**WEEK – 4**

**SPRING REST USING SPRING BOOT 3**

**CREATE A SPRING WEB PROJECT USING MAVEN**

**SME Walkthrough**

**src/main/java:**

* The main class SpringLearnApplication.java is inside the package: com.cognizant.springlearn
* The src/main/java folder contains the main source code of the Spring Boot application, including the entry point class and all functional components.

**src/main/resources:**

* application.properties – to define settings like server port, logging level, database URL, etc.
* The src/main/resources folder is used to store configuration and property files required for application setup.

**src/test/java:**

* It contains package : com.cognizant.springlearn
* Uses JUnit testing libraries
* The src/test/java folder contains test cases to validate the application logic and ensure code reliability.

**SpringLearnApplication.java:**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

System.*out*.println("SpringLearnApplication main method started...");

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

}

}

**SpringLearnApplicationTests.java:**

package com.cognizant.spring\_learn;

import org.junit.jupiter.api.Test;

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

@SpringBootTest

class SpringLearnApplicationTests {

@Test

void contextLoads() {

}

}

**Dependencies on pom.xml:**

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

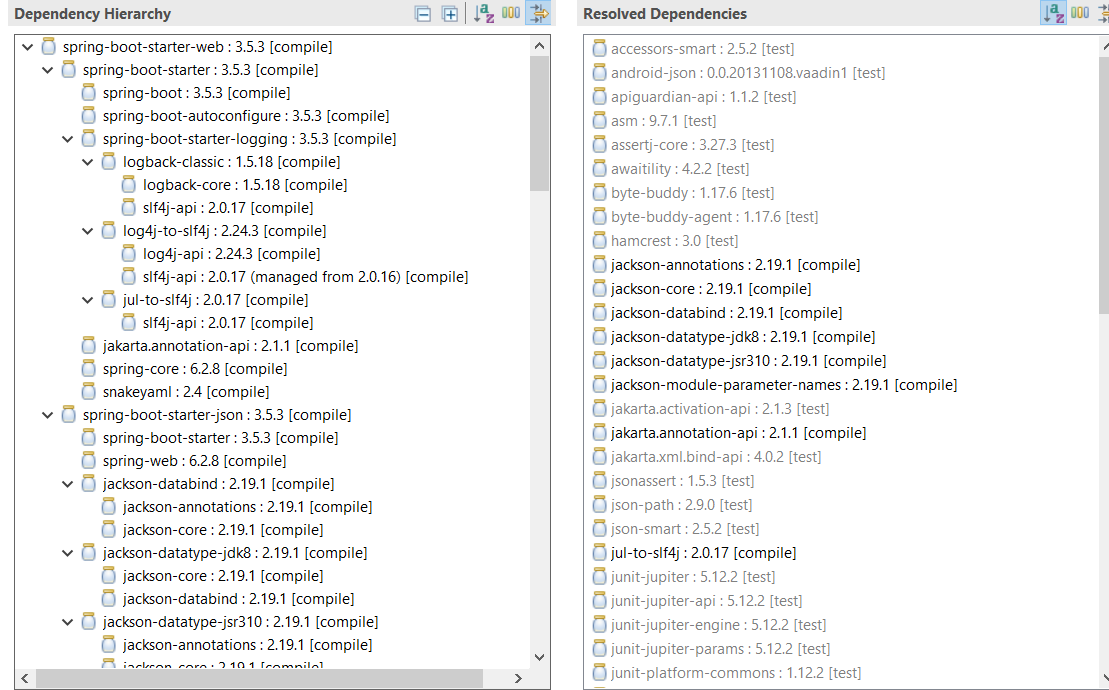
<artifactId>spring-boot-devtools</artifactId>

