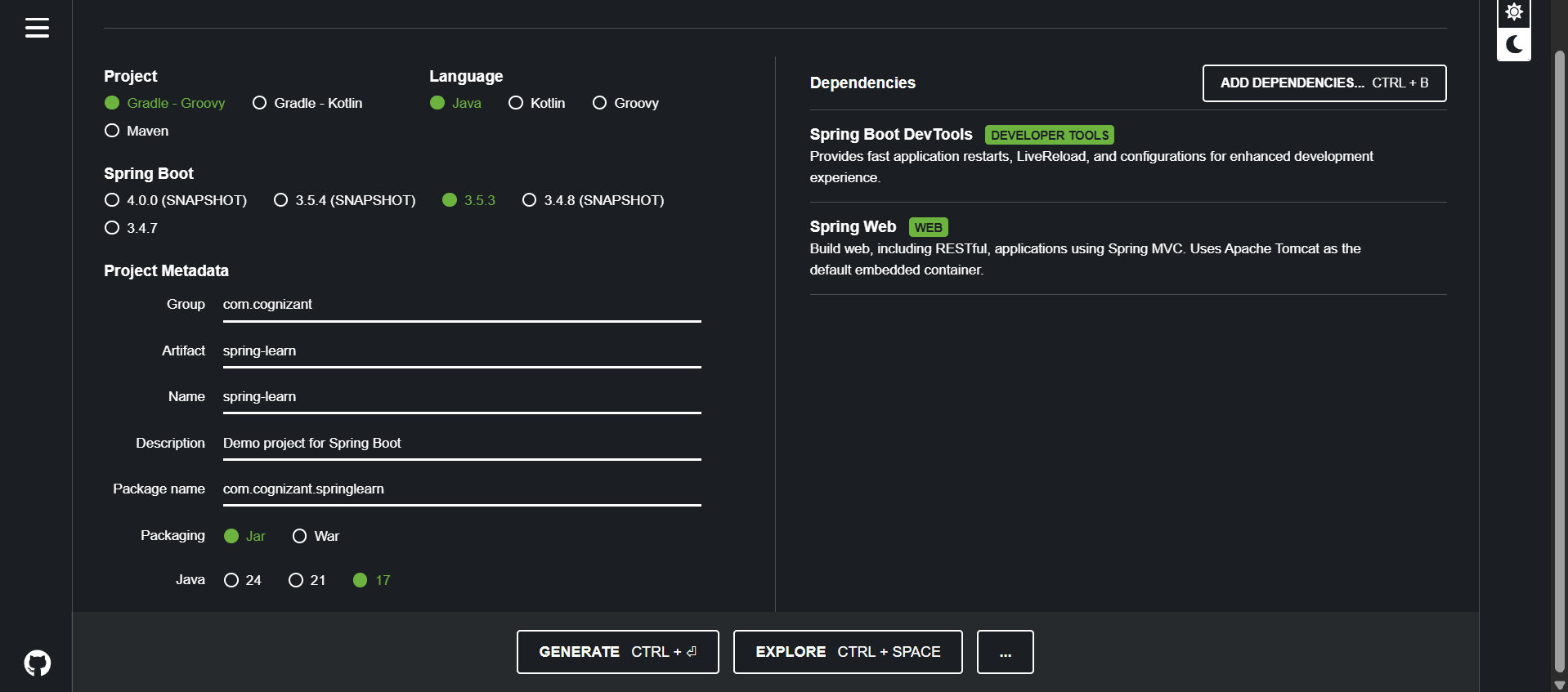
WEEK-4

**1.spring-rest-handson**

**Create a Spring Web Project using Maven**



**SpringLearnApplication.java:**

package com.cognizant.springlearn;

