**Week 4 : Spring REST using Spring Boot**

**Hands on 1**

**Create a Spring Web Project using Maven**

**Code :**

**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) {

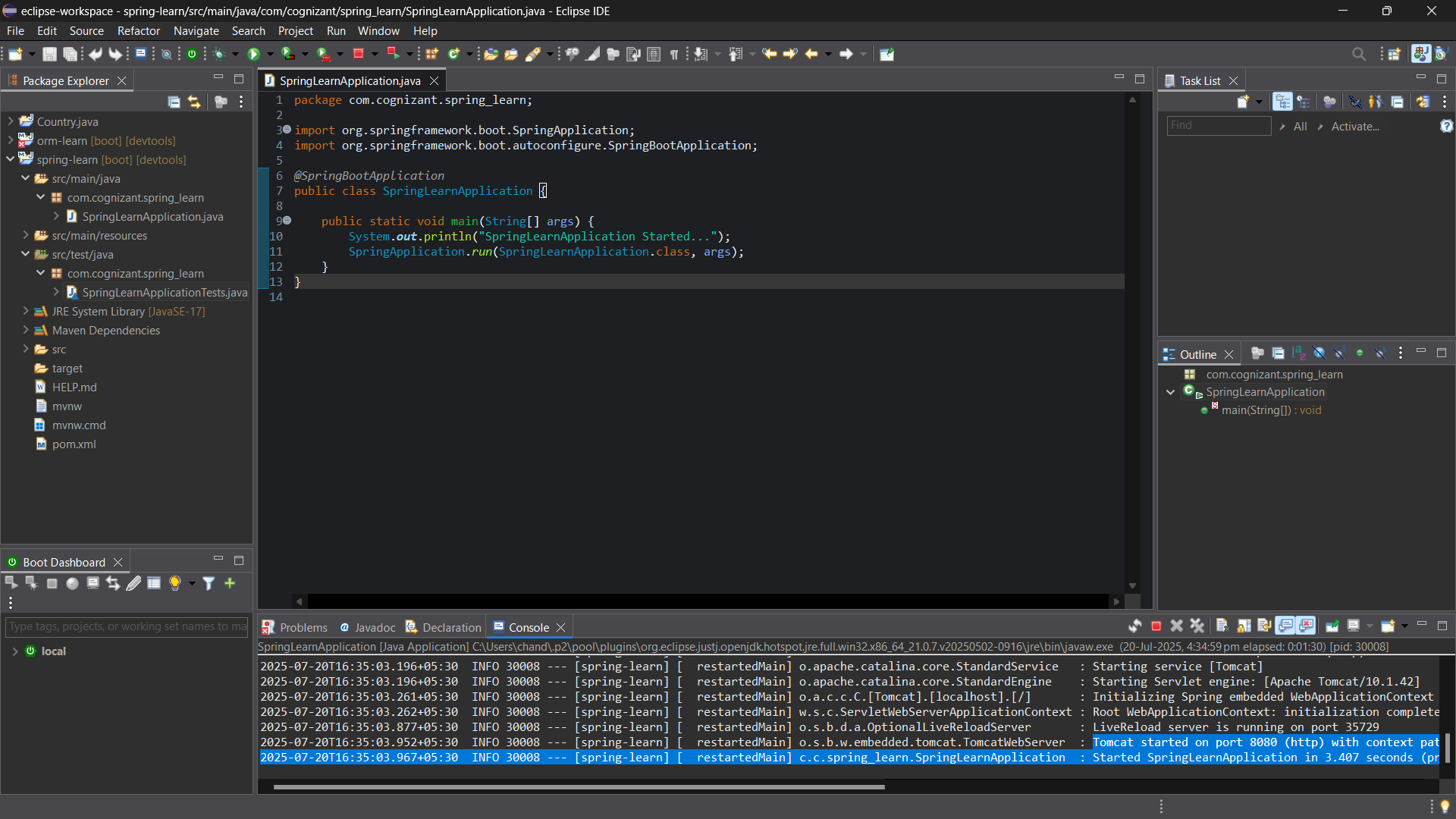
System.out.println("SpringLearnApplication Started...");

SpringApplication.run(SpringLearnApplication.class, args);

    }

}

**Output :**



**Hands on 2**

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

**Code :**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.text.SimpleDateFormat;

import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) throws Exception {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate(); // call the method after Spring Boot starts

}

public static void displayDate() throws Exception {

ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

Date date = format.parse("31/12/2018");

System.out.println("Parsed Date: " + date);