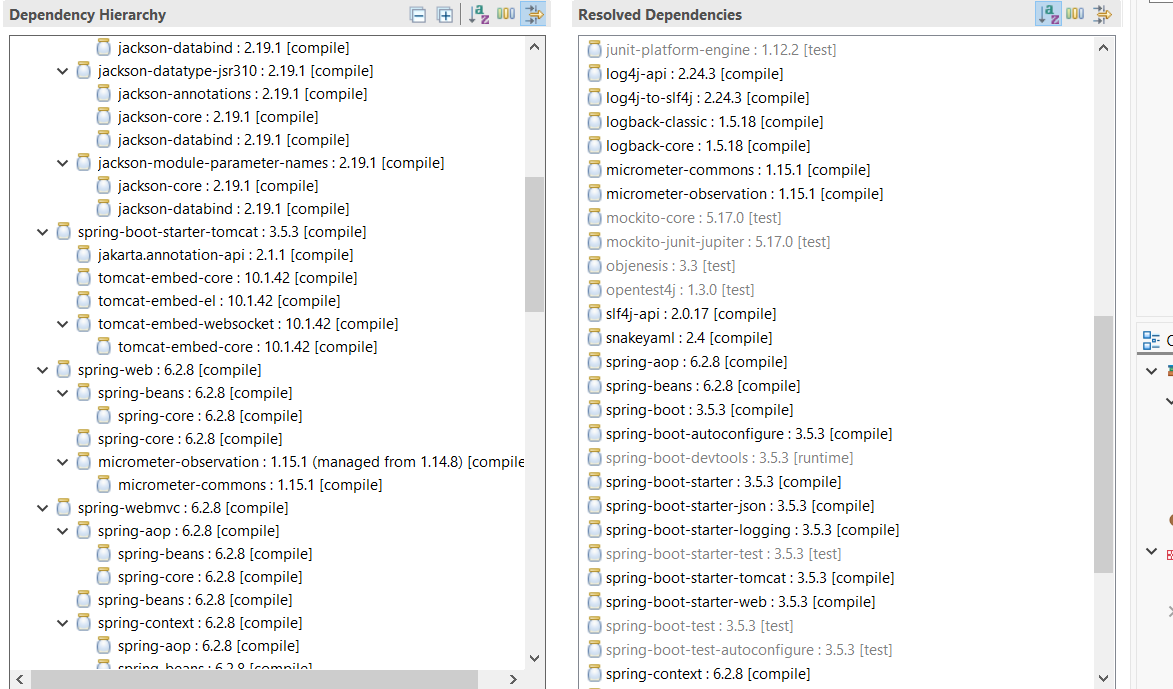
<scope>runtime</scope>

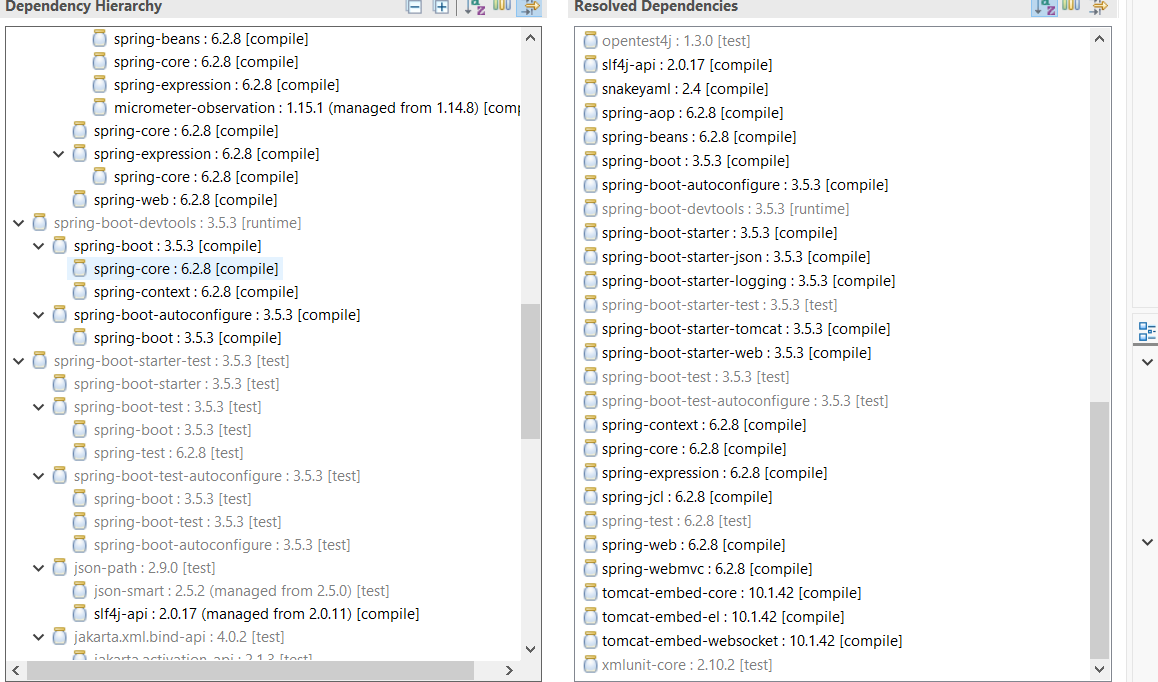
</dependency>

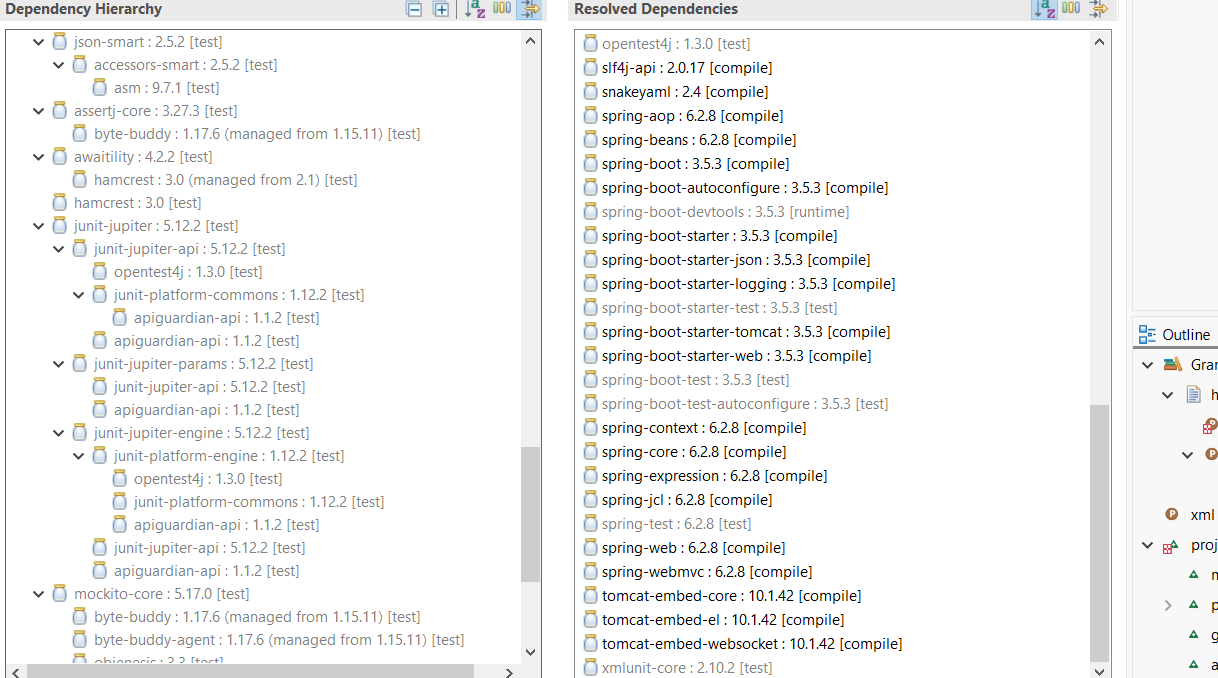
</dependencies>

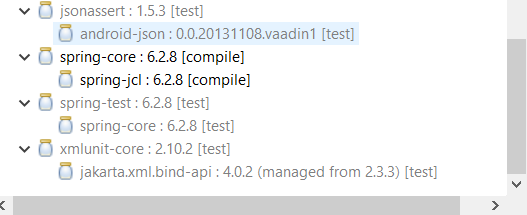
**DEPENDENCY HIERARCHY**



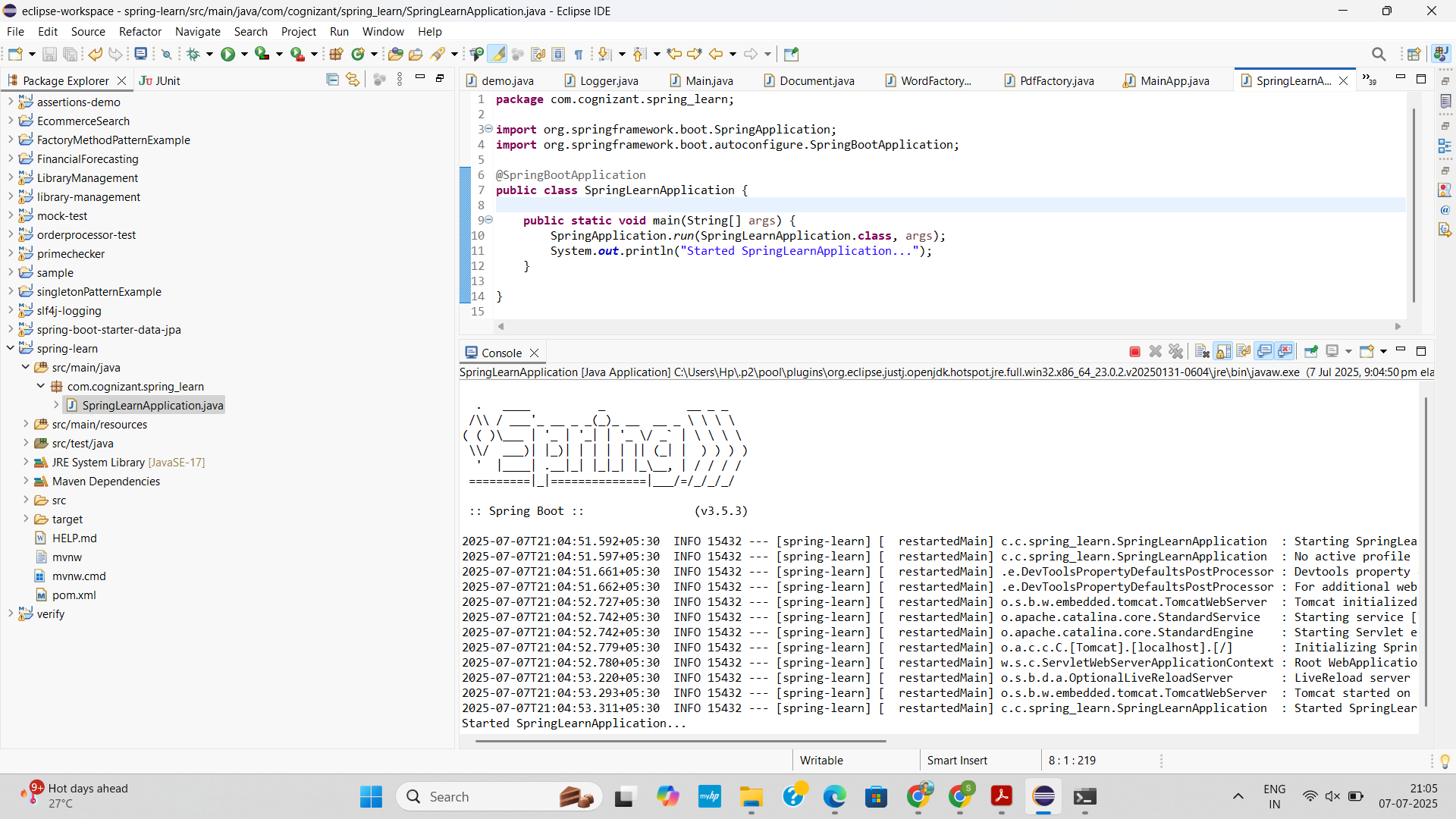








**OUTPUT**

****

**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 + '\'' +

'}';

}

}

**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;

import org.springframework.context.ApplicationContext;

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) {

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

*displayCountry*();

}

public static void displayCountry() {

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

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

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

}

}

**application.properties**

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

server.port=9090

**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>

**SpringLearnApplicationTests.java**

package com.cognizant.spring\_learn;

import org.junit.jupiter.api.Test;

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

@SpringBootTest

class SpringLearnApplicationTests {

@Test

void contextLoads() {

}

}

**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> <!-- latest GA as of Jul 2025 -->

<relativePath/> <!-- look up in Maven Central -->

</parent>

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-learn</name>

<description>Learning project for Spring Boot Web</description>

<packaging>jar</packaging>

<properties>

<java.version>17</java.version> <!-- Boot 3.x requires ≥17 -->

<!-- <native.image.enabled>true</native.image.enabled> # GraalVM, optional -->

</properties>

<dependencies>

<!-- Web (Spring MVC + embedded Tomcat) -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</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>

- spring-boot-starter-data-jpa

- mysql‑connector‑j

- spring-boot-starter-security

- etc. -->

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<!-- No extra config needed; Boot 3 handles layers & aot automatically -->

