**Cognizant Digital Nurture 4.0**

**Name: Siri Chandana Chittipolu**

**Email:** [**sirichittipolu11@gmail.com**](mailto:sirichittipolu11@gmail.com)

**Superset ID: 6386277**

**Mandatory Hands-On Exercises**

**Spring REST using Spring Boot 3**

**Task 1: Hands on 1: Create a Spring Web Project using Maven**

**Solution:**

**Step 1: Creating project from Spring Initializer**

1. Go to: <https://start.spring.io/>
2. Entering:
   * Group: com.cognizant
   * Artifact: spring-learn
3. Adding dependencies:
   * Spring Boot DevTools
   * Spring Web
4. Clicking on Generate to download the ZIP file.

**Step 2: Extract project**

* Extracting the downloaded ZIP into my Eclipse workspace folder.

**Step 3: Import project into Eclipse**

1. In Eclipse, go to:  
   File → Import → Maven → Existing Maven Projects → Next
2. Click Browse, select the extracted folder.
3. Click Finish.

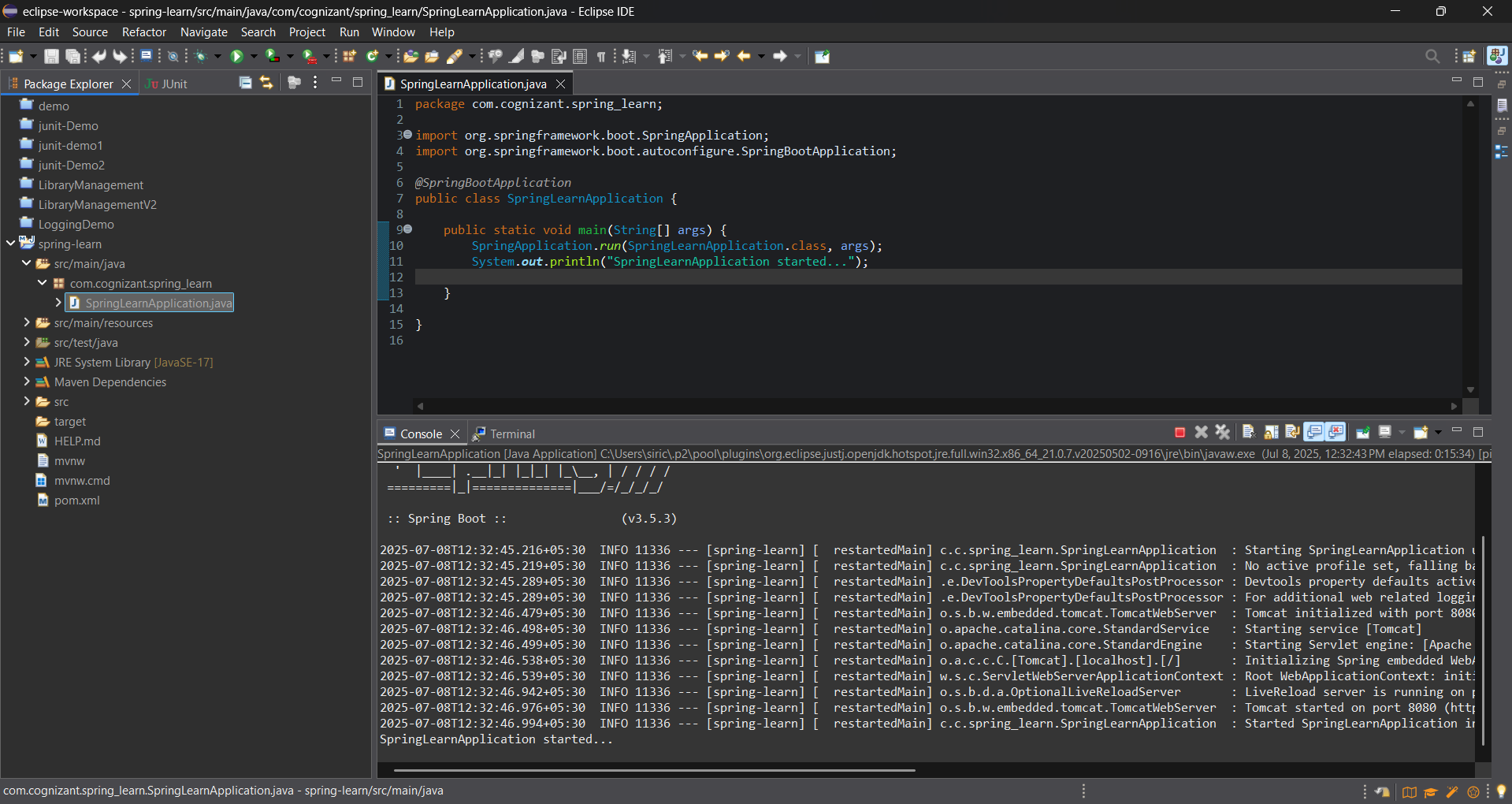
**Step 5: Verify main() method and add log**

* Open src/main/java/com/cognizant/springlearn/SpringLearnApplication.java.
* Adding this line to the above directory: System.out.println("SpringLearnApplication started...");

**Step 6: Running the application**

* Right-click SpringLearnApplication.java → Run As → Java Application.
* Observe the console output.