}

}

**application.properties**

spring.application.name=spring-learn

server.port=8080

**dateformat.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">

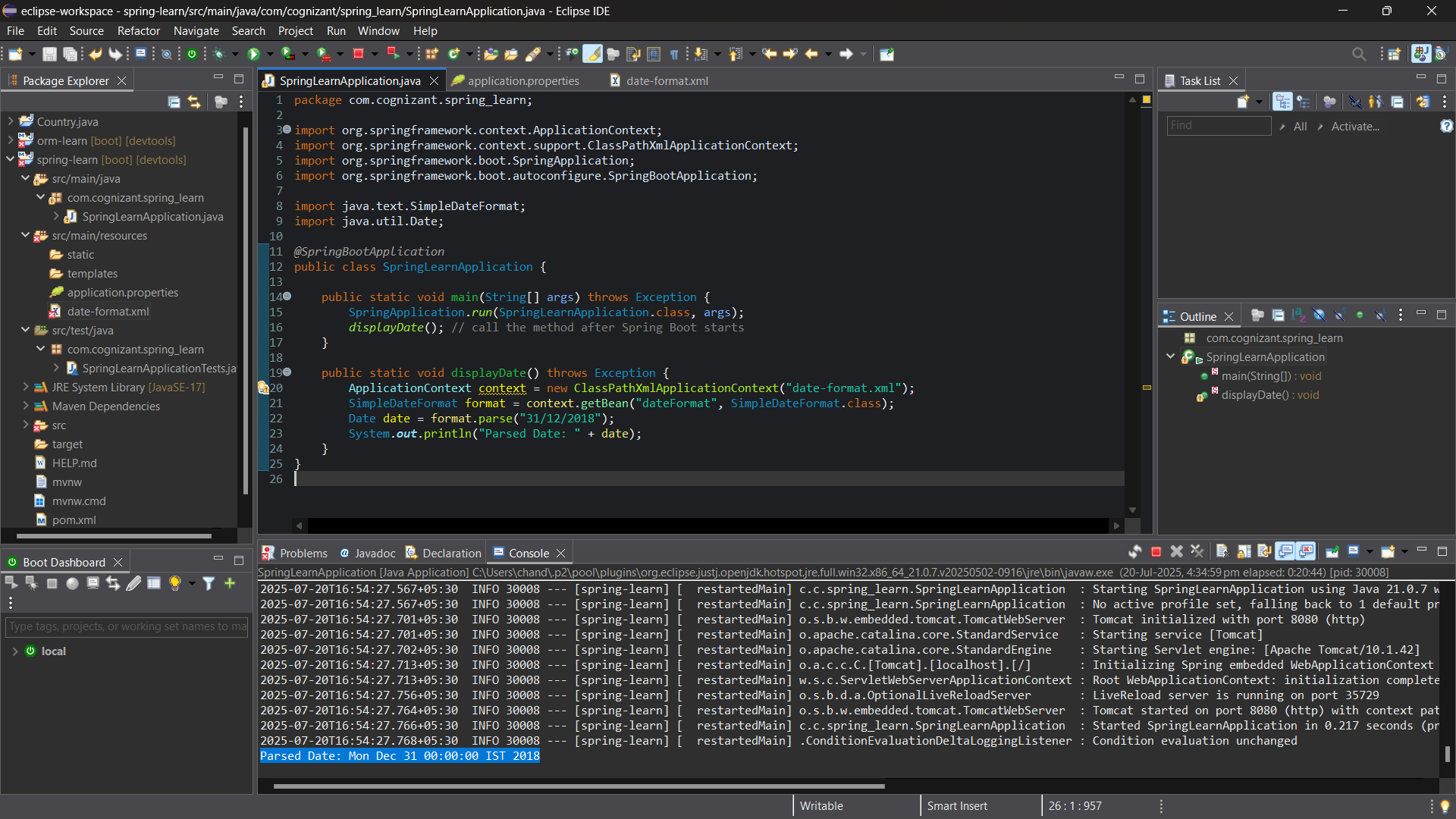
<bean id="dateFormat" class="java.text.SimpleDateFormat">

<constructor-arg value="dd/MM/yyyy"/>

</bean>

</beans>

**Output :**



**Hands on 3**

**Spring Core - Incorporate Logging**

**Code :**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.text.SimpleDateFormat;

import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) throws Exception {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate(); // call the method after Spring Boot starts

}

public static void displayDate() throws Exception {

ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

Date date = format.parse("31/12/2018");

System.out.println("Parsed Date: " + date);