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) {

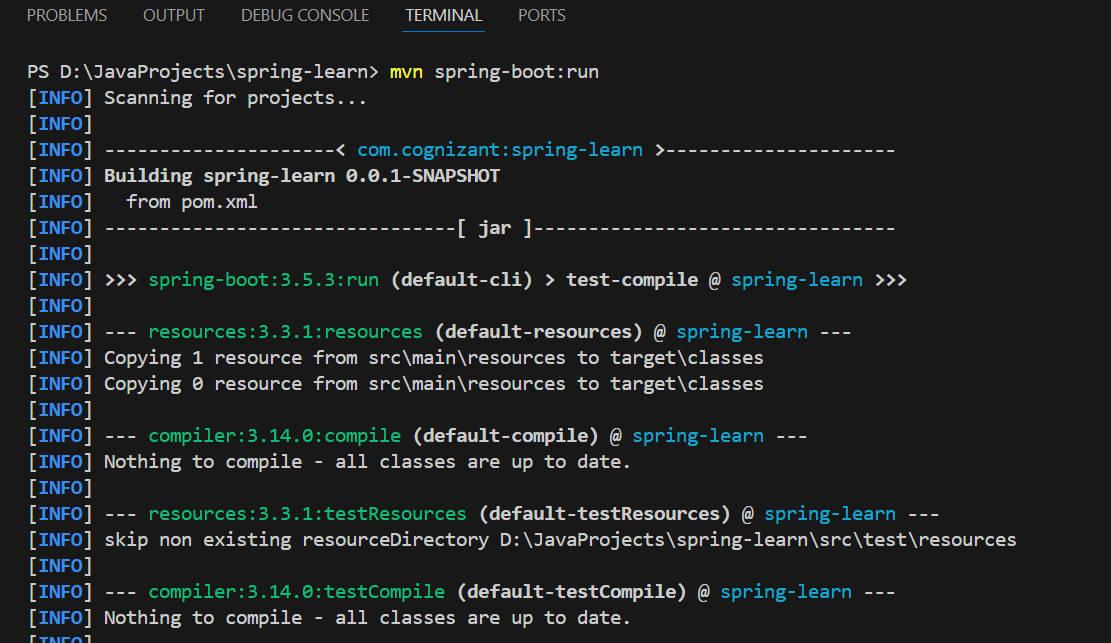
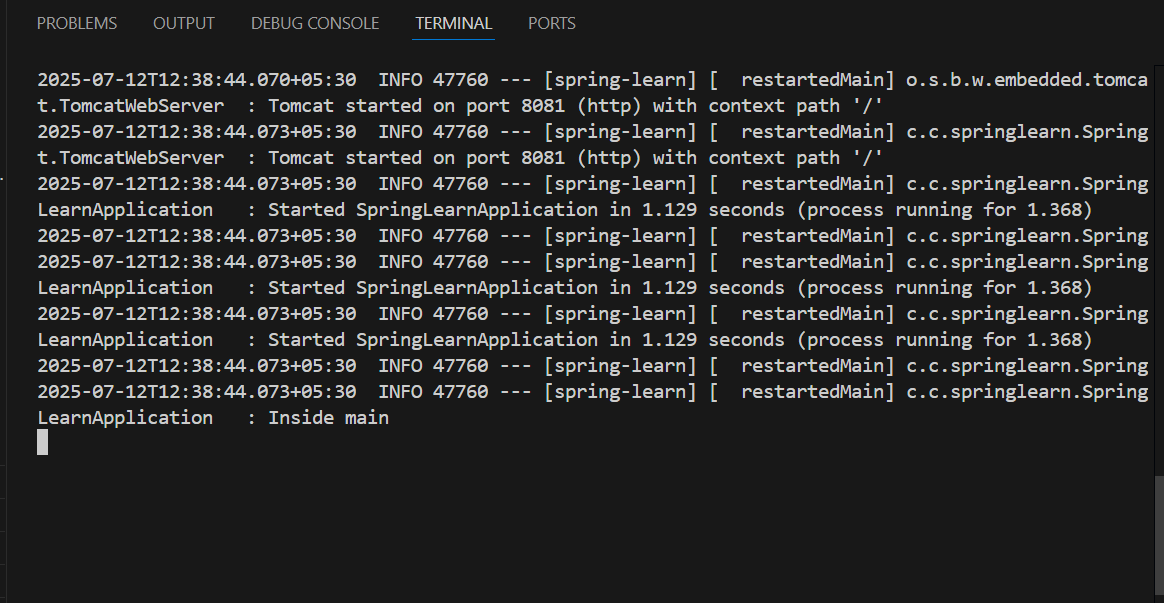
SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("Inside main");

