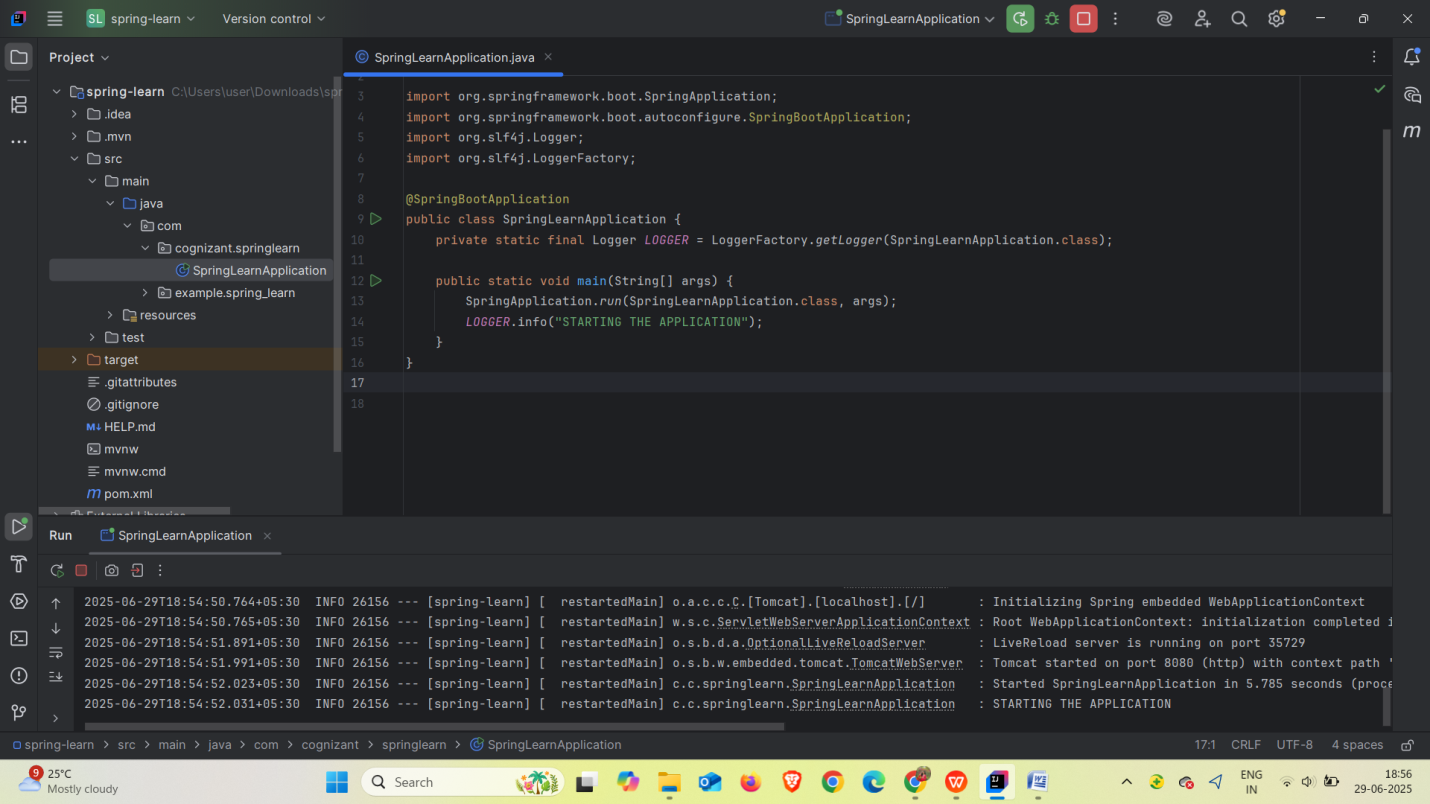
**WEEK-4**

**Hands on 1**

**Create a Spring Web Project using Maven**



SME Walkthrough Points

1. `src/main/java`  
\*Java application code lives here

\*Follows base package structure:

`com.cognizant.springlearn`

Contains:

\* Main class

\* Future controllers, services, models

2. `src/main/resources`

For configuration files

\* `application.properties` goes here

\* Static resources and templates

\* Spring reads this for boot-time setup

3. `src/test/java`

For unit testing

\* IntelliJ supports running tests easily

4. `SpringLearnApplication.java`

\* This is the main class

\* Contains the `main()` method which bootstraps the Spring Boot app

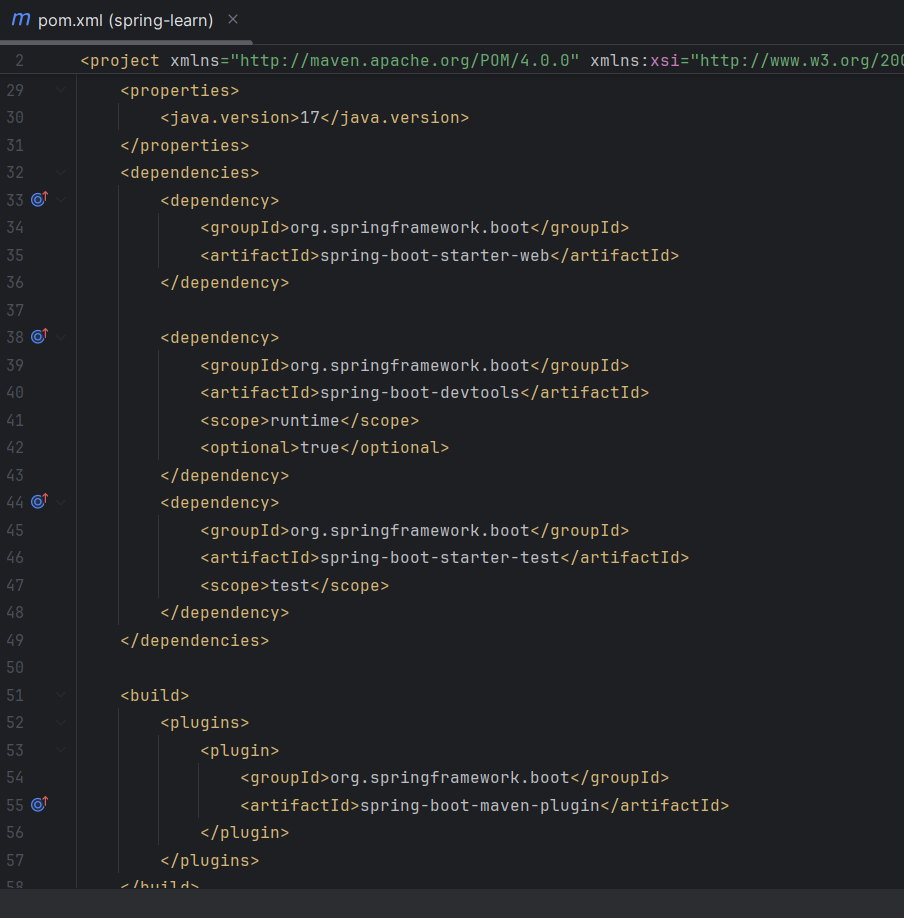
\* Uses: SpringApplication.run(SpringLearnApplication.class, args);

5.Purpose of `@SpringBootApplication`

This is a meta-annotation that includes:

|  |  |
| --- | --- |
| ANNOTATION | PURPOSE |
| |  | | --- | | @Configuration |  |  | | --- | |  | | |  | | --- | | Marks this class as Spring config class |  |  | | --- | |  | |
| |  | | --- | | @EnableAutoConfiguration |  |  | | --- | |  | | |  | | --- | | Enables Spring Boot's auto setup |  |  | | --- | |  | |
| |  | | --- | | @ComponentScan |  |  | | --- | |  | | Enables scanning for beans and components |