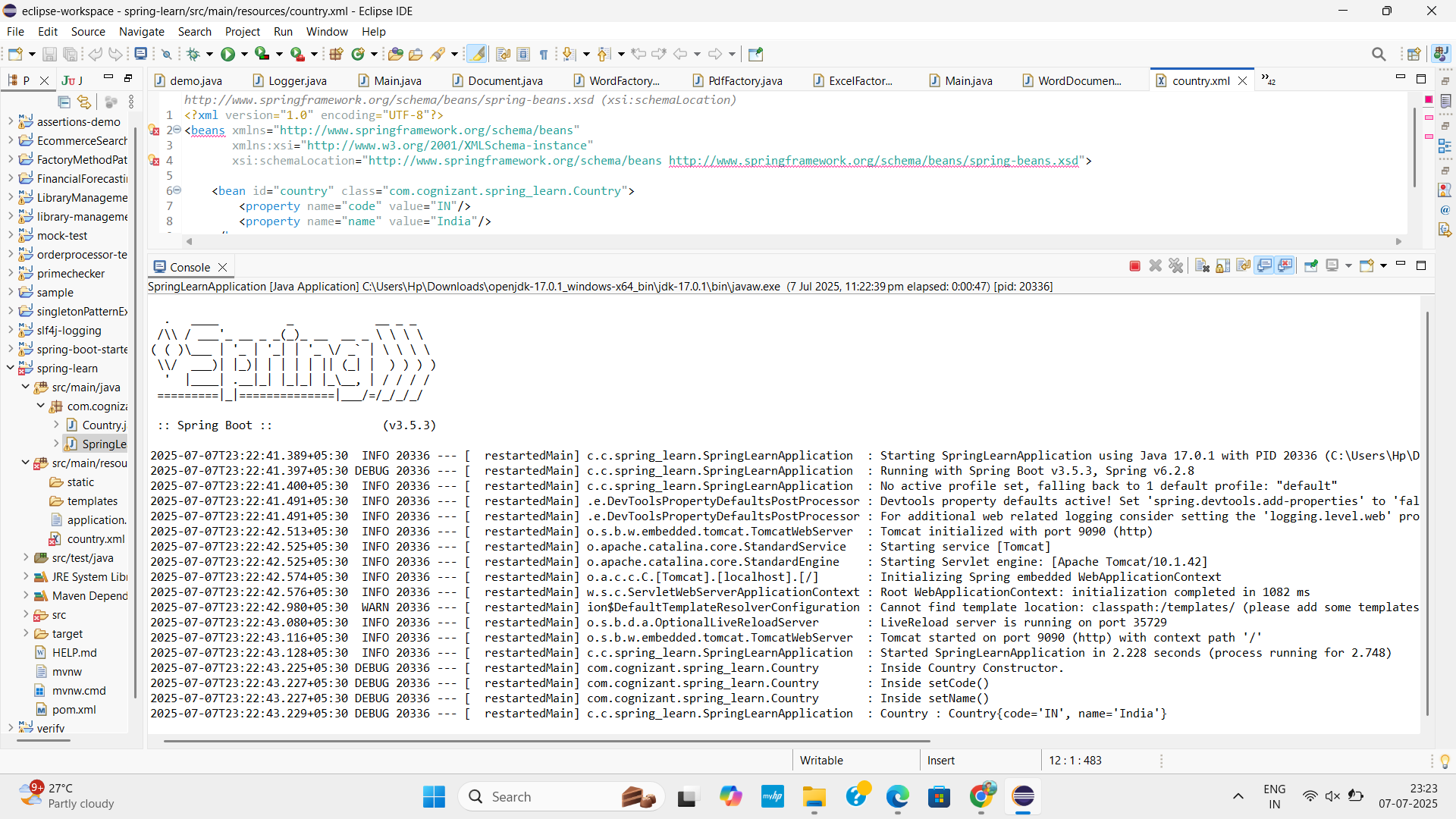
</plugin>

</plugins>

</build>

</project>

**OUTPUT**

****

**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!!

**APPROACH**

* Configure Port in application.properties with server.port=8083.
* Create HelloController in com.cognizant.spring\_learn.controller with @RestController.
* Define sayHello() method using @GetMapping("/hello") to return "Hello World!!".
* Add Logging with Logger to log start and end of the method.
* Run and Test via browser/Postman at http://localhost:8083/hello and check headers/logs.

**CODE**

**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()");

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

return "Hello World!!";