}

}

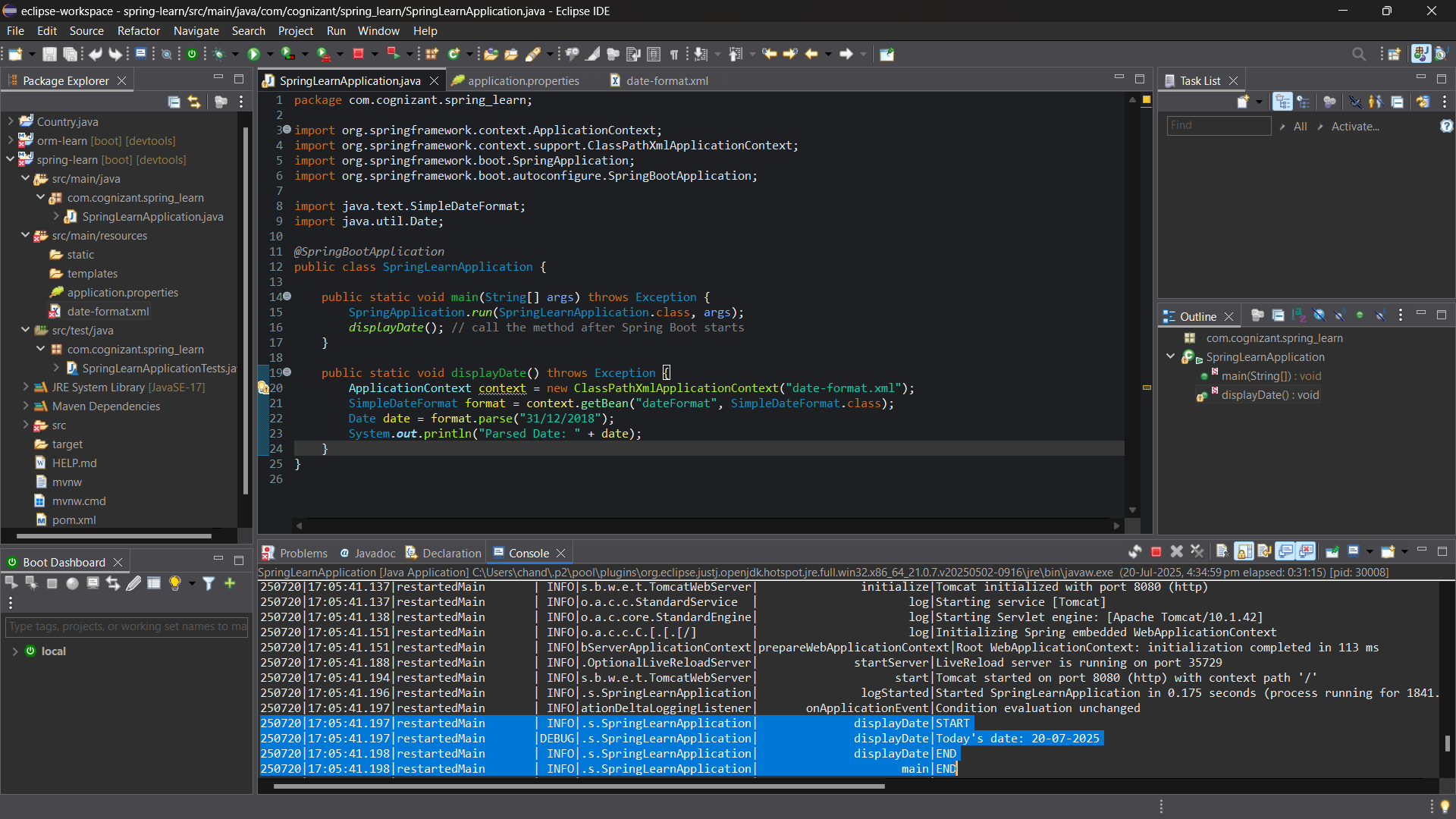
**application.properties**

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n

**Output :**



**Hands on 4**

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

**Code :**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

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) {

System.setProperty("debug", "true"); // Enables debug logging explicitly

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

}

}

**Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String name;

private String code;

// Getters and Setters

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

@Override

public String toString() {

return "Country{name='" + name + "', code='" + code + "'}";