}

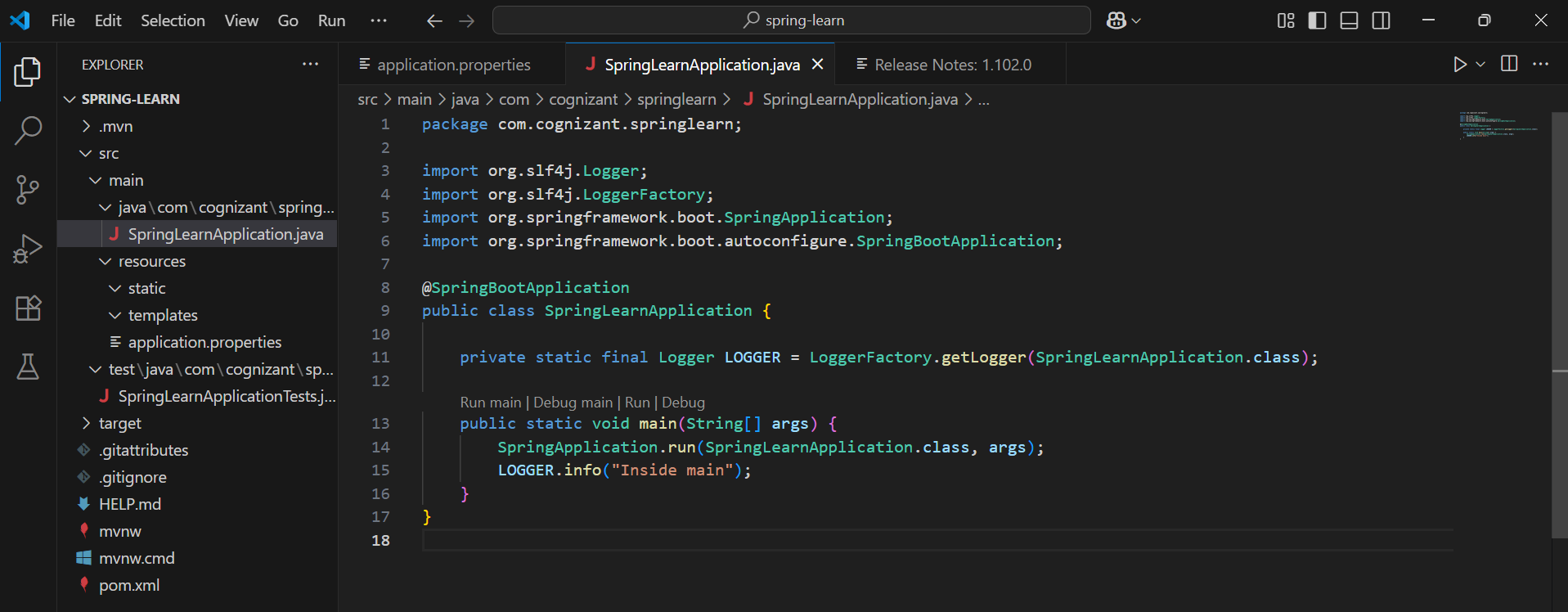
}

**Output:**

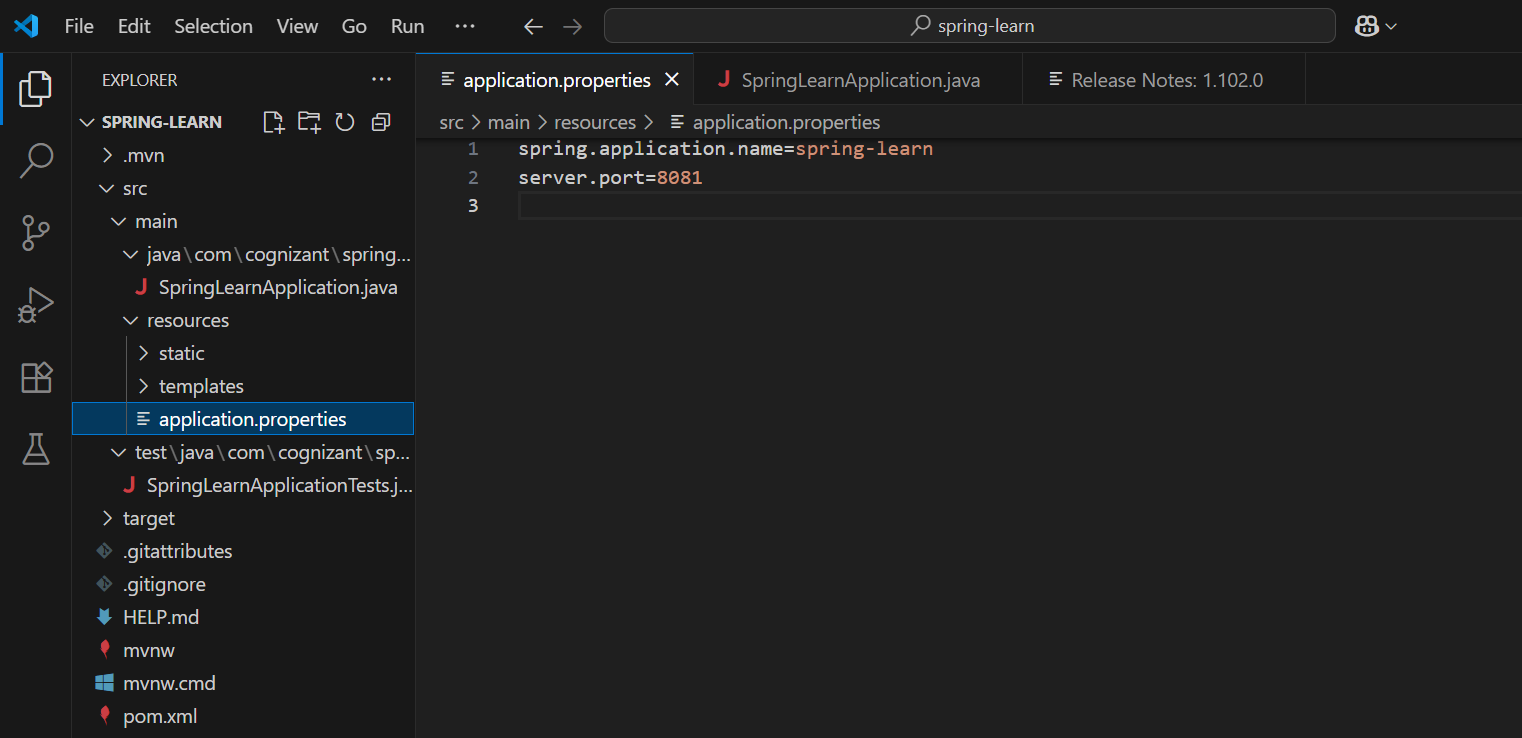
 

**WALKTHROUGH:**

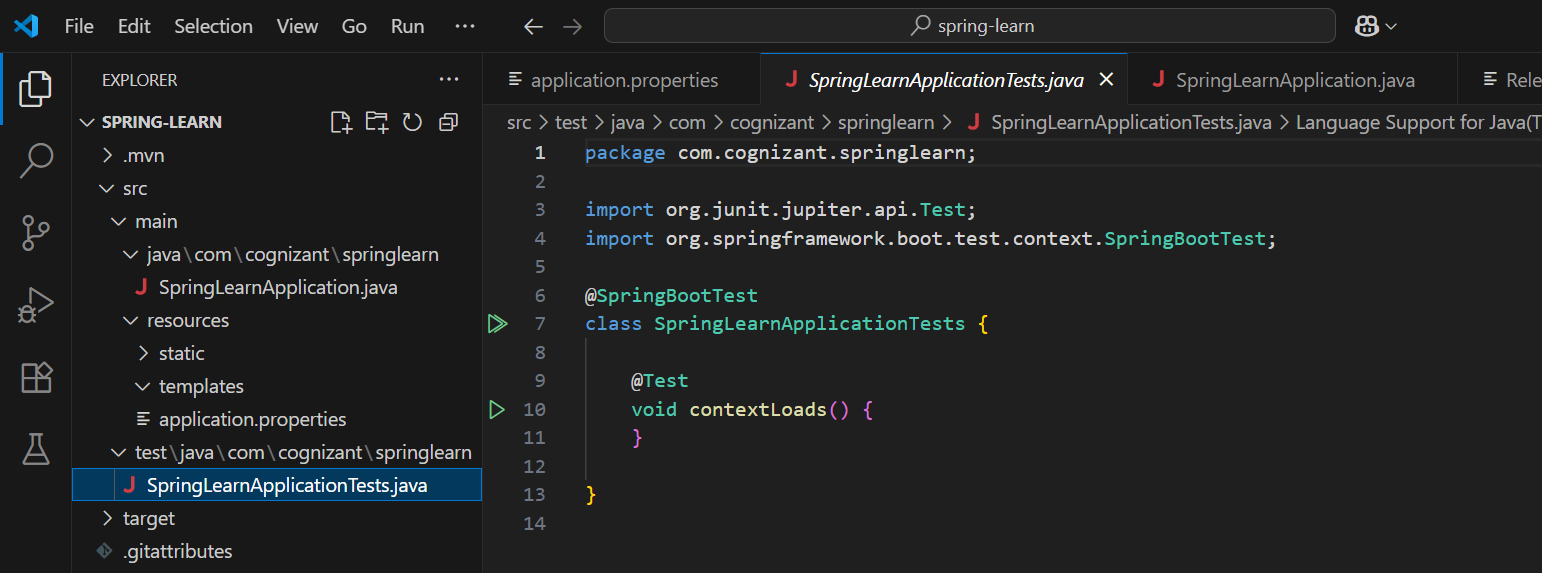
1. src/main/java - Folder with application code



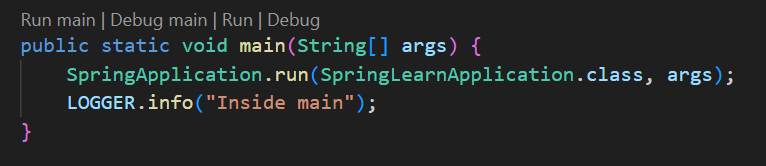
1. src/main/resources - Folder for application configuration