**Output:**

****

**Task 2: Hands on 4: Spring Core – Load Country from Spring Configuration XML**

**Solution:**

**Step 1:** Created a Maven project named spring-learn with group id com.cognizant.

**Step 2: Added dependencies in pom.xml:**

* spring-boot-starter-web
* spring-boot-starter-logging (for debug logs)
* spring-context
* spring-boot-devtools

**Step 3: Created a Country bean in country.xml:**

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

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

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

</bean>

**Step 4: Added application.properties:**

logging.level.com.cognizant.spring\_learn=DEBUG

logging.level.root=DEBUG

**Step 5: Created Country class:**

* With fields code and name
* Default constructor with debug log & println
* Setter and getter methods with debug logs
* toString() method

**Step 6: In SpringLearnApplication:**

* Defined method displayCountry() to load the bean:

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

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

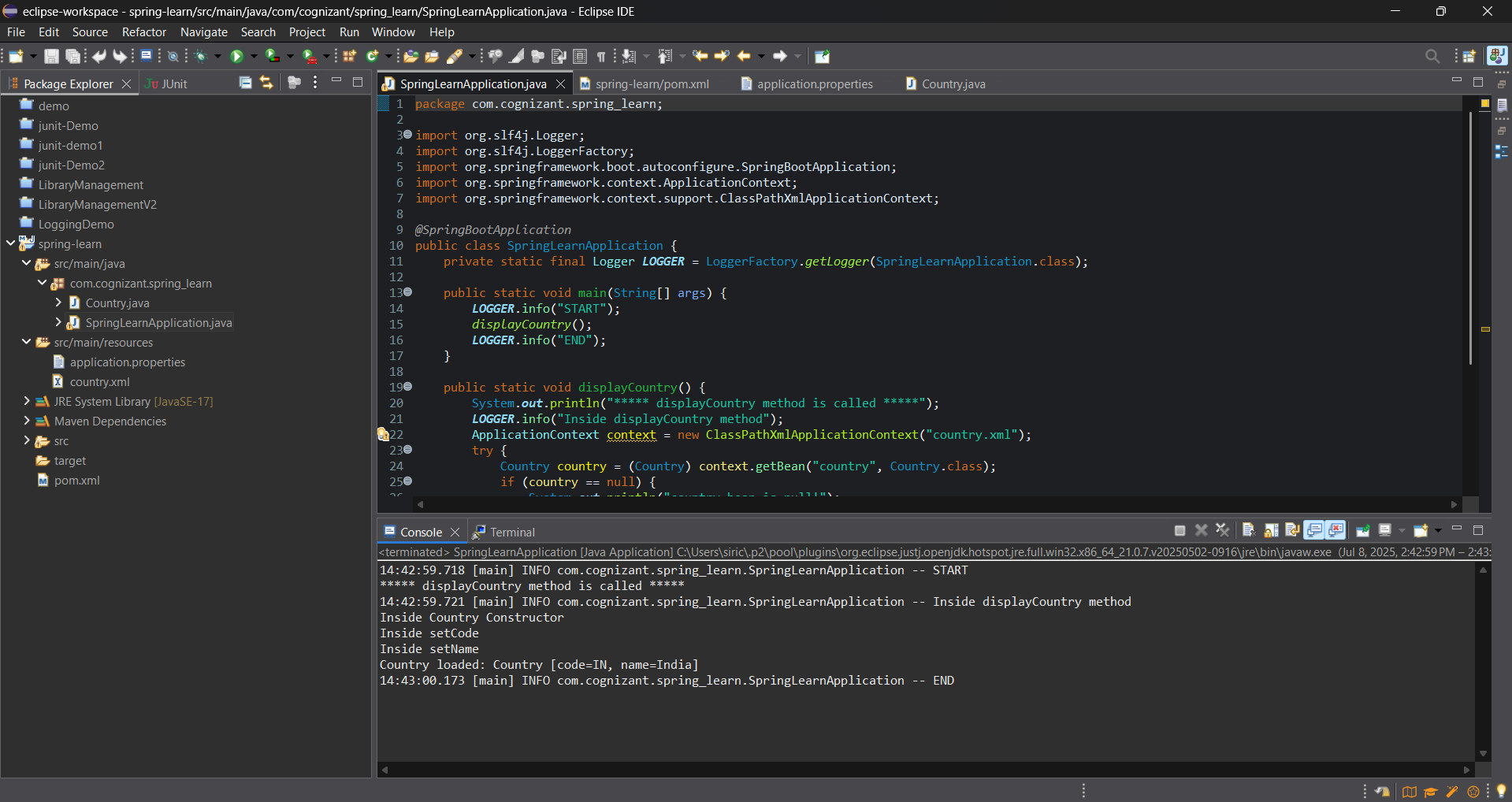
LOGGER.debug("Country : {}", country.toString());

System.out.println("Country loaded: " + country);

* Called displayCountry() inside main() method.

**Step 7:** Ran the application and checked the logs.

**Output:**

****

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

**Solution:**

**Step 1:Created a REST controller**

* Added HelloController inside com.cognizant.spring\_learn.controller.
* Used @RestController and added method public String sayHello() mapped to GET /hello.
* Method returns hard-coded string "Hello World!!" and includes start & end log messages.

HelloController.java:

package com.cognizant.helloworldservice.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 response = "Hello World!!";

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

return response;

}

}

**Step 2: Configured port**

* Added server.port=8085 in application.properties.