}

}

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

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

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

}

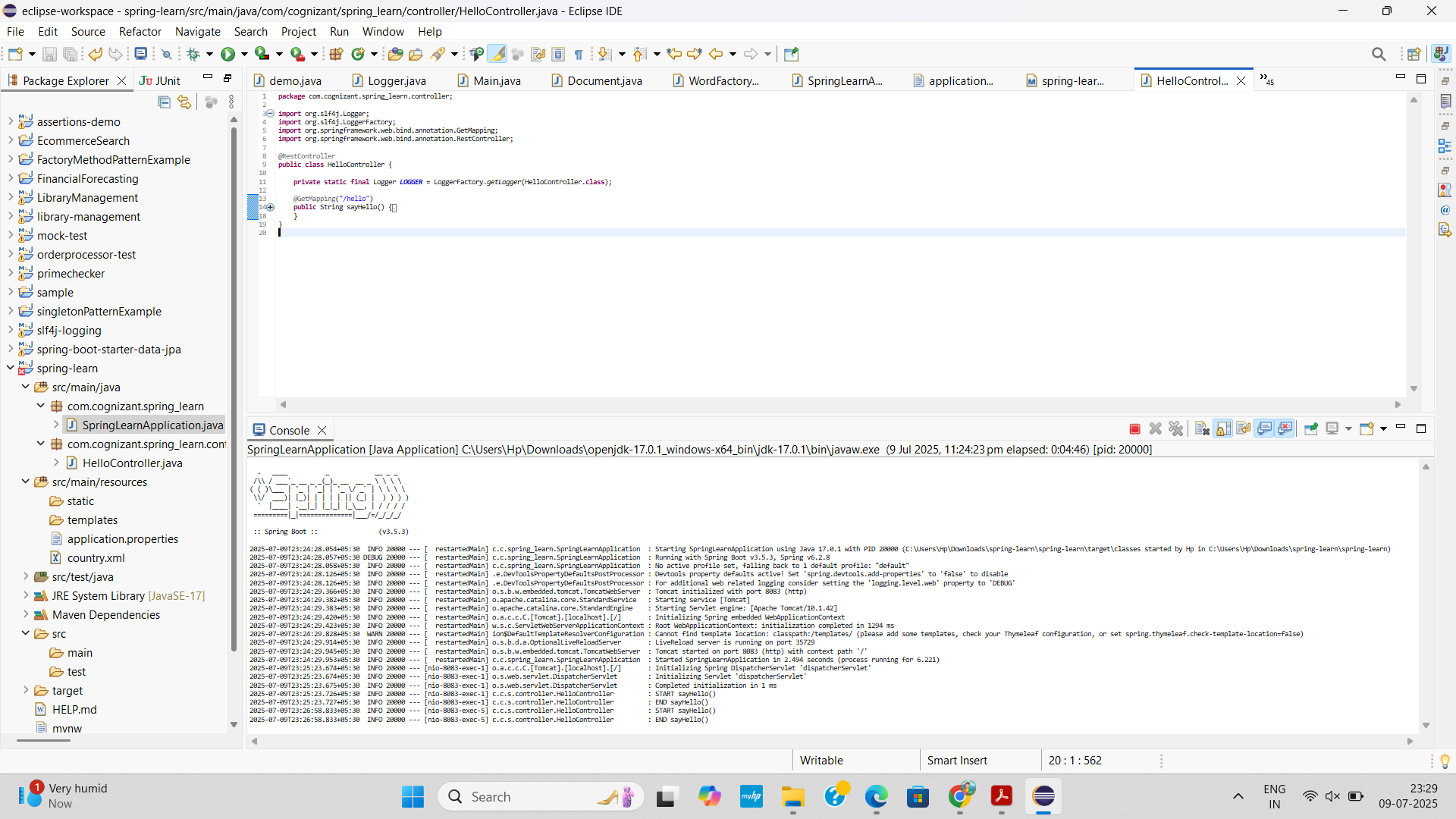
}

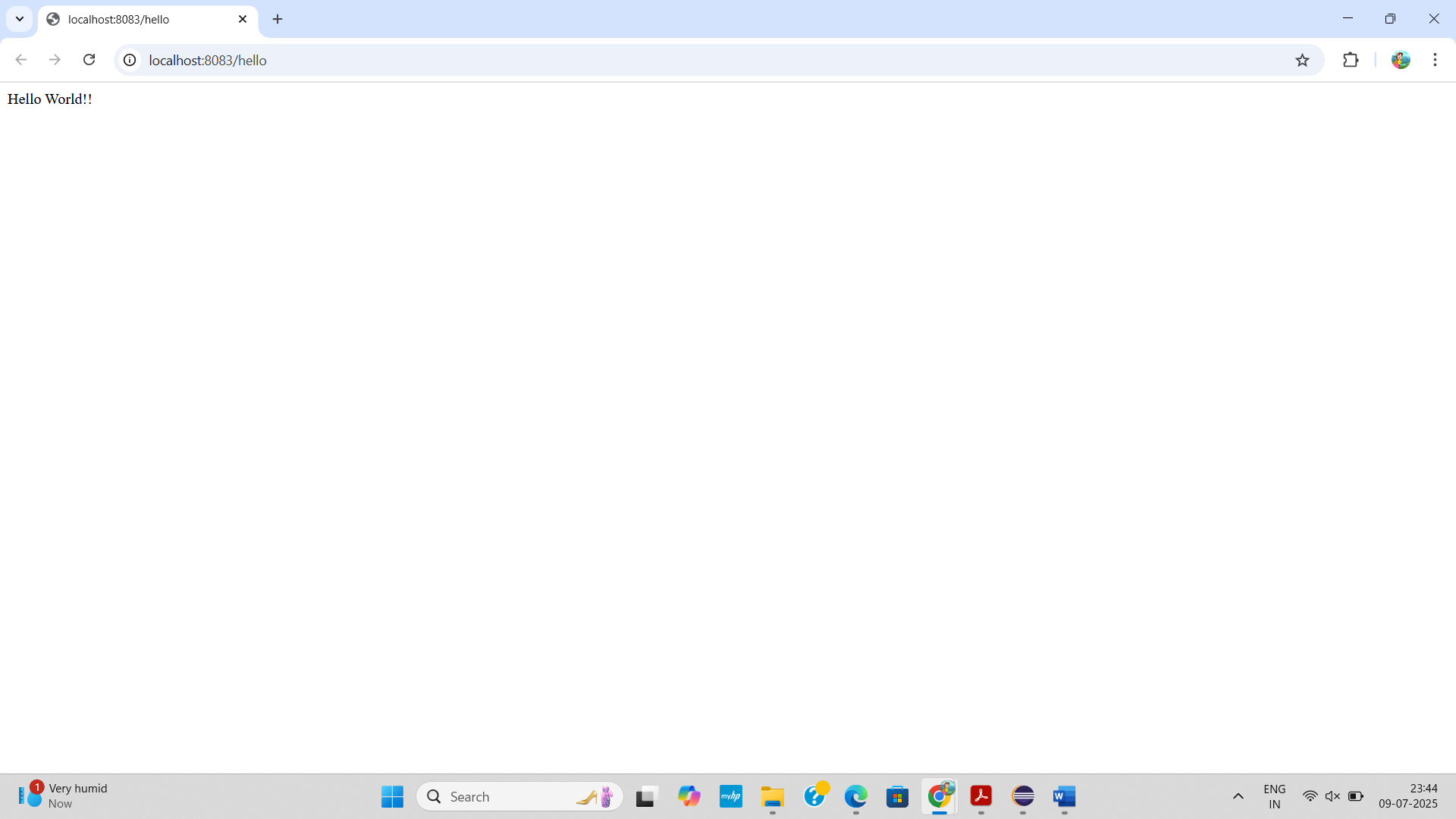
**application.properties**

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

server.port=8083

**OUTPUT**

****



**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"

}

**APPROACH**

* Create POJO Country with code and name fields and getter/setter methods.
* Define XML Bean in in src/main/resources/country.xml with values "IN" and "India".
* Create REST Controller CountryController with @RestController and @RequestMapping("/country").
* Load XML Bean using ClassPathXmlApplicationContext and return the Country object in getCountryIndia().
* Run App and Test at http://localhost:8083/country to receive JSON response via browser or Postman.

**CODE**

**Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {}

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;

}

}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

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

}

}

**CountryController.java**

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 CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

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

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

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

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

return country;

}

}

**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()");

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

return "Hello World!!";