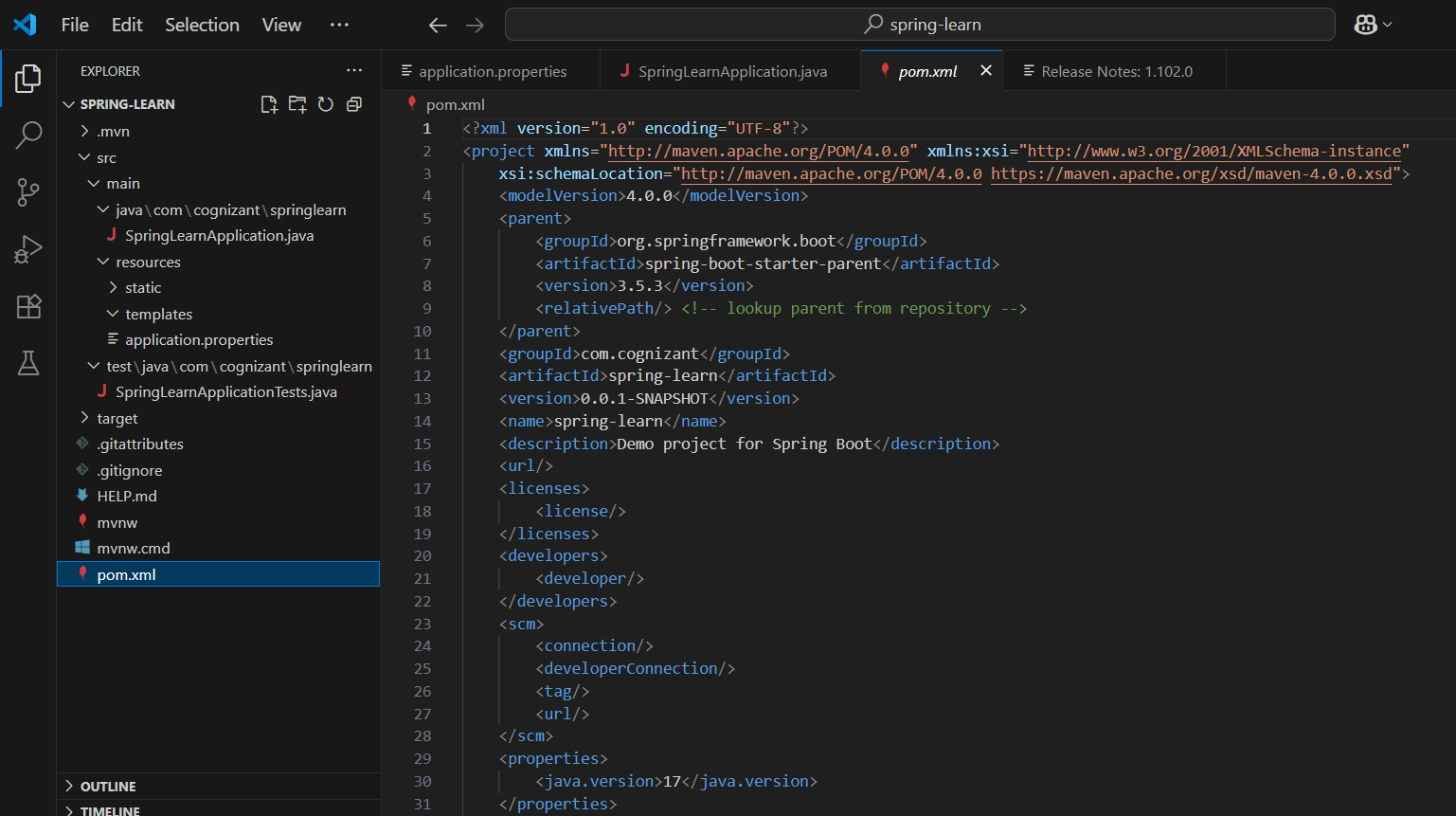
1. src/test/java - Folder with code for testing the application

