**WEEK-4**

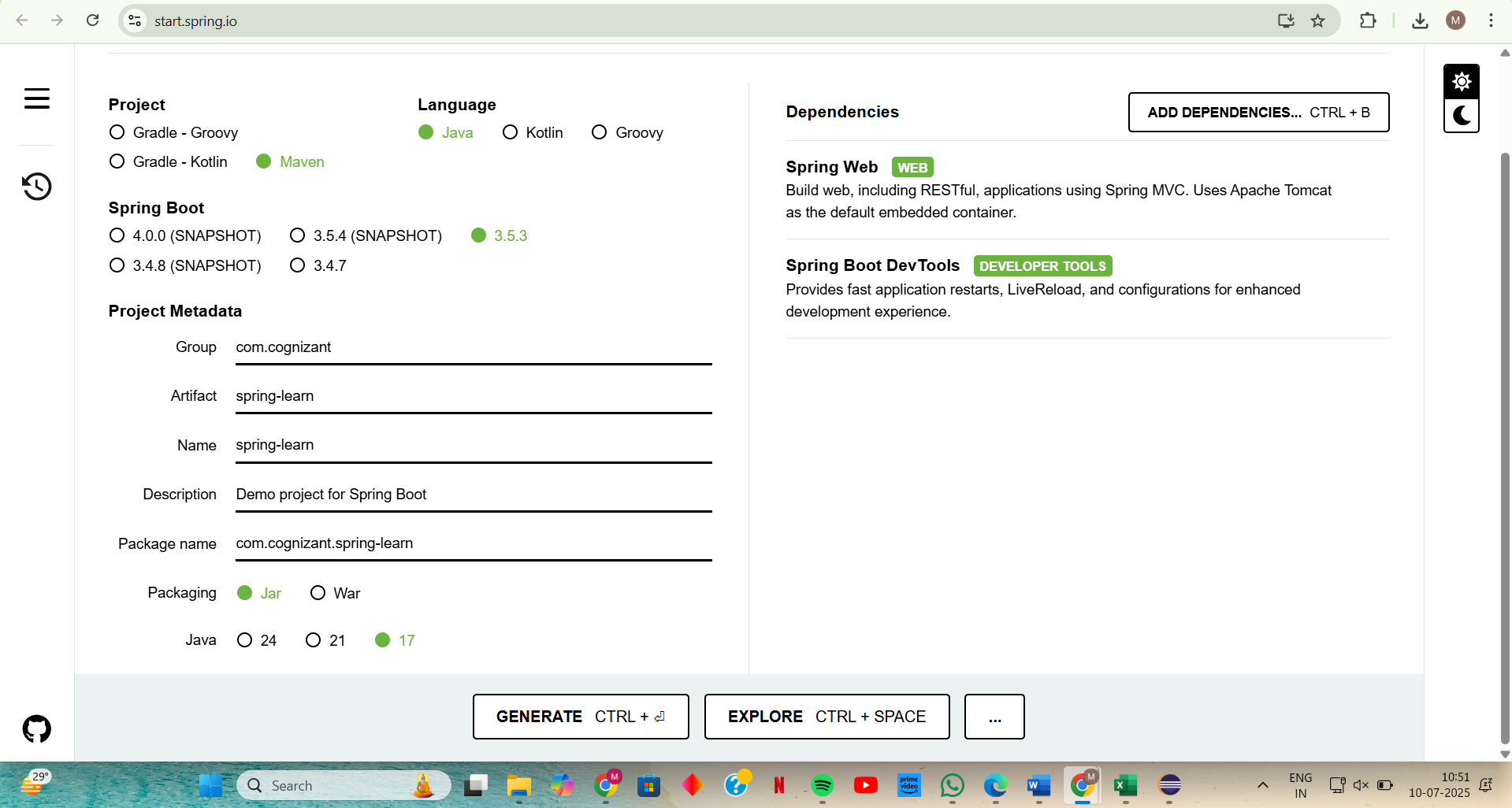
**SPRING REST USING SPRING BOOT 3**

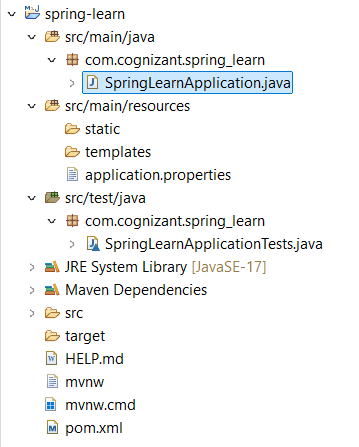
**SPRING REST HANDS ON**

**Create a Spring Web Project using Maven**

**OUTPUT:**

**STEP 1: Created and downloaded the project zip**

****



Here

1. **src/main/java – Folder with Application Code**

* This folder contains your source code.
* com.cognizant.spring\_learn.SpringLearnApplication.java  
  is the main class that runs your Spring Boot application.