}

}

**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">

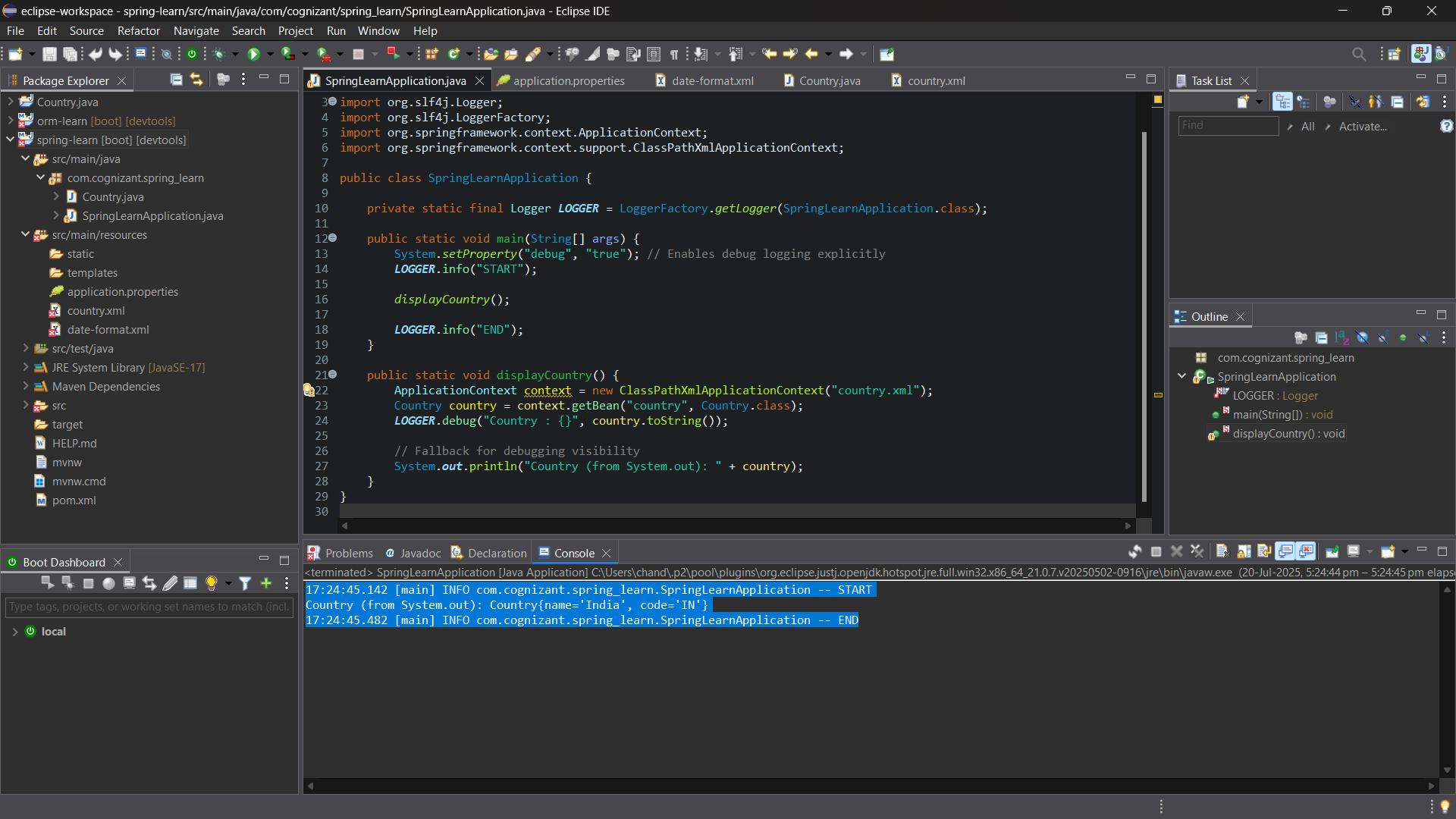
<property name="name" value="India"/>

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

</bean>

</beans>

**Output :**



**Hands on 5**

**Spring Core – Demonstration of Singleton Scope and Prototype Scope**

**Code :**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

public static void main(String[] args) {

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

System.out.println("START");

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

System.out.println(country);

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

System.out.println(anotherCountry);

System.out.println("END");

}

}

**Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String name;

private String code;

public Country() {

System.out.println("Constructor Called for Country");