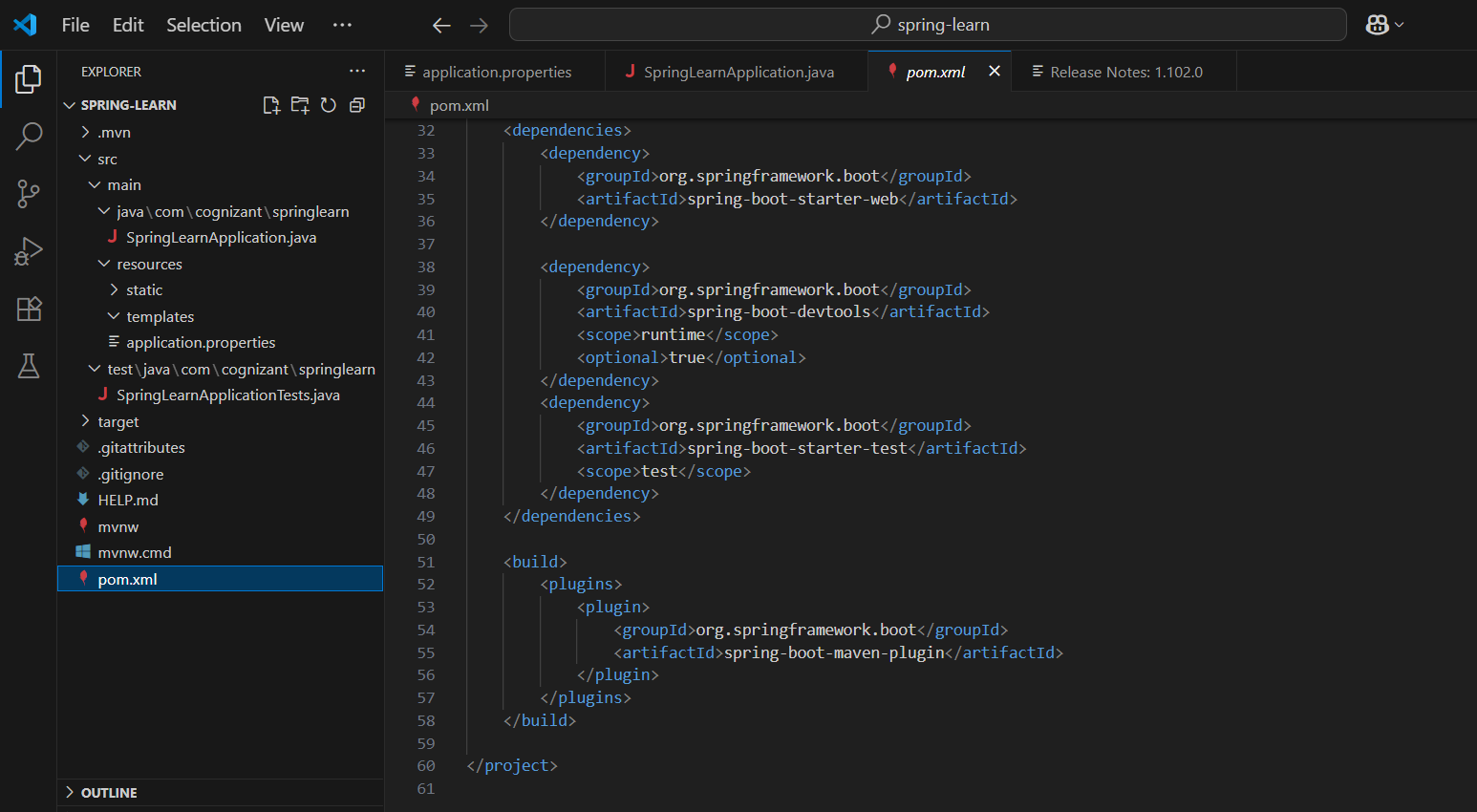


1. SpringLearnApplication.java - Walkthrough the main() method.