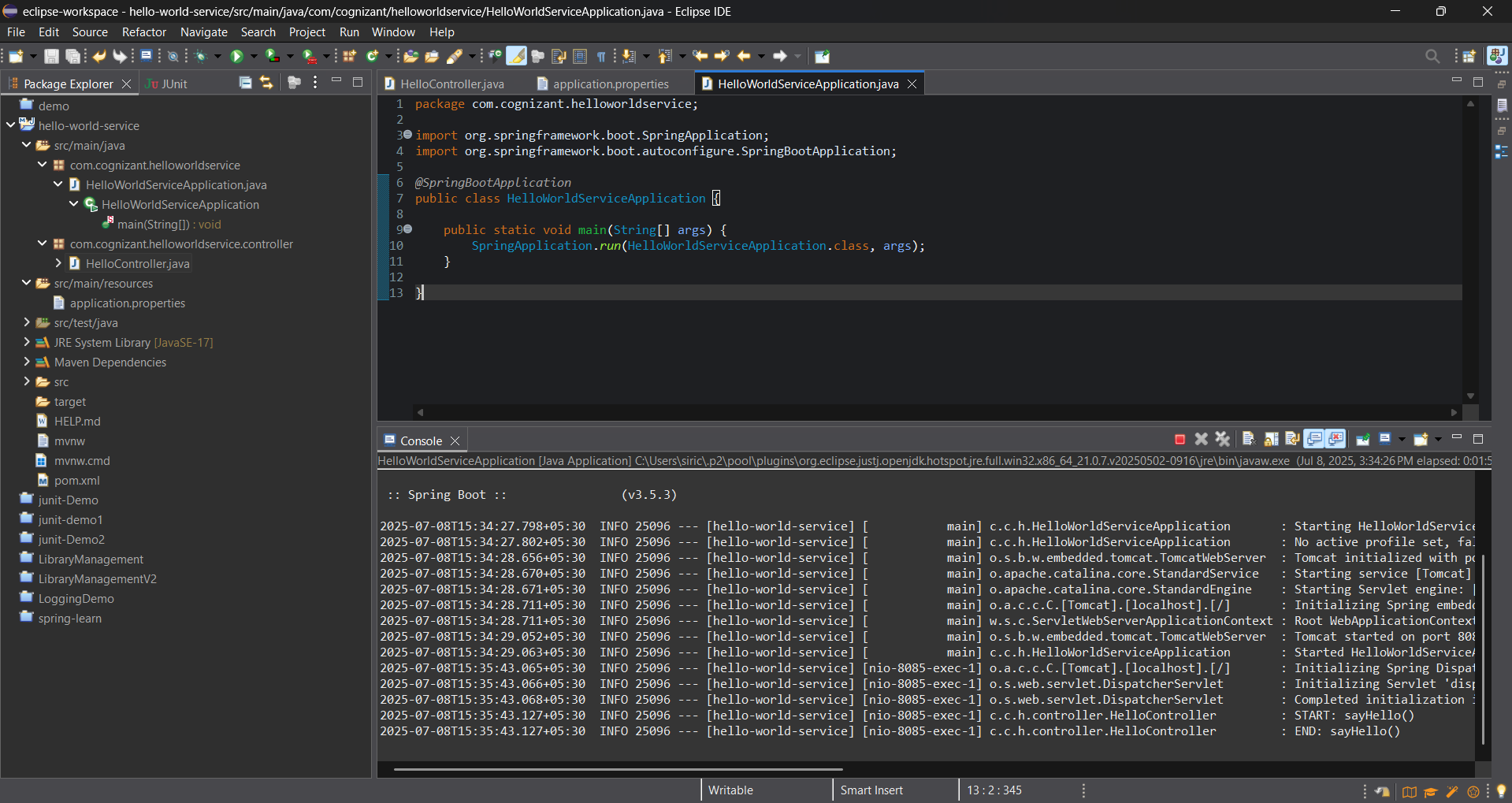
**Step 3: Ran and tested**

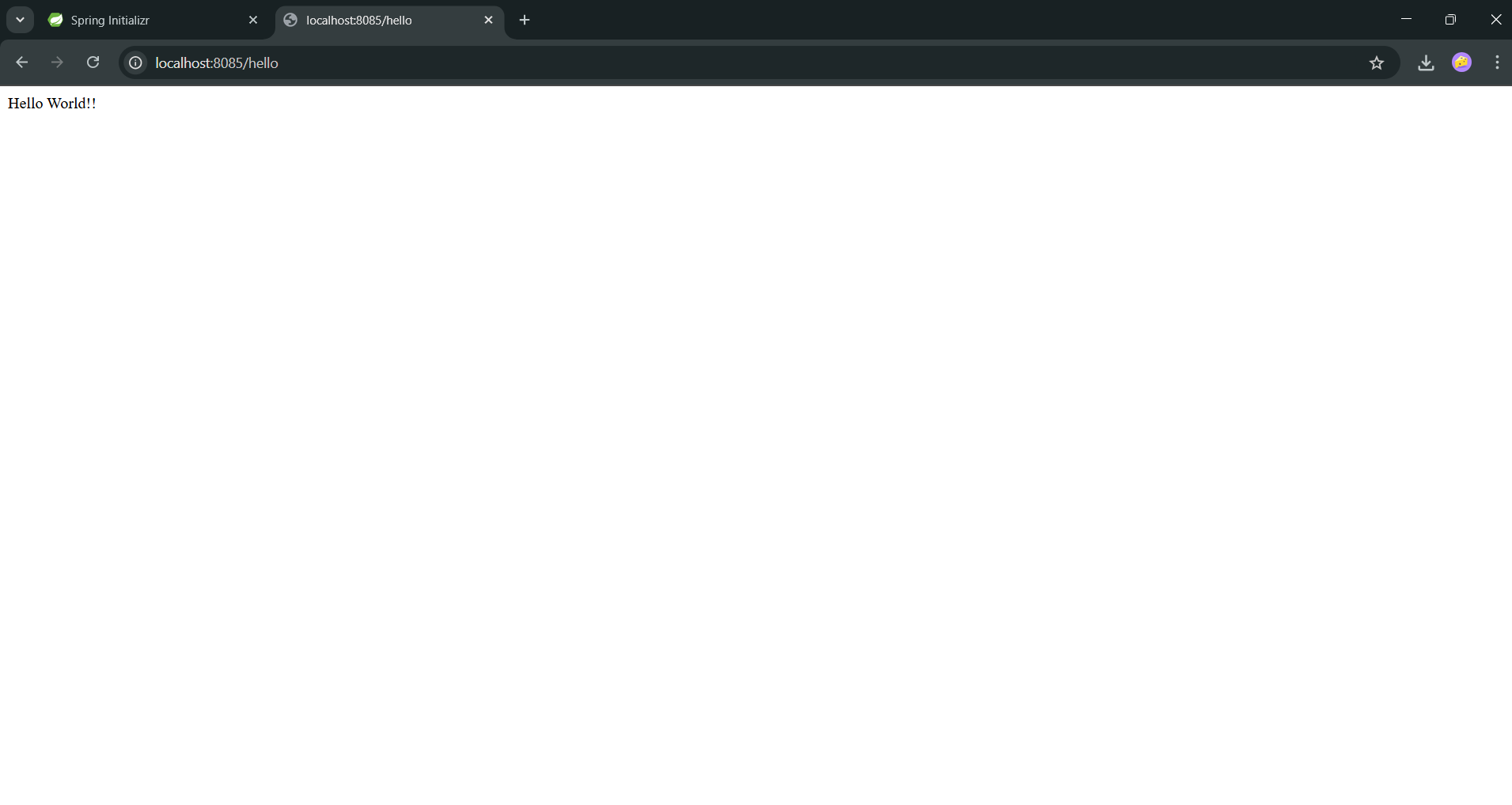
* Opened http://localhost:8083/hello in Chrome → saw Hello World!!.
* Sent GET request to same URL in Postman → got Hello World!! response.

**Step 4: Checked HTTP headers**

* In browser: Developer Tools → Network → clicked on request → viewed Headers tab.
* In Postman: clicked “Headers” tab under the response to view received headers.

**Output:**

****

****

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

**Solution:**

**Step 1: Create the REST controller**

In the earlier created spring-learn application, I added a controller:

com.cognizant.spring\_learn.controller.CountryController

Inside it, I wrote the method:

@RequestMapping("/country")

public Country getCountryIndia() {

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

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

return country;

}

This method loads the India bean from spring.xml and returns it.

**Step 2: Add the Country bean in spring.xml**

I configured the bean like this:

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

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

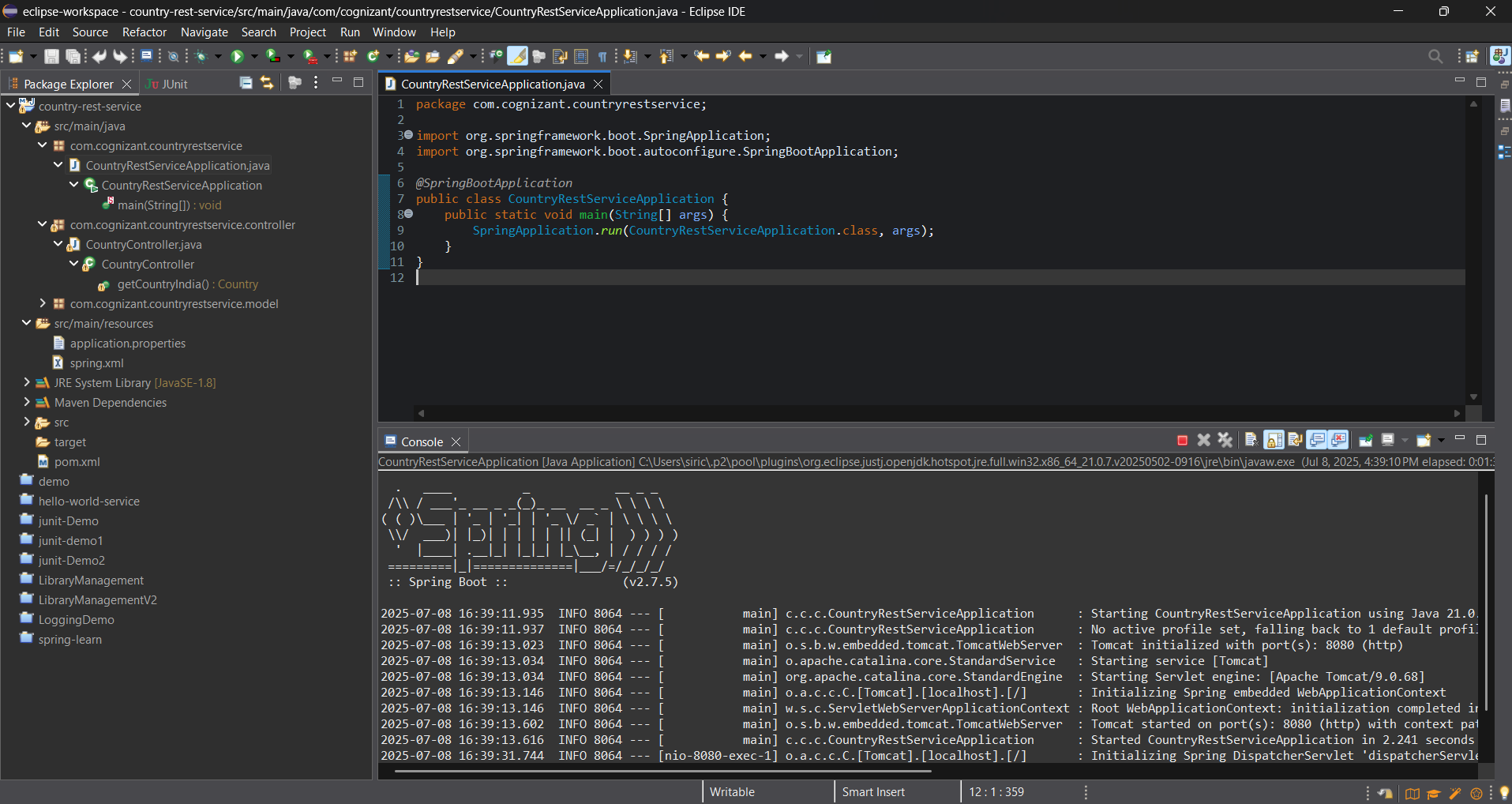
<property name="name" value="India"/>

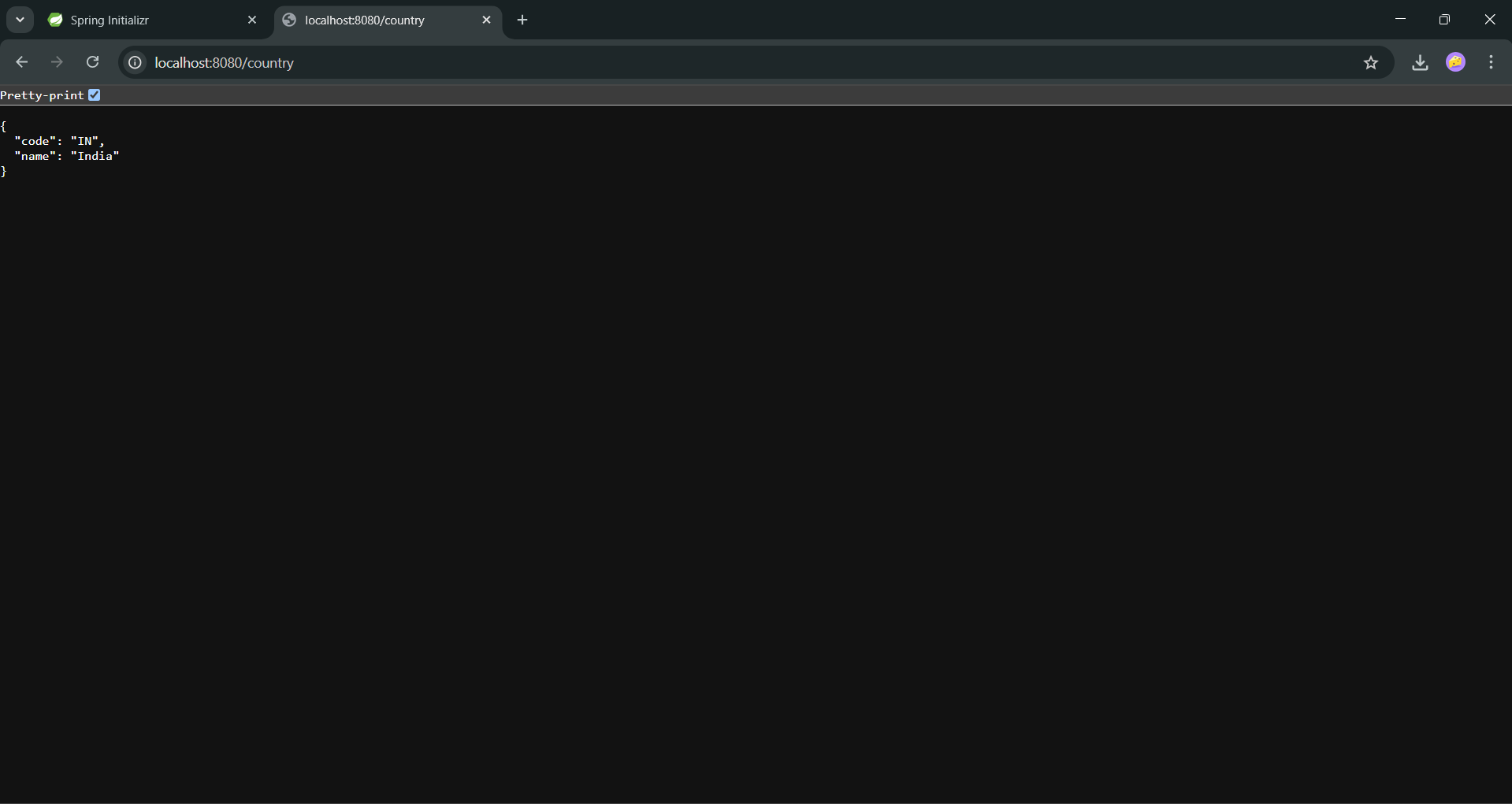
</bean>