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

@Override

public String toString() {

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

}

}

**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" scope="prototype">

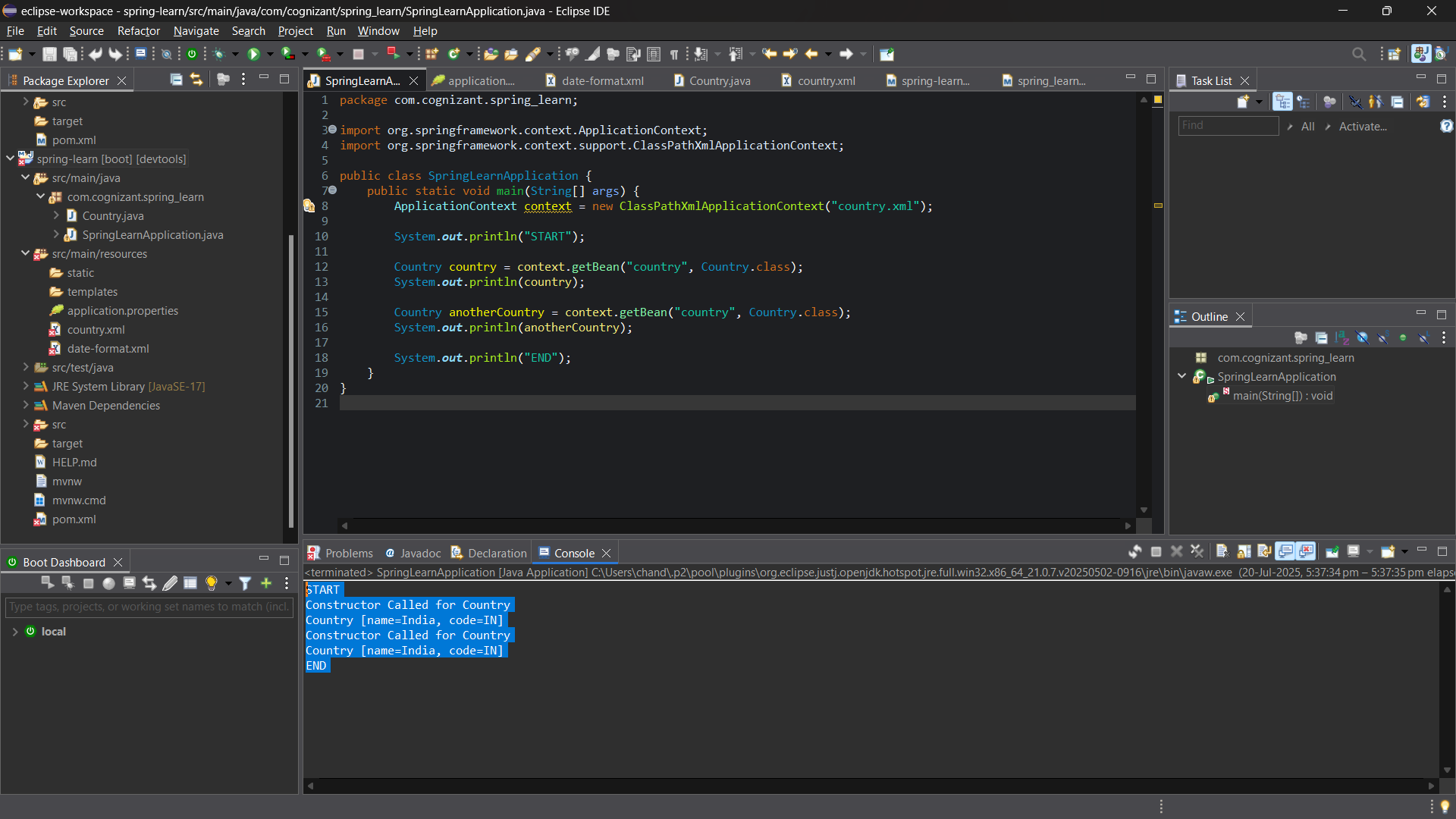
<property name="name" value="India"/>

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

</bean>

</beans>

**Output :**



**Hands on 6**

**Spring Core – Load list of countries from Spring Configuration XML**

**Code :**

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

public static void main(String[] args) {

System.out.println("START main");

displayCountries();

System.out.println("END main");

}

public static void displayCountries() {

System.out.println("START displayCountries");

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

@SuppressWarnings("unchecked")

List<Country> countryList = (List<Country>) context.getBean("countryList");

for (Country country : countryList) {

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

}

System.out.println("END displayCountries");

}

}

**Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {

System.out.println("Country constructor called");

}

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;

}

@Override

public String toString() {

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

}

}

**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">

<!-- Individual Country Beans -->

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

<!-- List of Country Beans -->

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="in"/>

<ref bean="us"/>

<ref bean="de"/>

<ref bean="jp"/>

</list>

</constructor-arg>

</bean>

</beans>

**Output :**