2. **src/main/resources – Folder for Application Configuration**

This folder is for external configuration and static files.

Key parts:

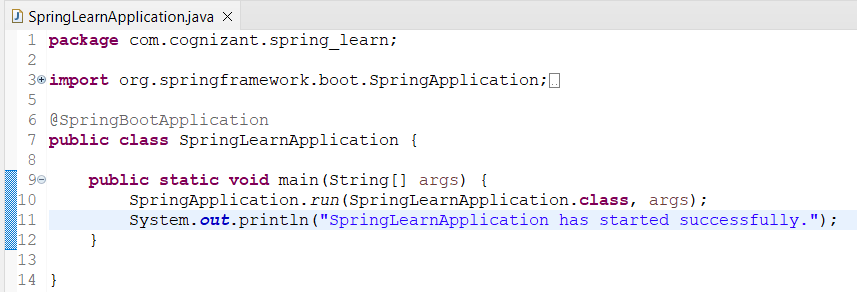
* application.properties – for setting port, DB URL, etc.
* static/ – used for static assets like CSS, JS, images.
* templates/ – used when using view technologies like Thymeleaf.

3. **src/test/java – Folder for Testing Code**

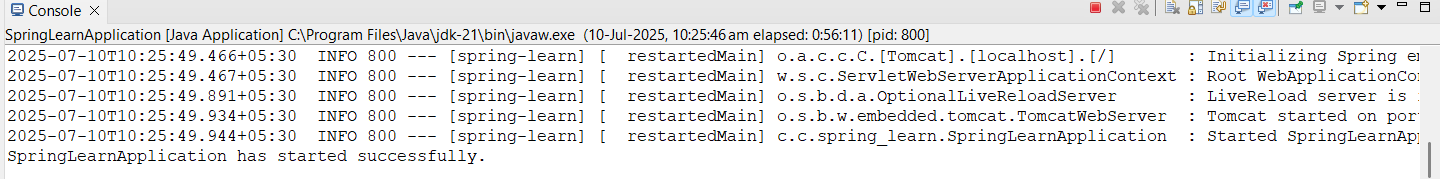
* This holds JUnit or Spring Boot test cases.
* Contains:
  + SpringLearnApplicationTests.java — Default generated test class

4. **SpringLearnApplication.java – Walkthrough**

* This is the entry point of your Spring Boot app.
* SpringApplication.run() launches your application context**.**

****

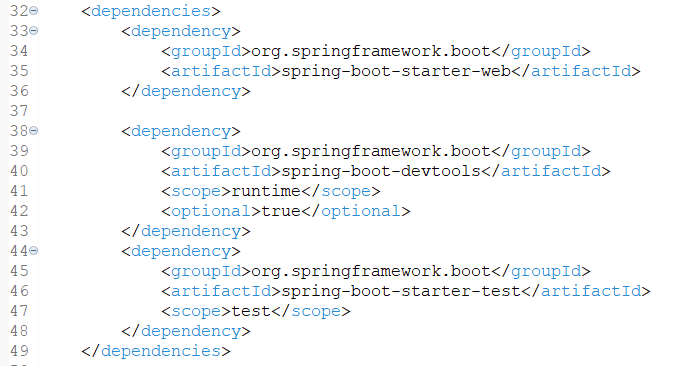
**Code output:**

****

**5. Purpose of @SpringBootApplication**

| Annotation | Purpose |
| --- | --- |
| @Configuration | Defines a configuration class |
| @EnableAutoConfiguration | Enables Spring Boot to configure the application based on dependencies |
| @ComponentScan | Scans the base package and its sub-packages for Spring Beans (components, services, repositories, etc.) |

