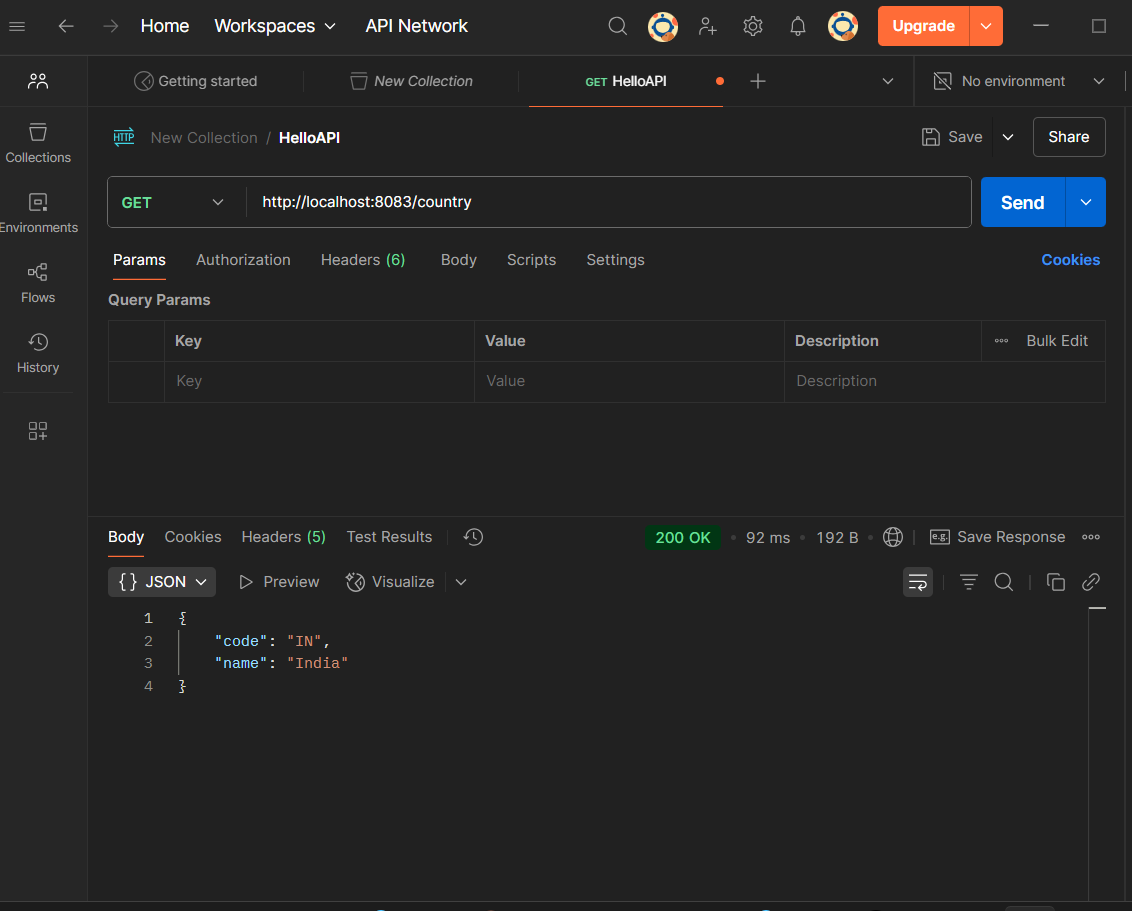
**country.xml**

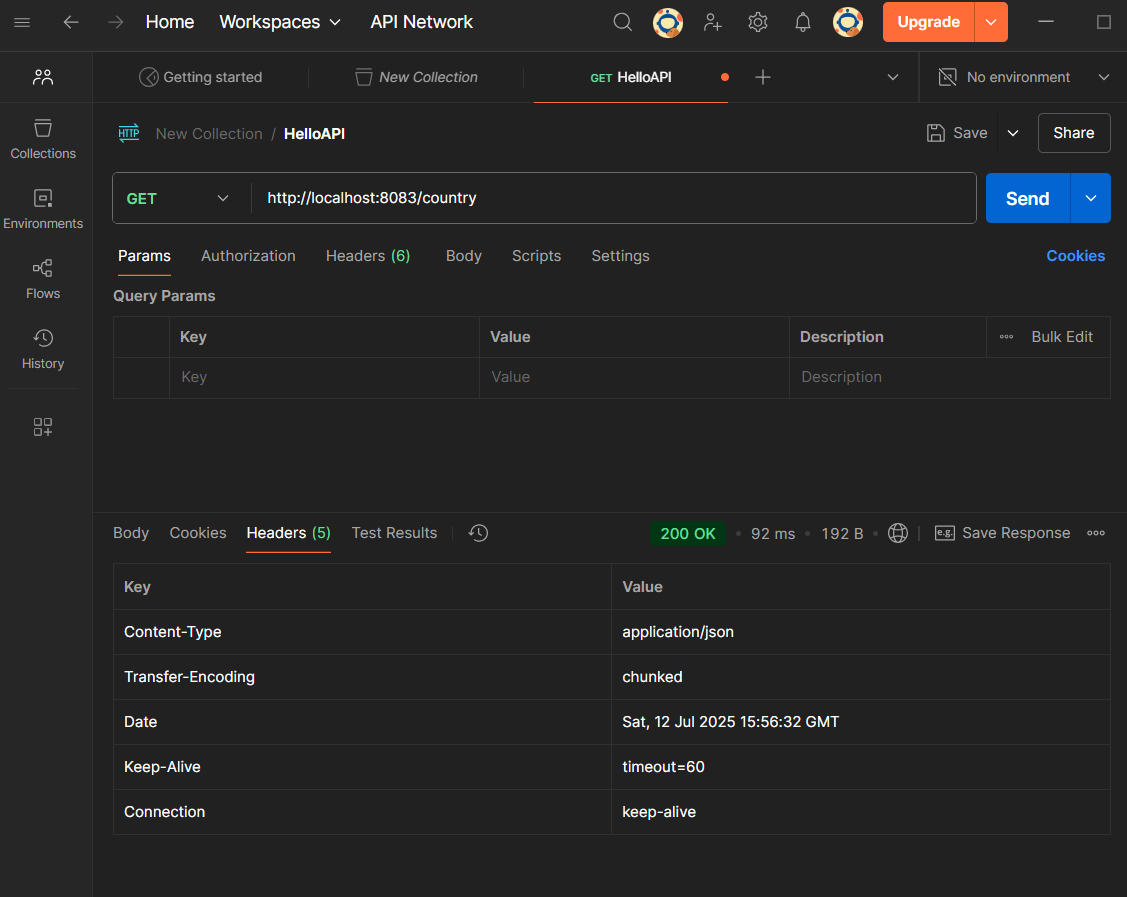
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="IN" />  
 <property name="name" value="India" />  
 </bean>  
  
</beans>

**Output**

****

****

****

****

**SME Explanation**

**What happens in the controller method?**

* When /country is hit via a GET request:
  + Spring Boot routes the request to getCountryIndia().
  + The method loads the country.xml Spring configuration file using ClassPathXmlApplicationContext.
  + It retrieves the country bean (with code "IN" and name "India").
  + It returns this Country object.

**How is the bean converted into JSON response?**

* Spring Boot (via spring-boot-starter-web) includes Jackson for automatic JSON serialization.
* The returned Country object from getCountryIndia() is automatically converted into JSON and returned in the HTTP response

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

**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  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <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="countryJP" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
  
 <bean id="countryDE" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="countryIN"/>  
 <ref bean="countryUS"/>  
 <ref bean="countryJP"/>  
 <ref bean="countryDE"/>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 }  
}

**CountryService.java**

package com.cognizant.spring\_learn.service;  
  
import com.cognizant.spring\_learn.Country;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Service;  
import java.util.List;  
  
@Service  
public class CountryService {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryService.class);  
  
 public Country getCountry(String code) {  
 *LOGGER*.debug("START: getCountry");  
 try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml")) {  
 List<Country> countries = context.getBean("countryList", List.class);  
  
 return countries.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*.debug("START: getCountry with code: {}", code);  
 Country country = countryService.getCountry(code);  
 *LOGGER*.debug("END: getCountry with result: {}", country);  
 return country;  
 }  
}

**output:**