**Step 3: Testing the API**

* URL: http://localhost:8083/country

**Output:**

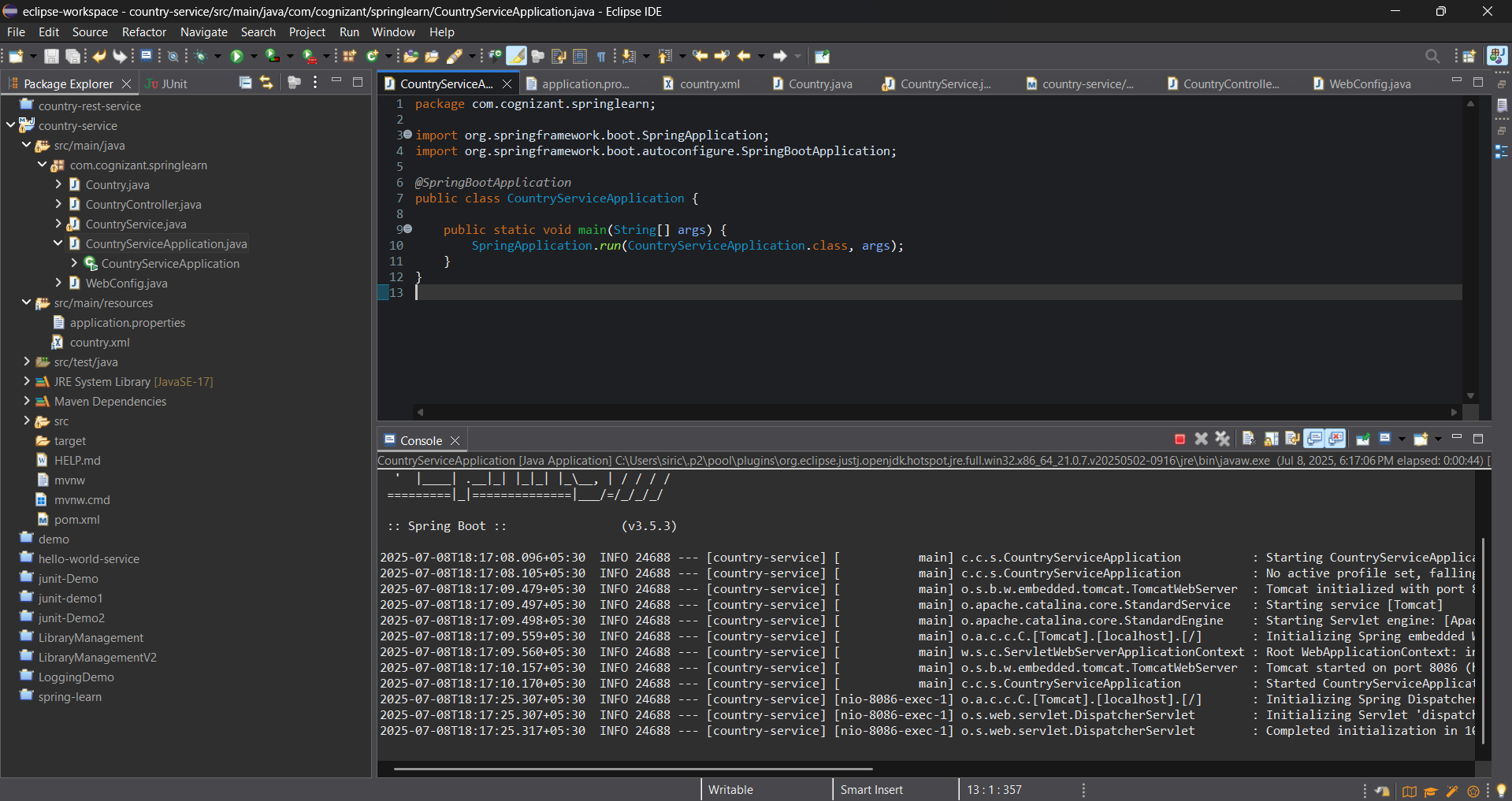
****