}

}

**application.properties**

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

server.port=8083

**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="in" class="com.cognizant.spring\_learn.Country">

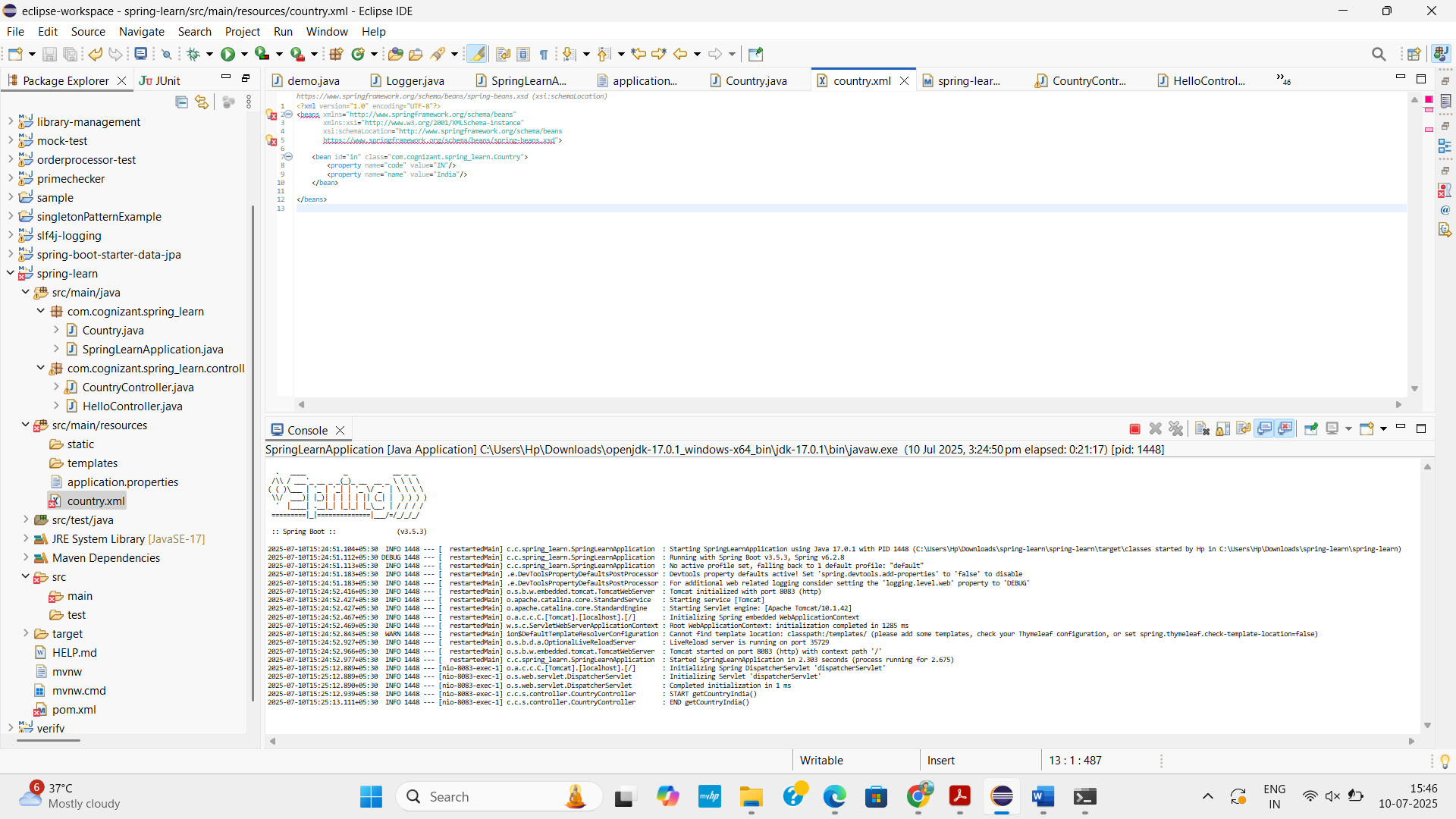
<property name="code" value="IN"/>

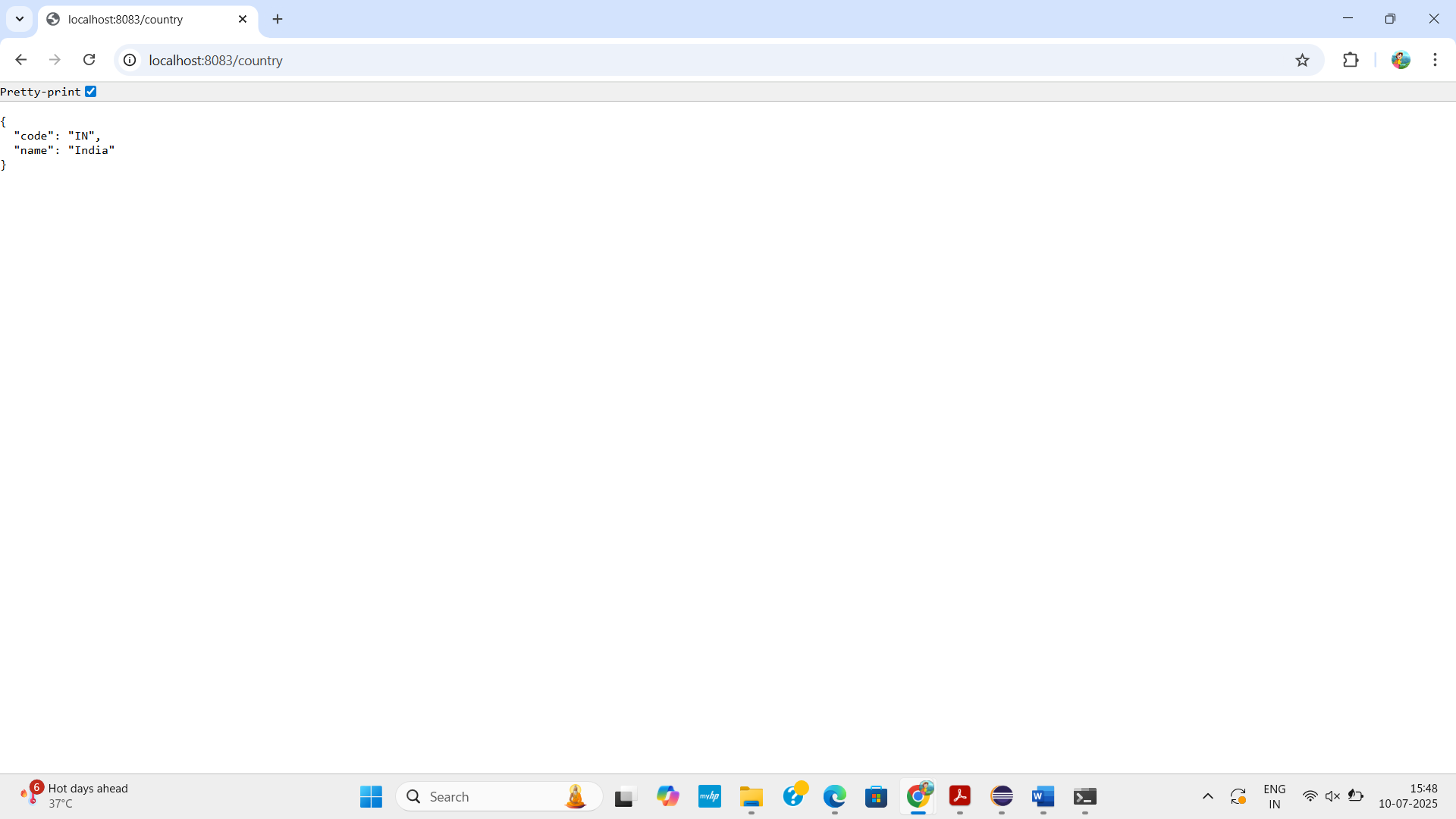
<property name="name" value="India"/>

</bean>

</beans>

**OUTPUT**





**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"

}

**APPROACH**

* Create POJO Country with code and name fields and provide getters/setters.
* Define country.xml in src/main/resources containing a list of Country beans using <bean> and <list> tags.
* Create CountryService class in service package, load the XML list using ClassPathXmlApplicationContext, and use a stream to match country code (case-insensitive).
* Create CountryController class in controller package with endpoint @GetMapping("/country/{code}"), inject CountryService, and return the matched country.
* Run the application and test with URL http://localhost:8083/country/in in browser or Postman to get a JSON response of the matching country.