**6. pom.xml Configuration file**

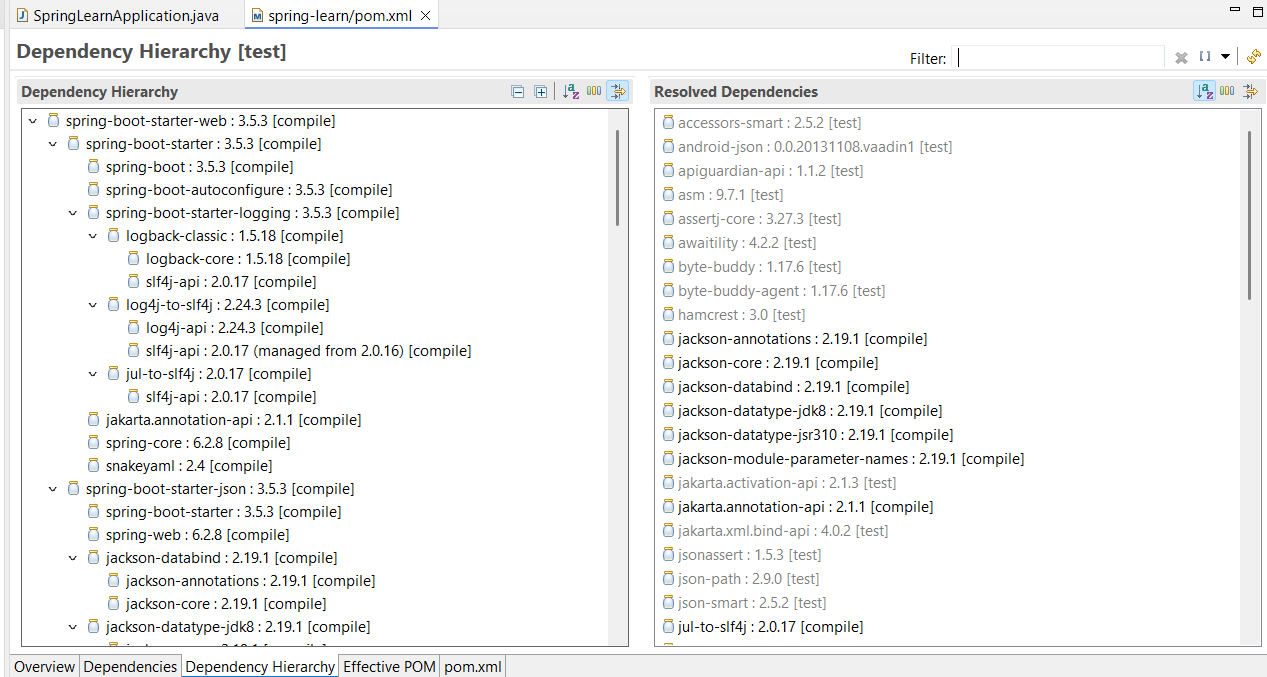
****

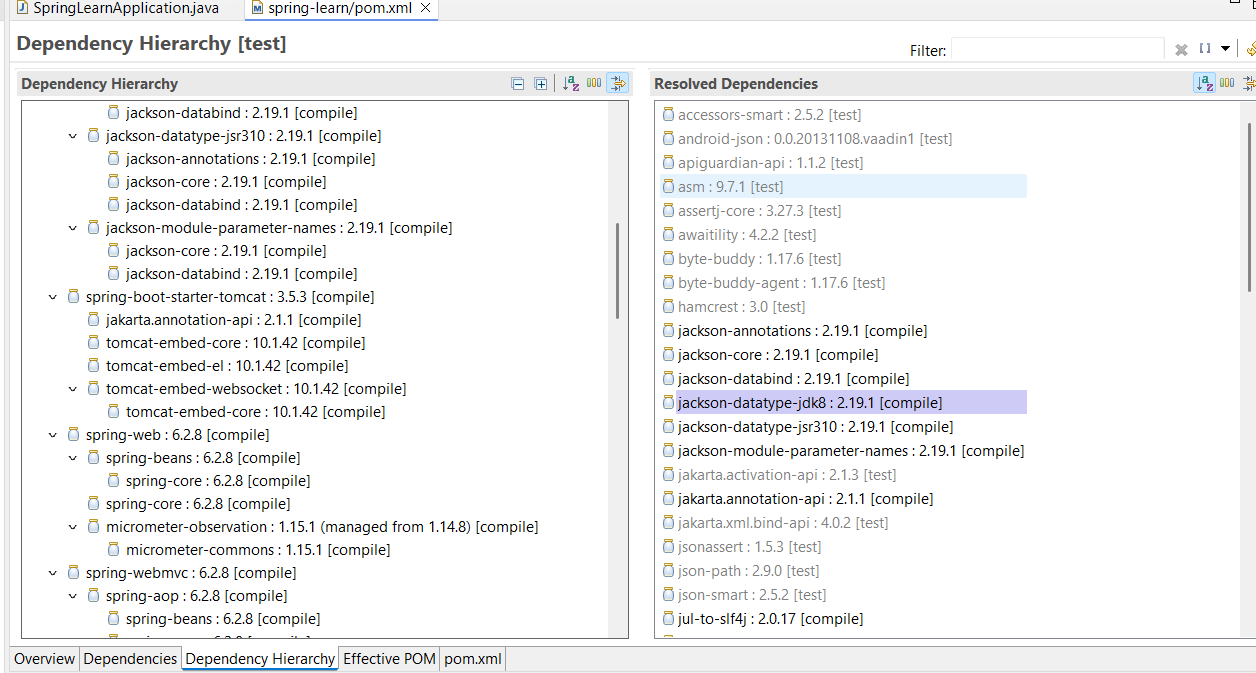