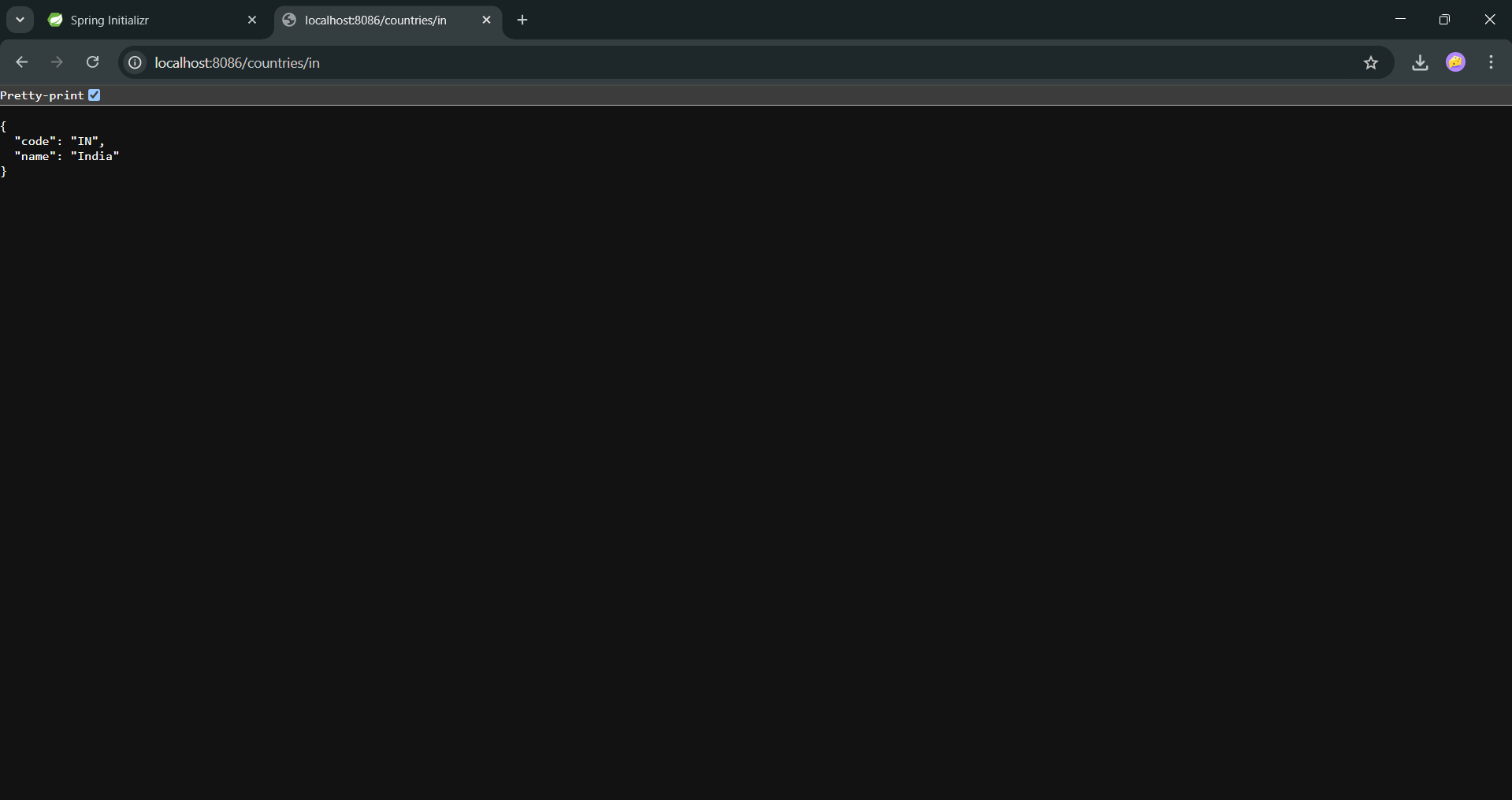
****

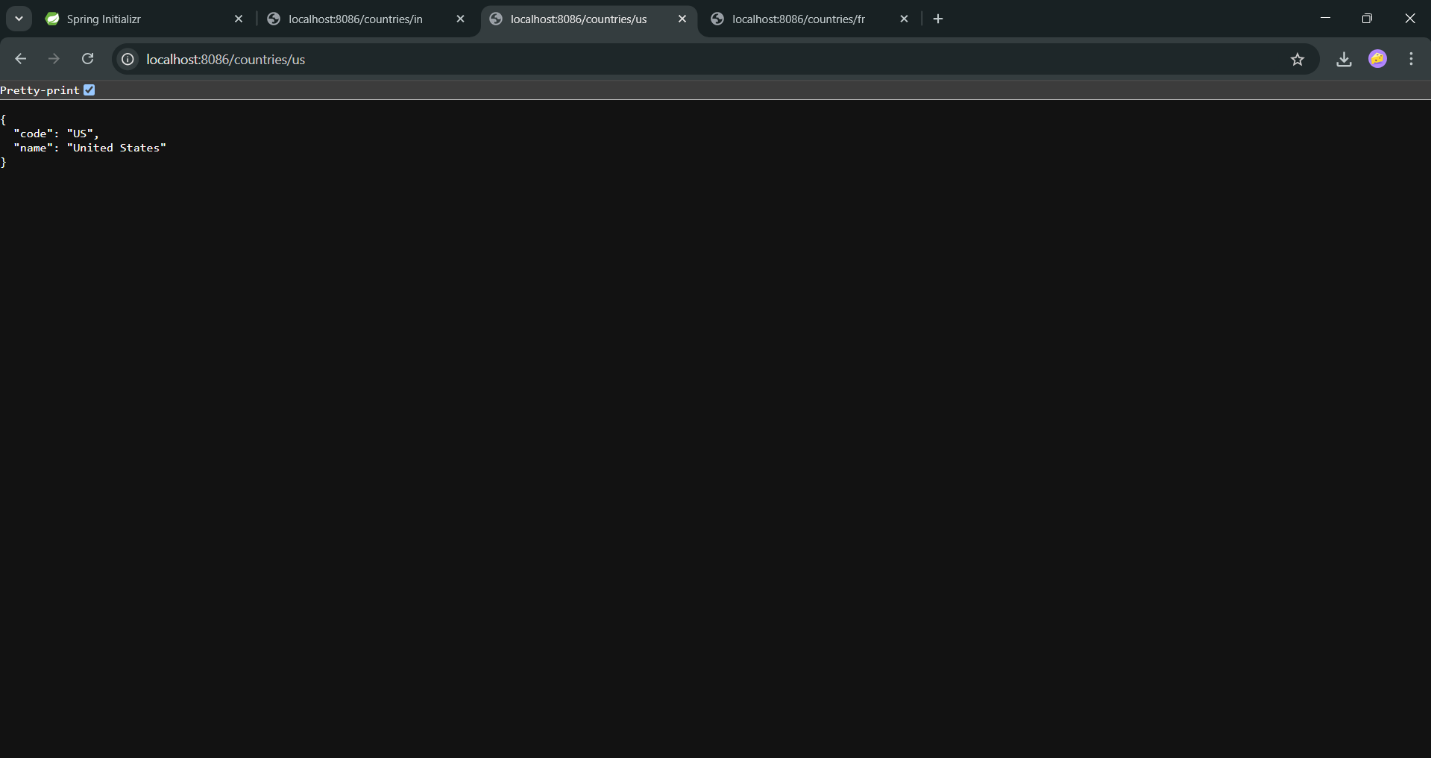
**Task 5: REST - Get country based on country code**