**CODE**

**Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {}

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;

}

}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

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

}

}

**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("/country/{code}")

public Country getCountry(@PathVariable String code) {

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

Country country = countryService.getCountry(code);

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

return country;

}

}

**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()");

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

return "Hello World!!";

}

}

**CountryService.java**

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.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);

}

}

**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="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

</list>

</constructor-arg>

</bean>

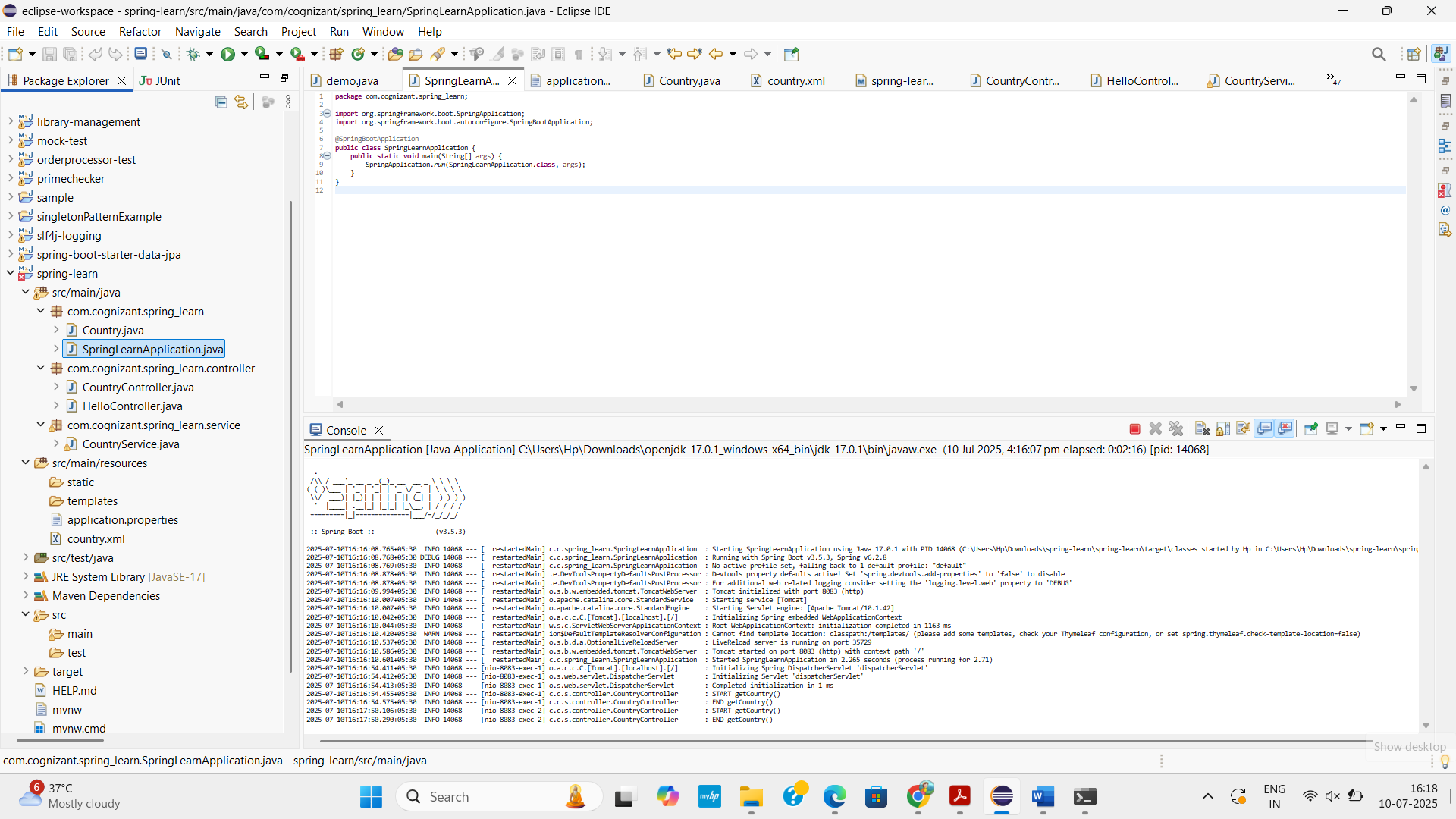
</beans>

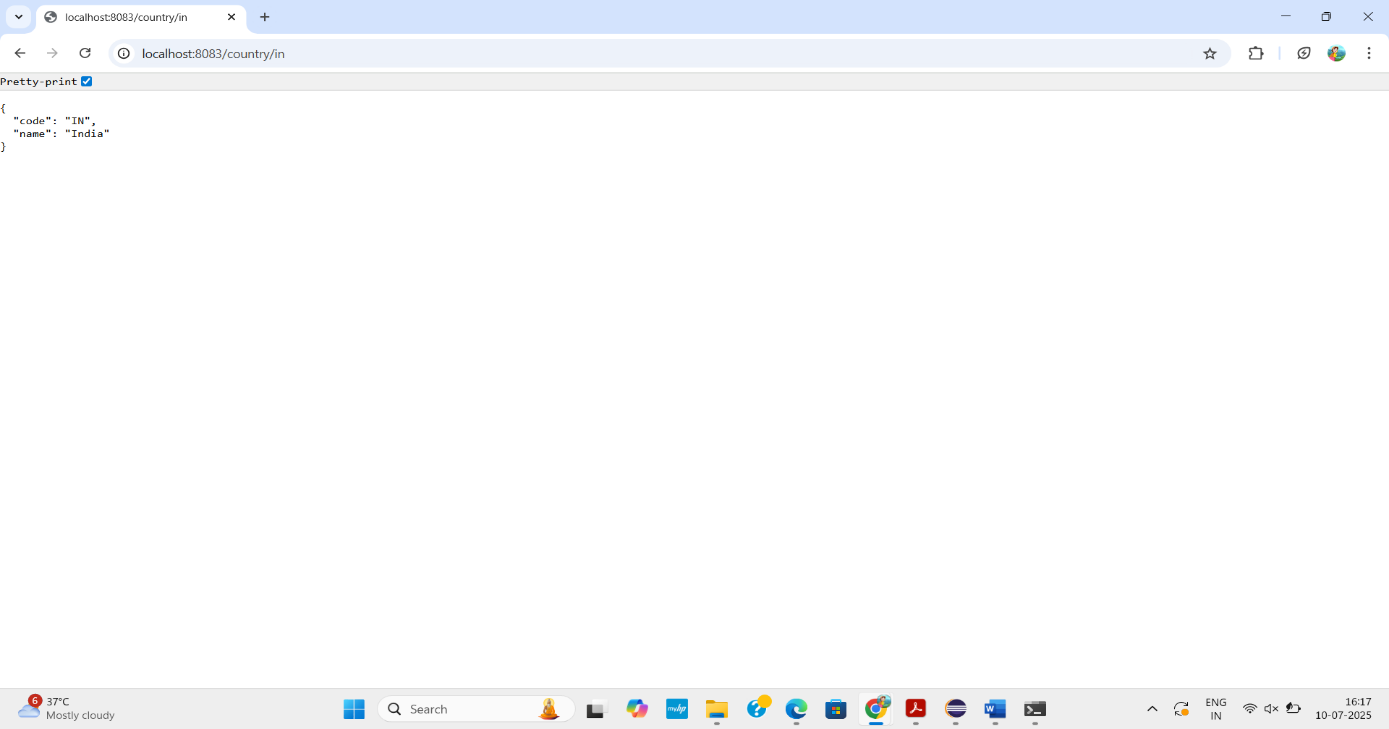
**application.properties**

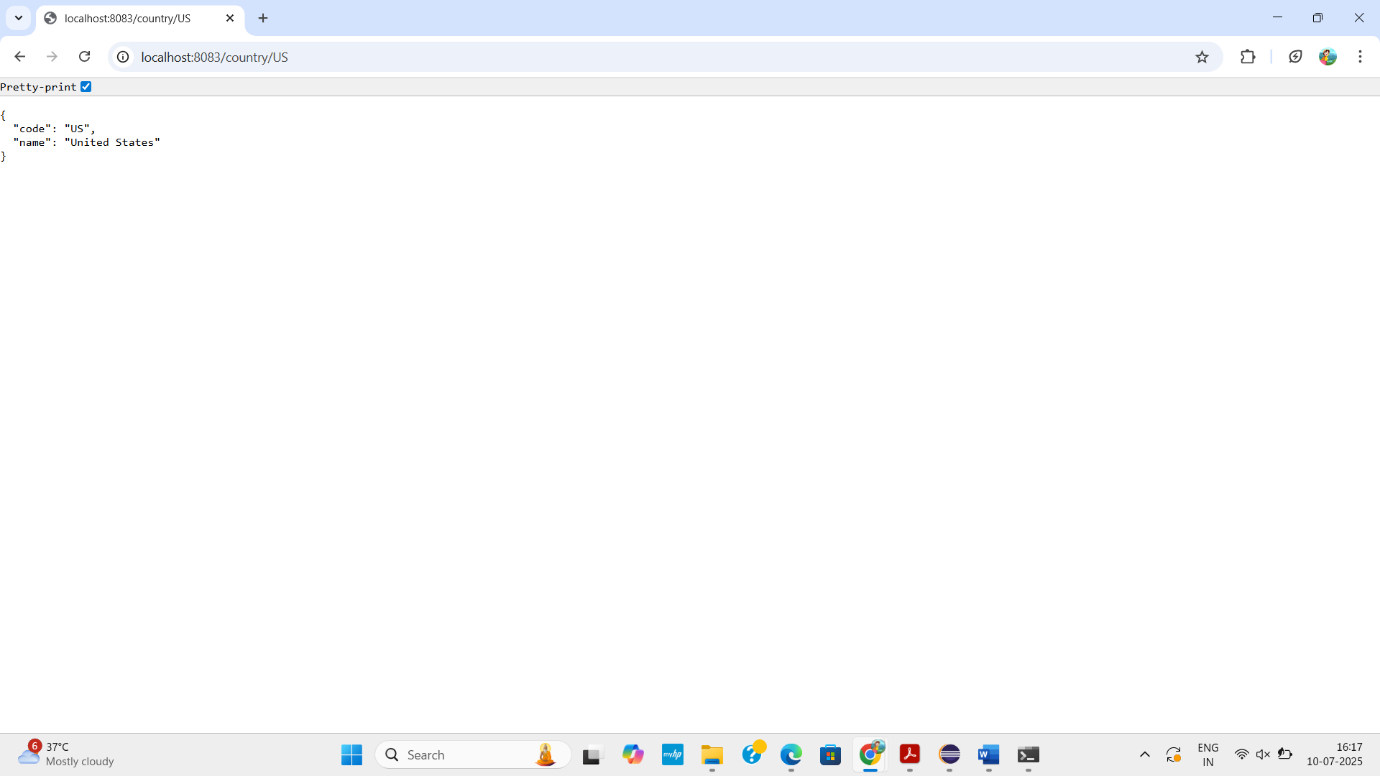
logging.level.com.cognizant.spring\_learn=DEBUG

server.port=8083

**OUTPUT**

****





**CREATE AUTHENTICATION SERVICE THAT RETURNS JWT**

**APPROACH**

* Create a Spring Boot Maven project with Web, Security, and JJWT dependencies.
* Configure SecurityConfig with SecurityFilterChain and in-memory user details.
* Implement AuthenticationController to generate JWT using Keys.secretKeyFor(SignatureAlgorithm.HS256).
* Set application.properties (e.g., spring.application.name, server.port=8083).
* Run the project and test the /authenticate endpoint using Basic Authentication to receive a JWT token.

**CODE**

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

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

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

import java.security.Key;

import java.util.Base64;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

@RestController

public class AuthenticationController {