6. `pom.xml` – Walkthrough



7. Dependency Hierarchy in IntelliJ

1. Open the `pom.xml`

2. At the bottom, click the or "Maven" tab

3. Expand `spring-boot-starter-web` → See all sub-dependencies like:

\* `spring-web`

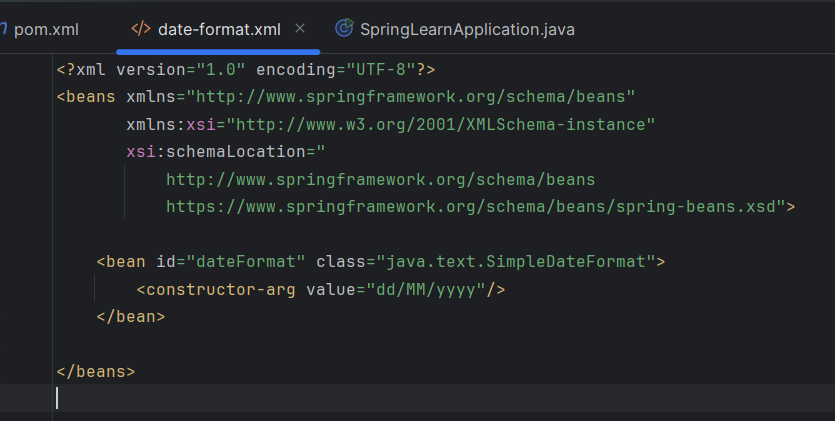
\* `spring-core`

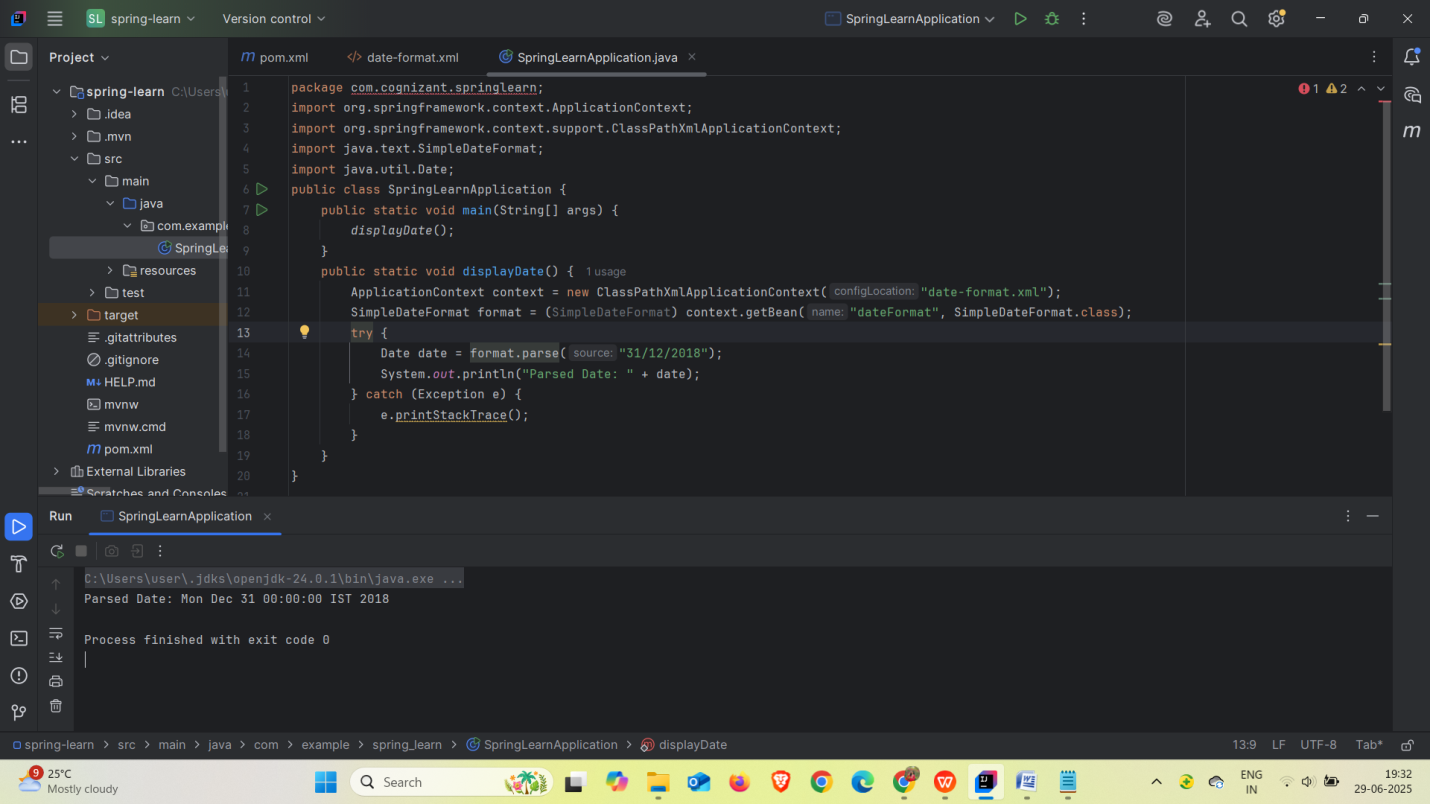
\* `jackson`

\* `spring-context`, etc.

**Hands on 2**

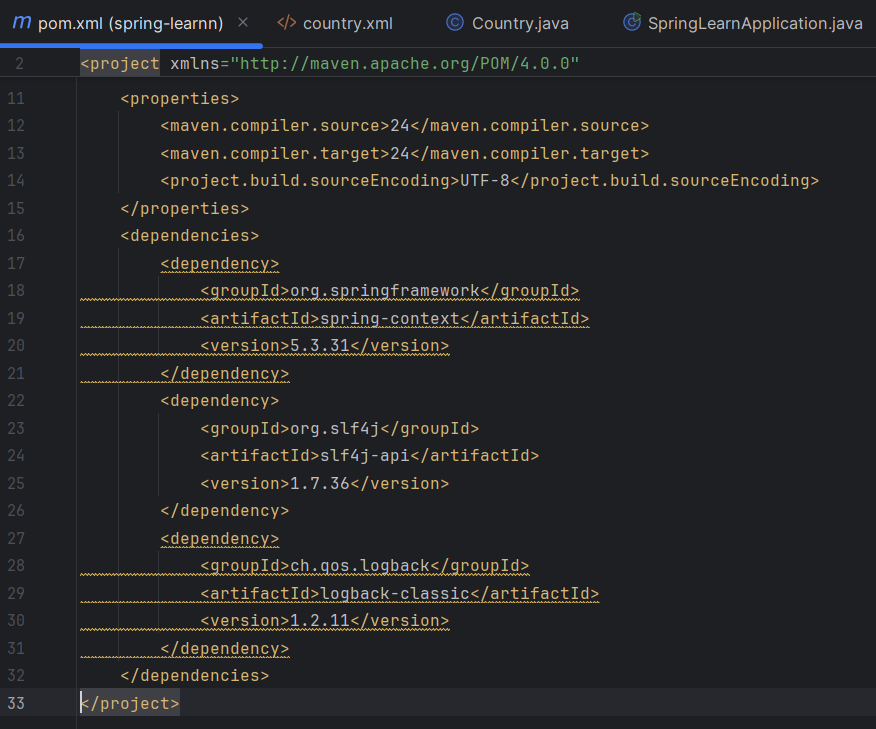
**Spring Core – Load SimpleDateFormat from Spring Configuration XML**