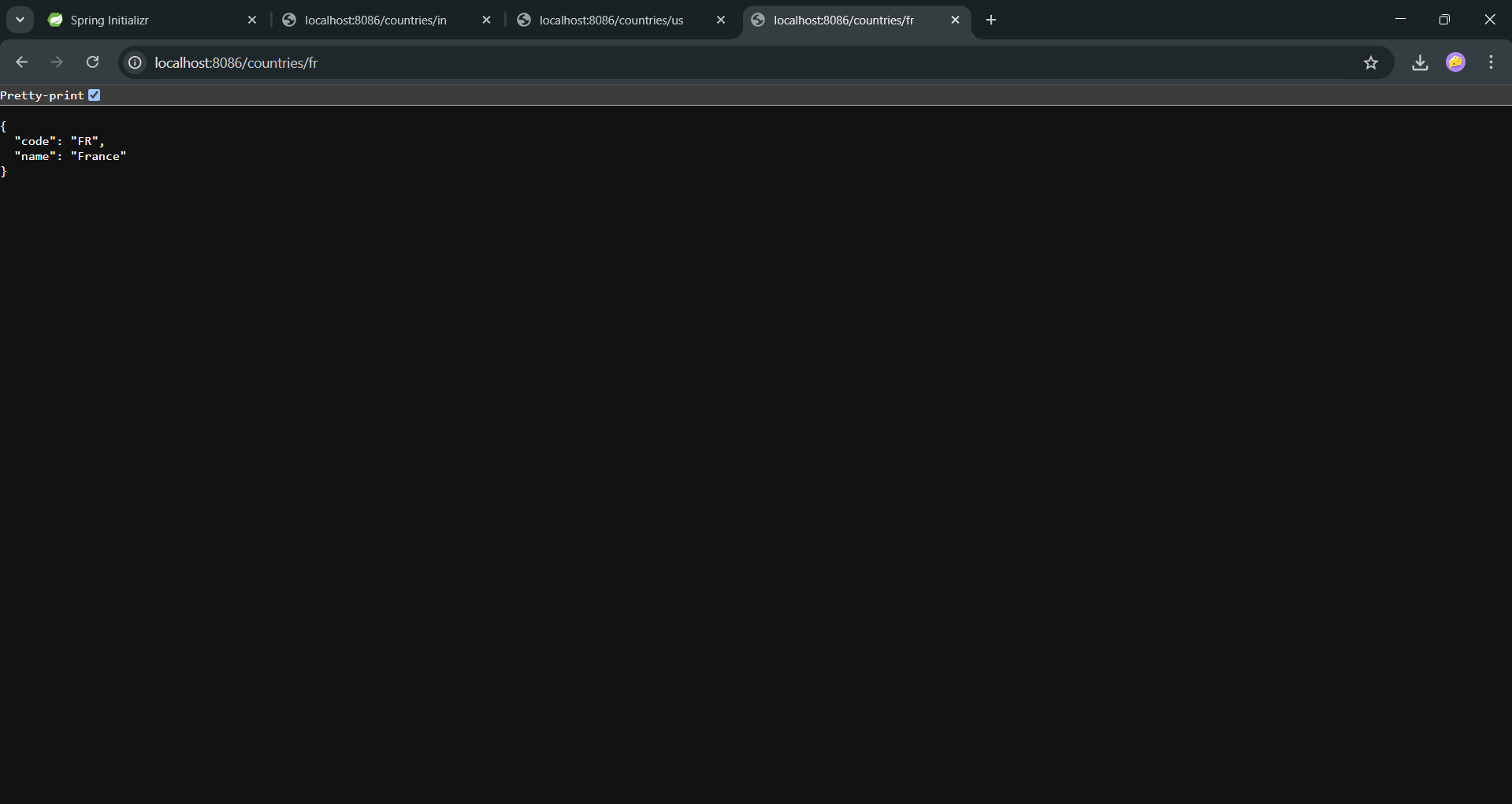
**Solution:**

1. Created a Spring Boot application with a REST controller to handle country code input using @PathVariable.
2. Defined the getCountry(String code) method in the controller, which calls the service method to fetch the country.
3. Loaded the list of countries from country.xml file using XmlMapper inside the service class.
4. Used a stream and lambda expression to iterate through the list and perform a case-insensitive match on the country code.
5. Returned the matched country object as a JSON response.

****

****

****

****