**Explanation:**

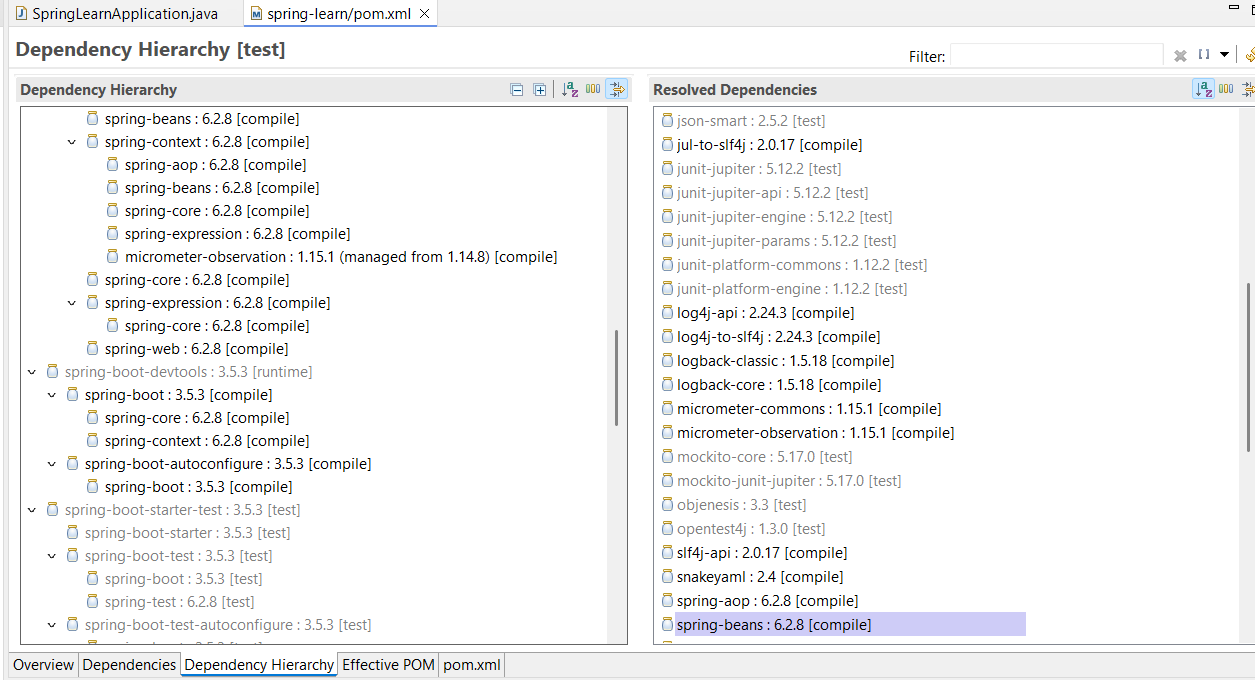
**spring-boot-starter-web –** adds REST API, embedded Tomcat, etc.