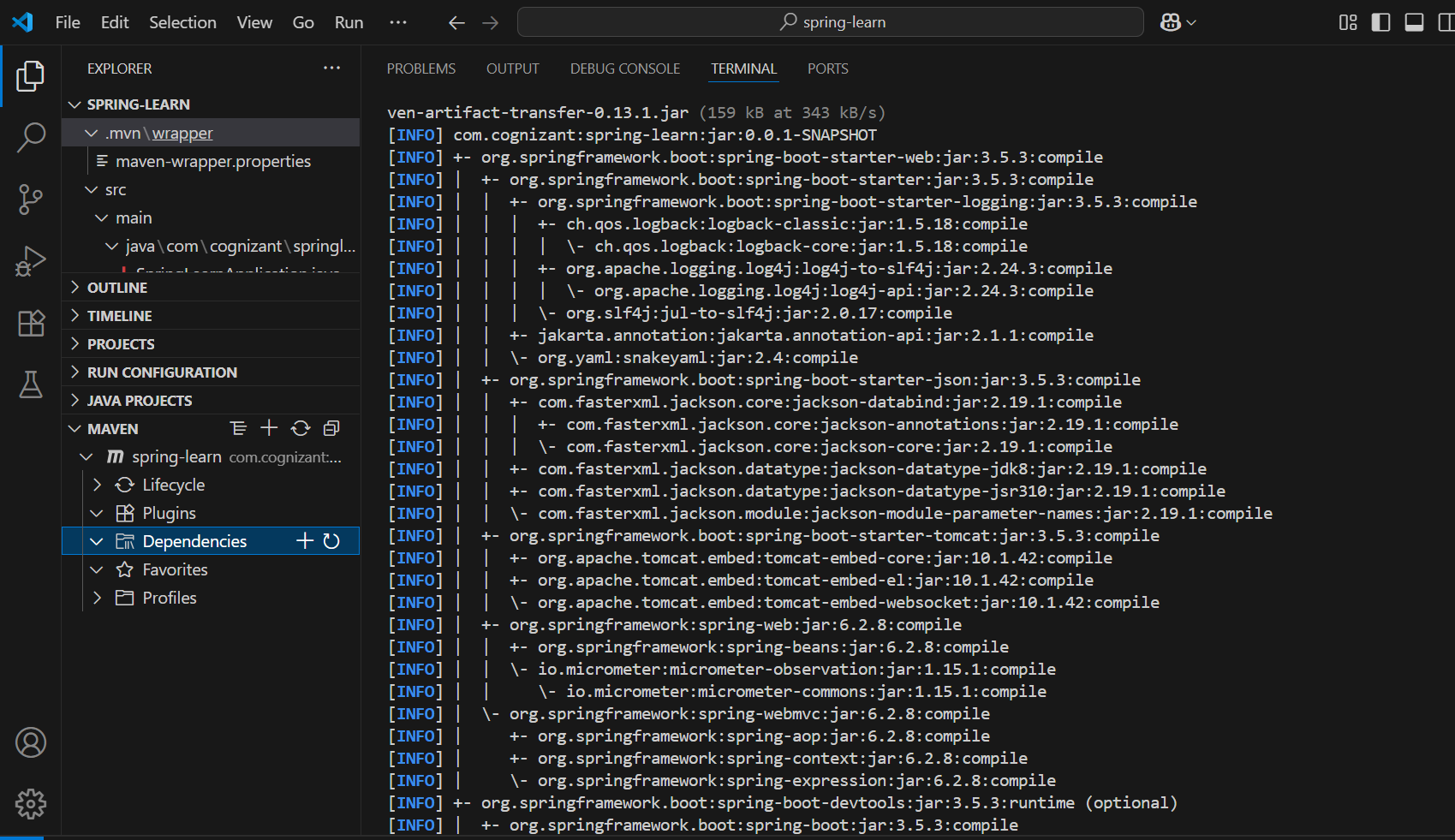
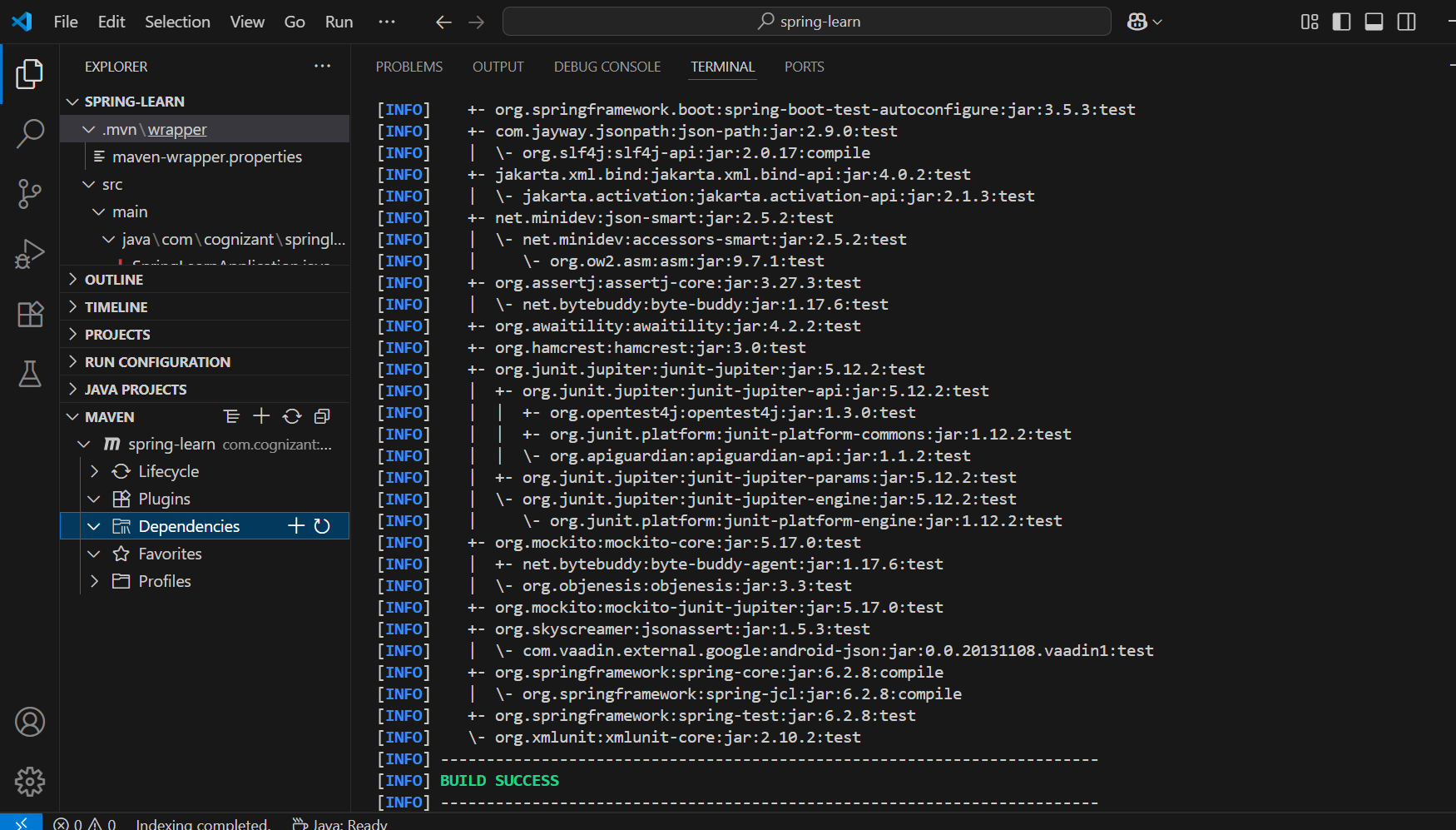