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

private final Key key = Keys.secretKeyFor(SignatureAlgorithm.HS256);

@GetMapping("/authenticate")

public Map<String, String> authenticate(@RequestHeader("Authorization") String authHeader) {

LOGGER.info("START authenticate()");

String user = getUser(authHeader);

String token = generateJwt(user);

Map<String, String> map = new HashMap<>();

map.put("token", token);

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

return map;

}

private String getUser(String authHeader) {

LOGGER.info("START getUser()");

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

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

String credentials = new String(credDecoded);

String user = credentials.split(":")[0];

LOGGER.info("Decoded User: {}", user);

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

return user;

}

private String generateJwt(String user) {

LOGGER.info("START generateJwt()");

String token = Jwts.builder()

.setSubject(user)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 1200000)) // 20 minutes

.signWith(key)

.compact();

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

return 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.config.annotation.web.configuration.EnableWebSecurity;

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

@EnableWebSecurity

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http.csrf().disable()

.authorizeHttpRequests(authz -> authz

.requestMatchers("/authenticate").hasAnyRole("USER", "ADMIN")

.anyRequest().authenticated())

.httpBasic();

return http.build();

}

@Bean

public UserDetailsService userDetailsService() {

UserDetails user = User.withDefaultPasswordEncoder()

.username("user")

.password("pwd")

.roles("USER")

.build();

return new InMemoryUserDetailsManager(user);

}

}

**application.properties**

server.port=8083

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

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

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

}

}

**OUTPUT**

