**WEEK – 04**

**Spring REST using Spring Boot 3**

**Superset ID: 6262264**

**Exercise 2:**

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

**SOLUTION:**

* In this hands-on, I define a SimpleDateFormat bean with the pattern dd/MM/yyyy inside the date-format.xml configuration file in src/main/resources of the spring-learn project.
* In the SpringLearnApplication class, I create an ApplicationContext to load this XML configuration.
* I then retrieve the SimpleDateFormat bean using the getBean() method and use it to parse the string 31/12/2018 into a Date object.
* Finally, I display the parsed date in the console.
* This approach helps me avoid creating SimpleDateFormat instances in multiple places and demonstrates dependency injection using Spring Core in VS Code.

***Country.java:***

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

    private String code;

    private String name;

    public Country() {

        Logger logger = LoggerFactory.getLogger(Country.class);

        logger.debug("Inside Country Constructor.");

    }

    public String getCode() {

        Logger logger = LoggerFactory.getLogger(Country.class);

        logger.debug("Inside getCode");

        return code;

    }

    public void setCode(String code) {

        Logger logger = LoggerFactory.getLogger(Country.class);

        logger.debug("Inside setCode");

        this.code = code;

    }

    public String getName() {

        Logger logger = LoggerFactory.getLogger(Country.class);

        logger.debug("Inside getName");

        return name;

    }

    public void setName(String name) {

        Logger logger = LoggerFactory.getLogger(Country.class);

        logger.debug("Inside setName");

        this.name = name;

    }

    @Override

    public String toString() {

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

    }

}

***Purpose:***

The purpose of this code is to define a simple Country Java bean with code and name properties, along with getter and setter methods that include debug logging to trace when the fields are accessed or modified.

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

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

}

}

***Purpose:***

The purpose of this code is to load the Country bean from the country.xml Spring configuration file using ClassPathXmlApplicationContext, retrieve its details, and print them to the console while logging the start and end of the application.

***SpringLearnApplicationTests.Java:***

package com.cognizant.springlearn;

import org.junit.jupiter.api.Test;

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

@SpringBootTest

class SpringLearnApplicationTests {

    @Test

    void contextLoads() {

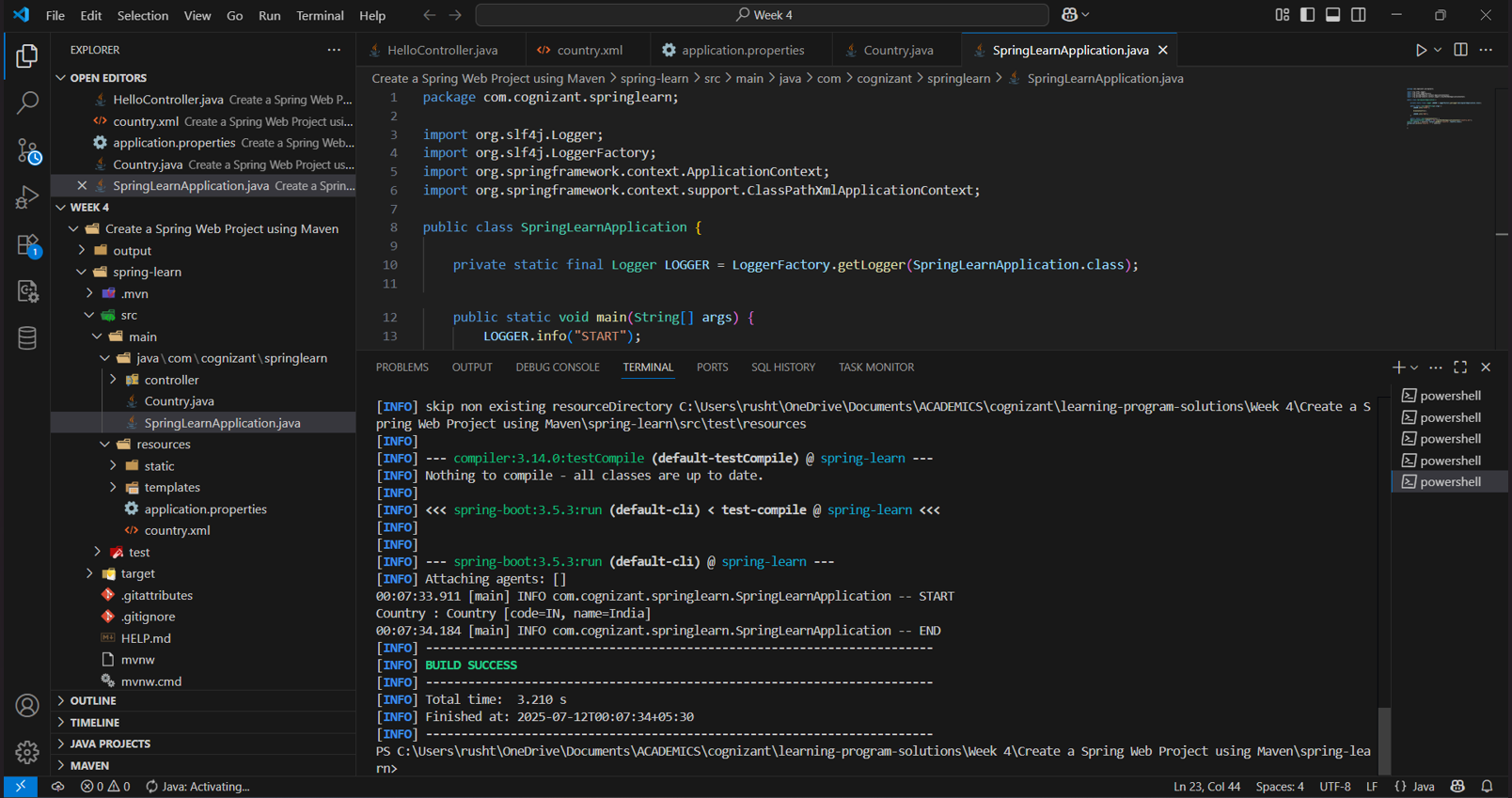
    }

}

***Purpose:***

The purpose of this code is to verify that the Spring application context loads correctly without any configuration errors during testing.

***OUTPUT:***

******

***EXPLANATION:***

* The Country class defines a simple Java bean with code and name properties, including getters, setters, and debug logging to trace object creation and property access.
* The SpringLearnApplication class loads the Country bean from the country.xml configuration file using ClassPathXmlApplicationContext and prints its details to the console with lifecycle logs.
* The SpringLearnApplicationTests class ensures the Spring application context loads successfully without errors during testing.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**