**spring-boot-devtools** – enables auto-restart during development.

**7. dependency hierarchy**

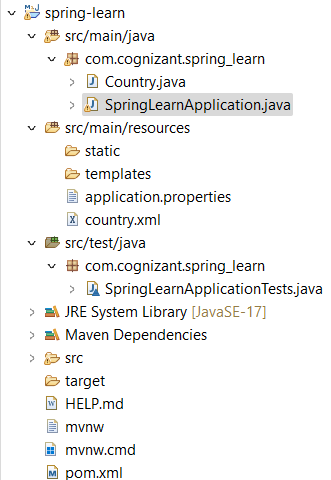
****

****

****

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

**Folder Structure:**

****

**CODE:**

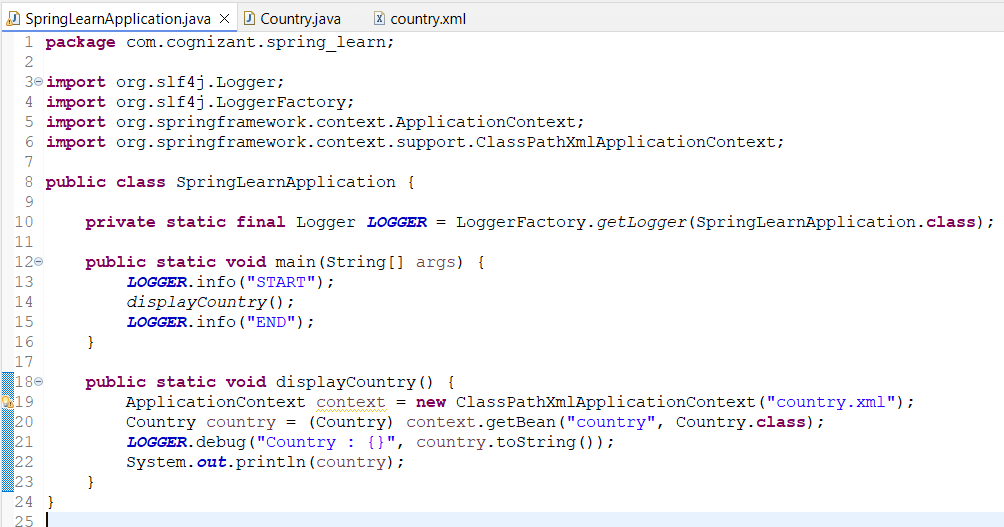
**Step 1: Create Country.java class**

****

**Step 2: Create country.xml Spring Configuration File**

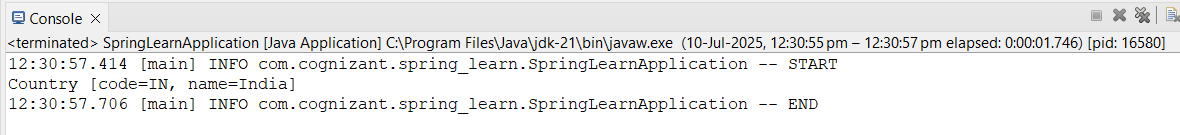
****

**Step 3: Modify SpringLearnApplication.java to Add displayCountry()**

****

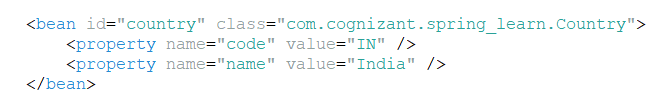
**Step 4:**

**OUTPUT:**

****

**Step 5: Concepts to Understand for SME Walkthrough**

**bean tag and attributes:**

****

* id: unique name used to access the bean.
* class: fully qualified class name of the bean.
* property: injects values into setter methods using name and value**.**

**ApplicationContext , ClassPathXmlApplicationContext**

* ApplicationContext: Spring’s central interface for providing configuration.
* ClassPathXmlApplicationContext: Implementation of ApplicationContext that loads the context from an XML file located in the classpath (like country.xml).

**What happens during context.getBean()?**

* Spring reads the XML config.
* Finds the <bean> definition.
* Instantiates the class via constructor.
* Calls setters for properties (setCode(), setName()).
* Returns a fully initialized bean instance.