1. Purpose of @SpringBootApplication annotation
2. pom.xml:
   1. Walkthrough all the configuration defined in XML file





* 1. Open 'Dependency Hierarchy' and show the dependency tree.

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

spring-learn

├──.mvn

├── src

│ ├── main

│ │ ├── java

│ │ │ └── com

│ │ │ └── cognizant

│ │ │ └── springlearn

│ │ │ ├── Country.java

│ │ │ └── SpringLearnApplication.java

│ │ └── resources

│ │ └── country.xml

│ └── logback.xml

│

├── pom.xml

**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.springlearn.Country">

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

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

    </bean>

</beans>

**logback.xml:**

<configuration>

    <root level="DEBUG">

        <appender-ref ref="STDOUT"/>

    </root>

    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">

        <encoder>

            <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>

        </encoder>

    </appender>

</configuration>

**Country.java:**

package com.cognizant.springlearn;

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("Getting Code: {}", code);

        return code;

    }

    public void setCode(String code) {

        LOGGER.debug("Setting Code: {}", code);

        this.code = code;

    }

    public String getName() {

        LOGGER.debug("Getting Name: {}", name);

        return name;

    }

    public void setName(String name) {

        LOGGER.debug("Setting Name: {}", name);

        this.name = name;

    }

    @Override

    public String toString() {

        return "Country [code=" + code + ", name=" + name + "]";

    }

}

**SpringLearnApplication.java**

package com.cognizant.springlearn;

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 main(String[] args) {

        LOGGER.info("START");

        displayCountry();

        LOGGER.info("END");

    }

    public static void displayCountry() {

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

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

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

    }

}

**🔹 <bean> tag**

Defines a bean (object) that Spring will manage.

**🔹 id attribute**

A unique name by which the bean will be accessed (e.g., "country").

**🔹 class attribute**

Specifies the fully qualified class name of the bean (e.g., com.cognizant.springlearn.Country).

**🔹 <property> tag**

Sets a value for a property of the bean.

**🔹 name attribute (in <property>)**

Specifies the property name (must match the setter method).

**🔹 value attribute**

The value to assign to the property.

**What is ApplicationContext?**

It is the central interface to provide configuration for a Spring application. It is used to load bean definitions and wires them together.

**ClassPathXmlApplicationContext**

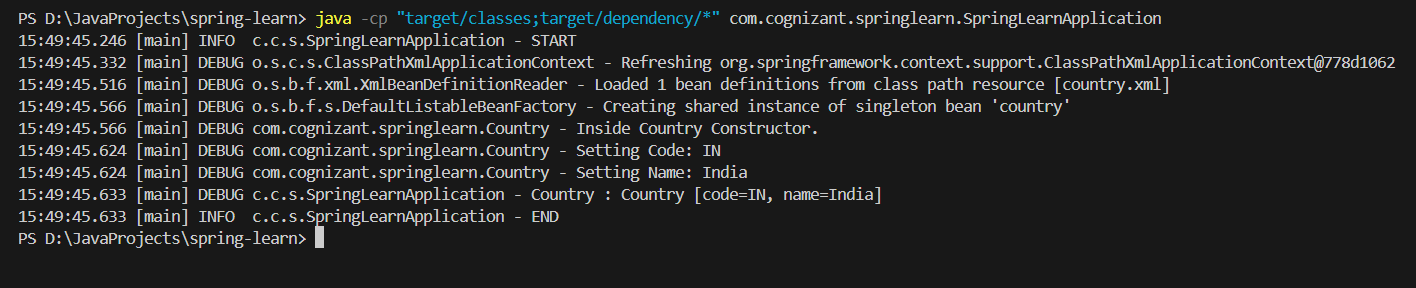
A concrete implementation of ApplicationContext that loads context definitions from an XML file located in the classpath.

**What Happens When context.getBean() is Called?**

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

This does:

1. Spring searches for the bean with ID "country" in the loaded XML.
2. It creates an instance of the specified class Country.
3. It injects values into the code and name properties using setters.
4. It returns the fully initialized bean object.

**Output:  
**