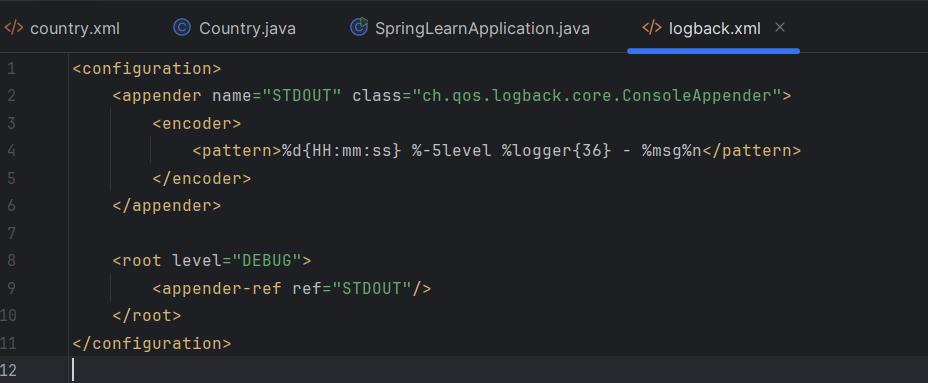




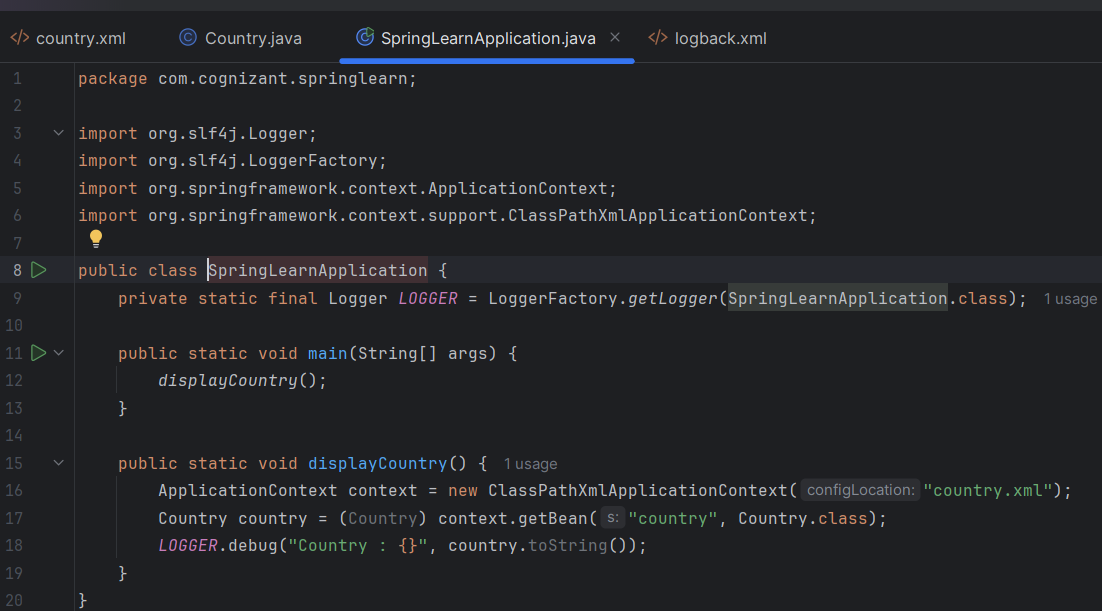
**Hands on 4**

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

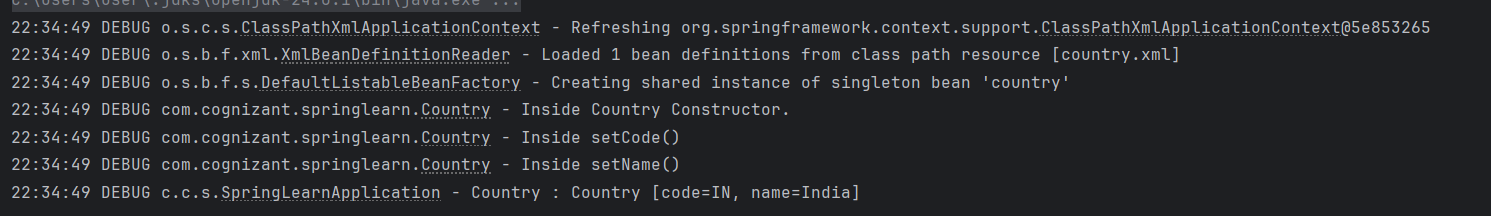
****

****

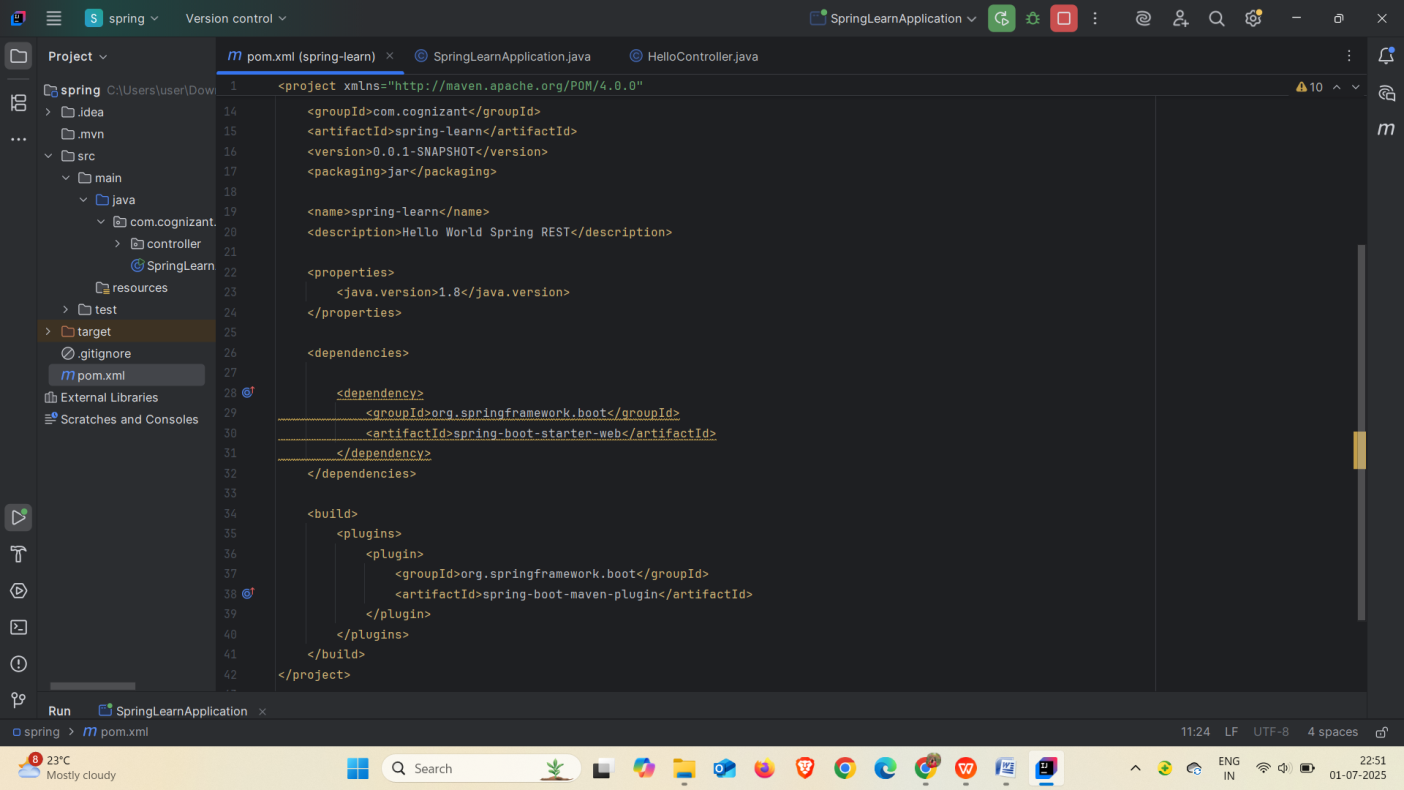
****

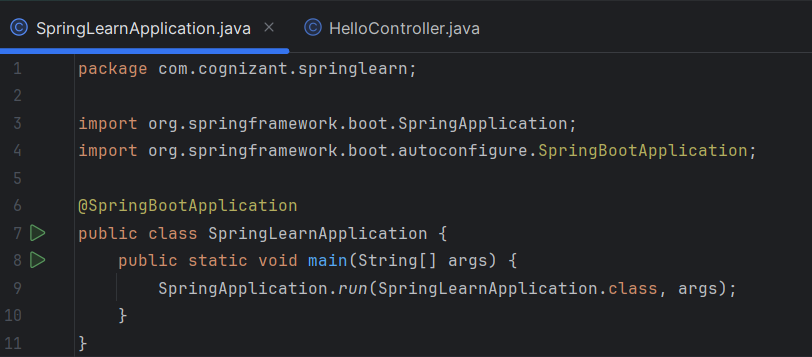
****

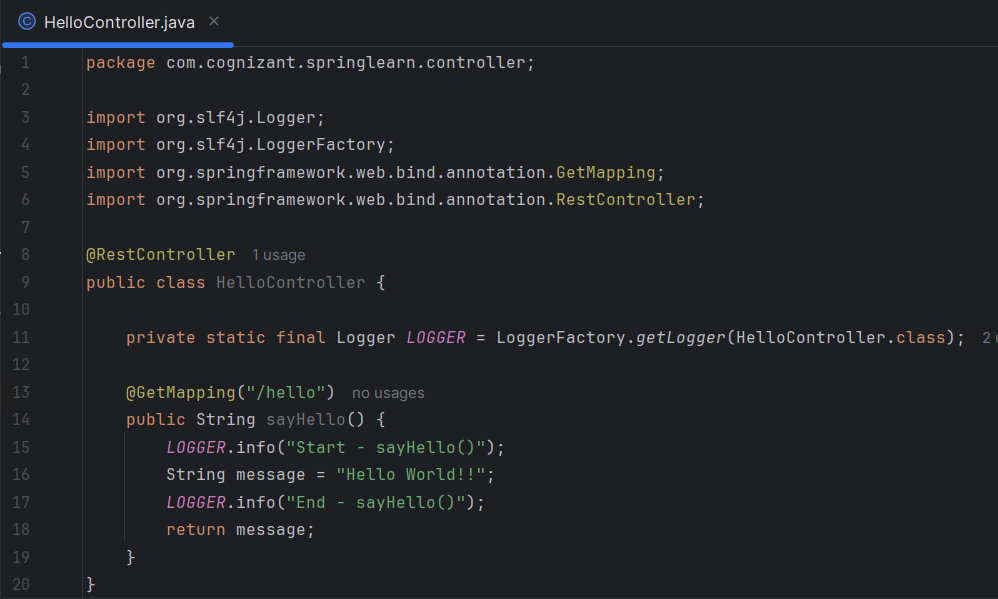
**OUTPUT**

****

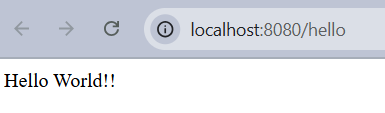
**Hello World RESTful Web Service**



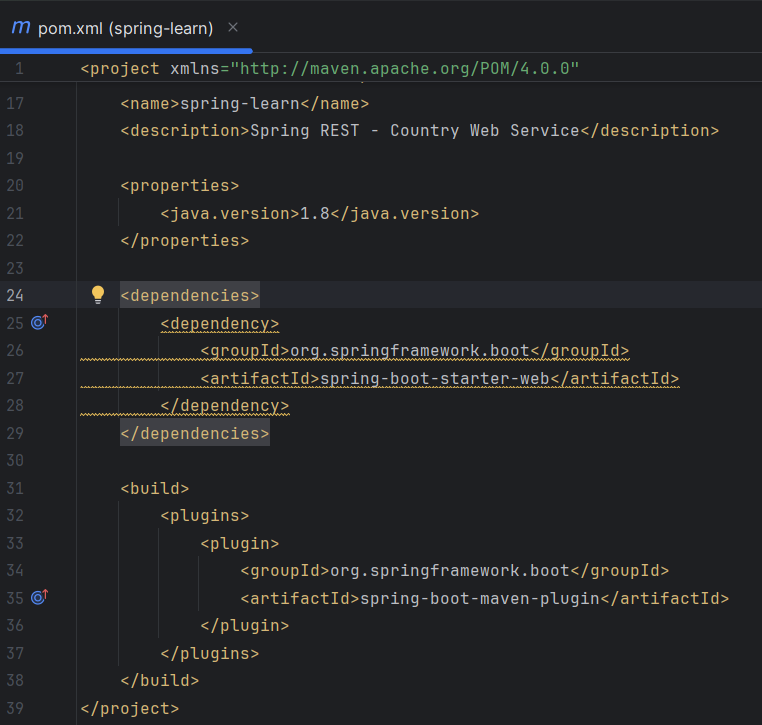


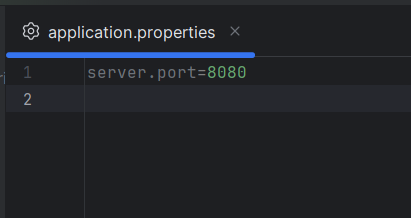


**OUTPUT**

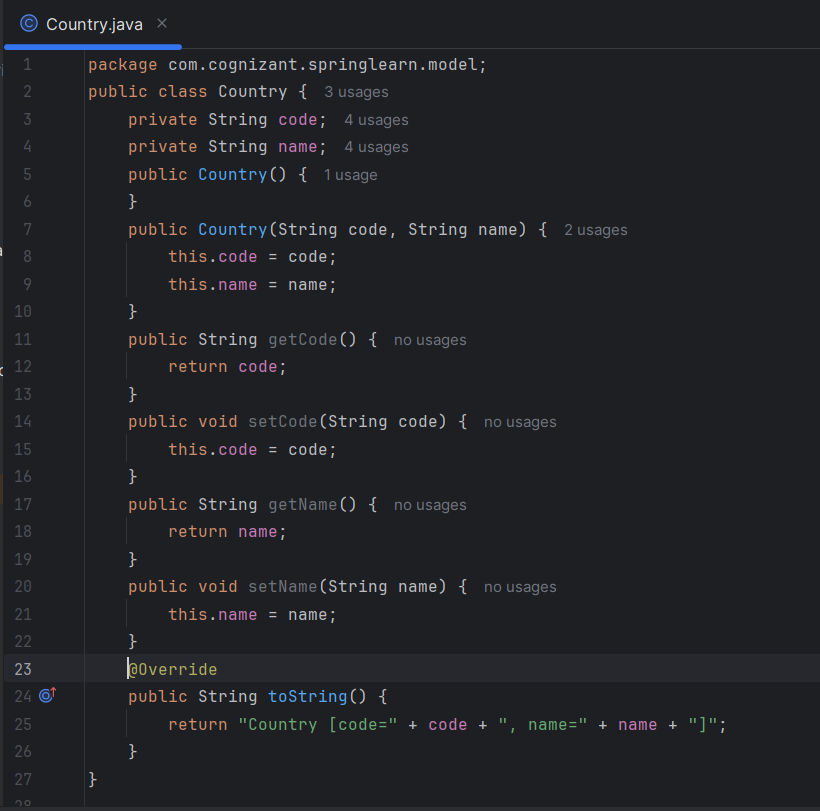
****

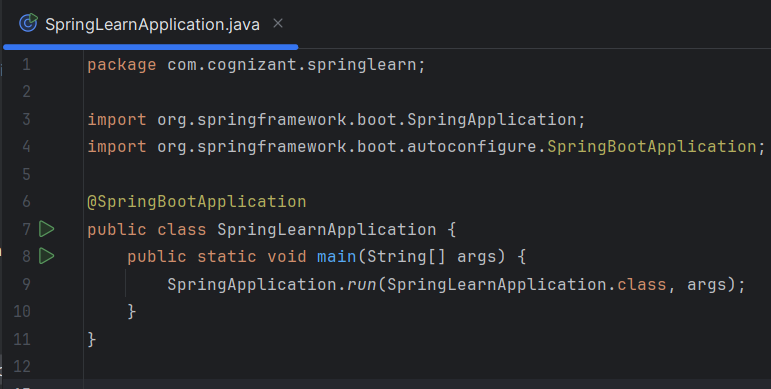
**REST - Country Web Service**

****

****

****

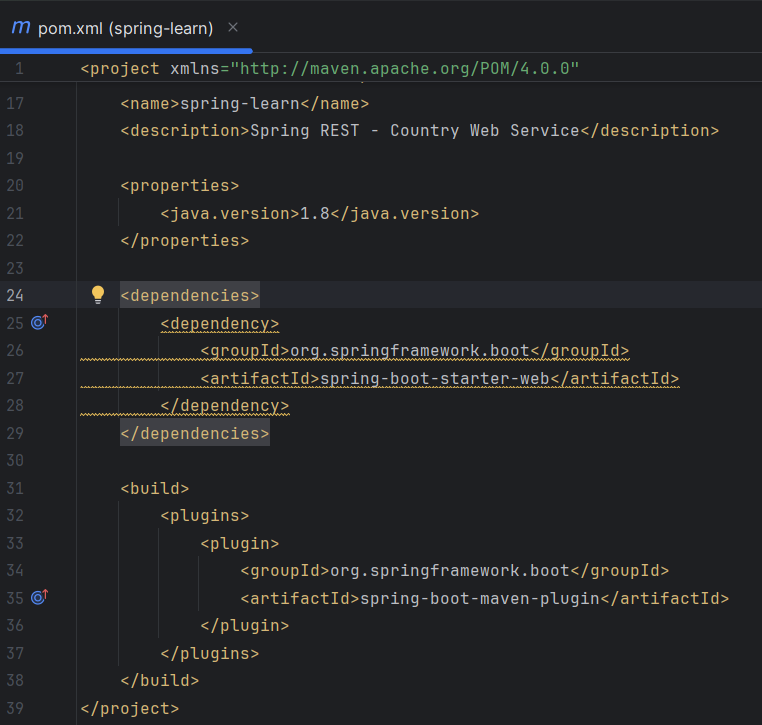
****

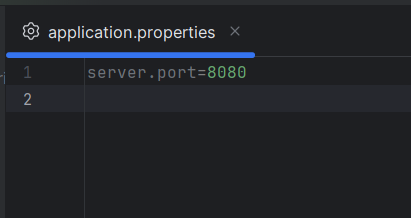
****

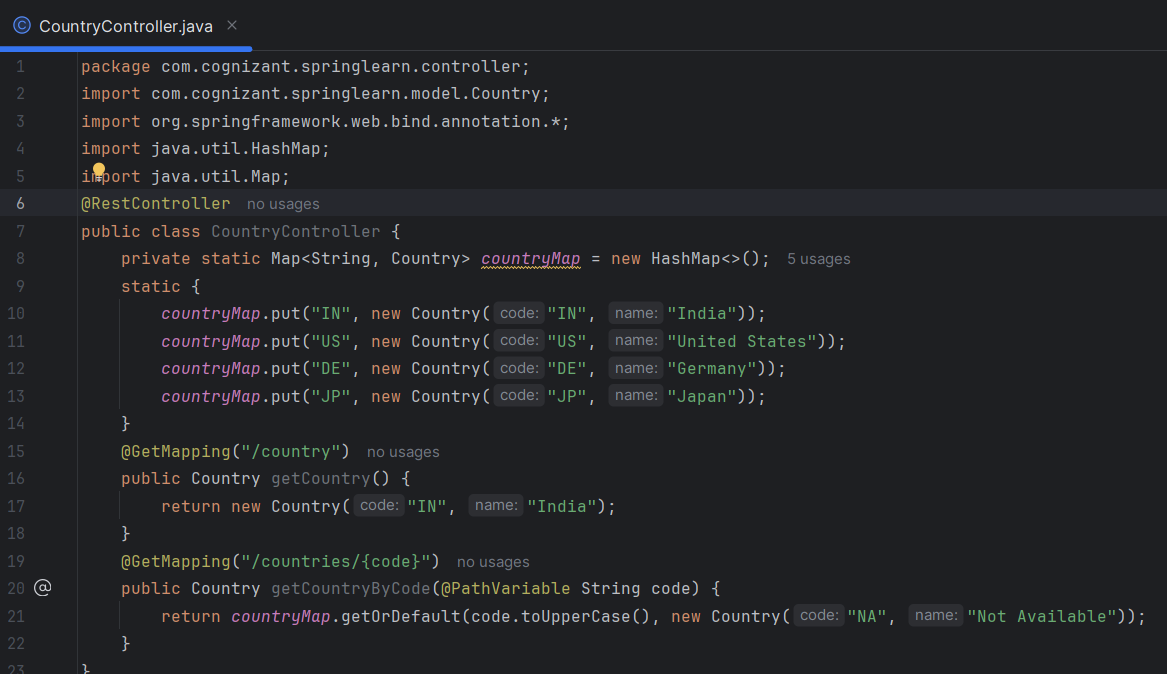
**OUTPUT**

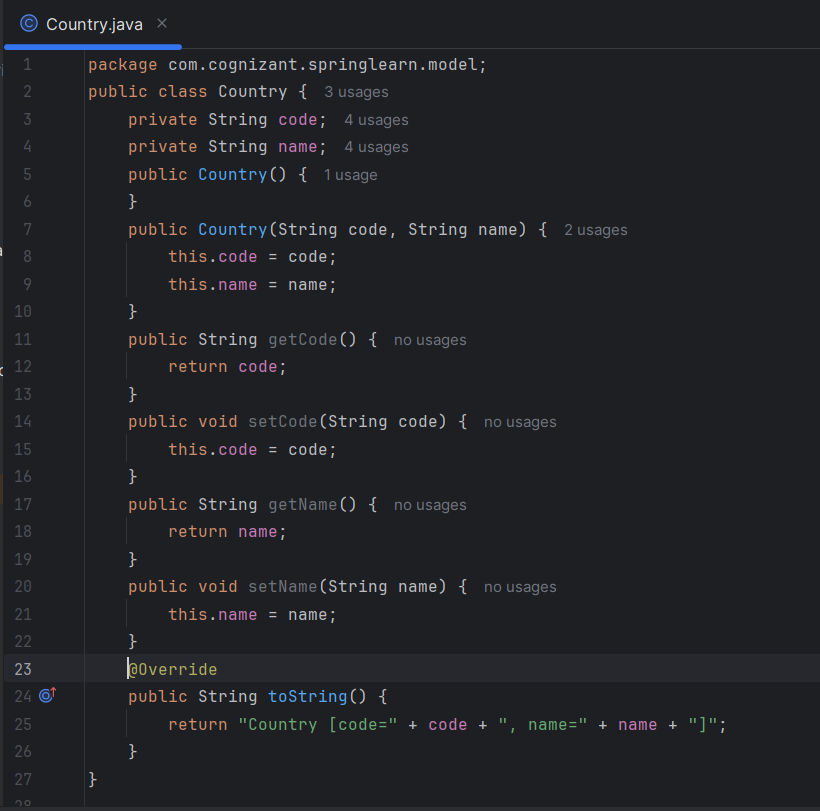
****

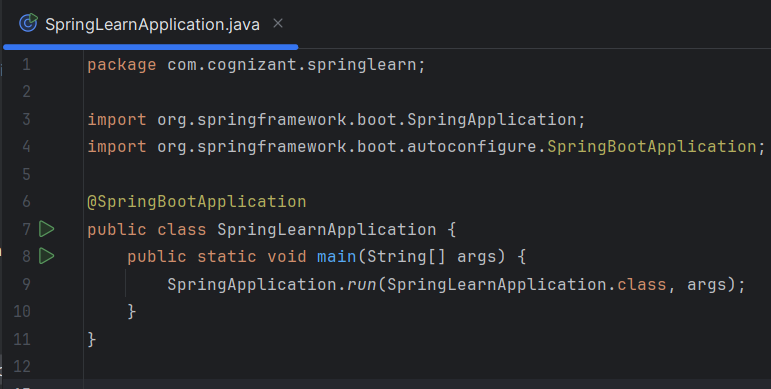
**REST - Get country based on country code**



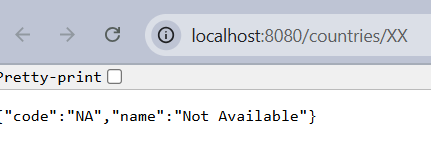
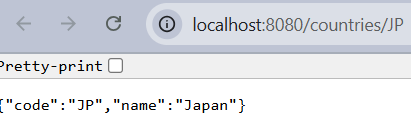
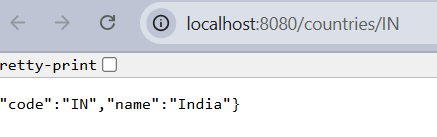


****

****

****

**OUTPUT**

****