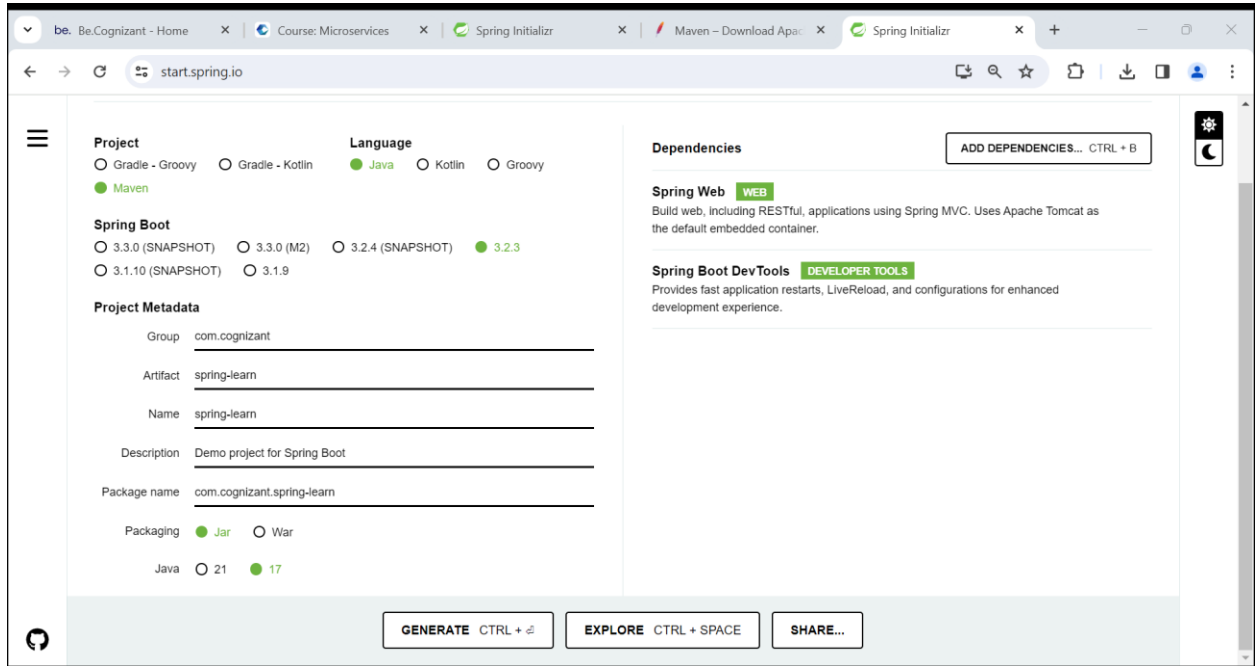


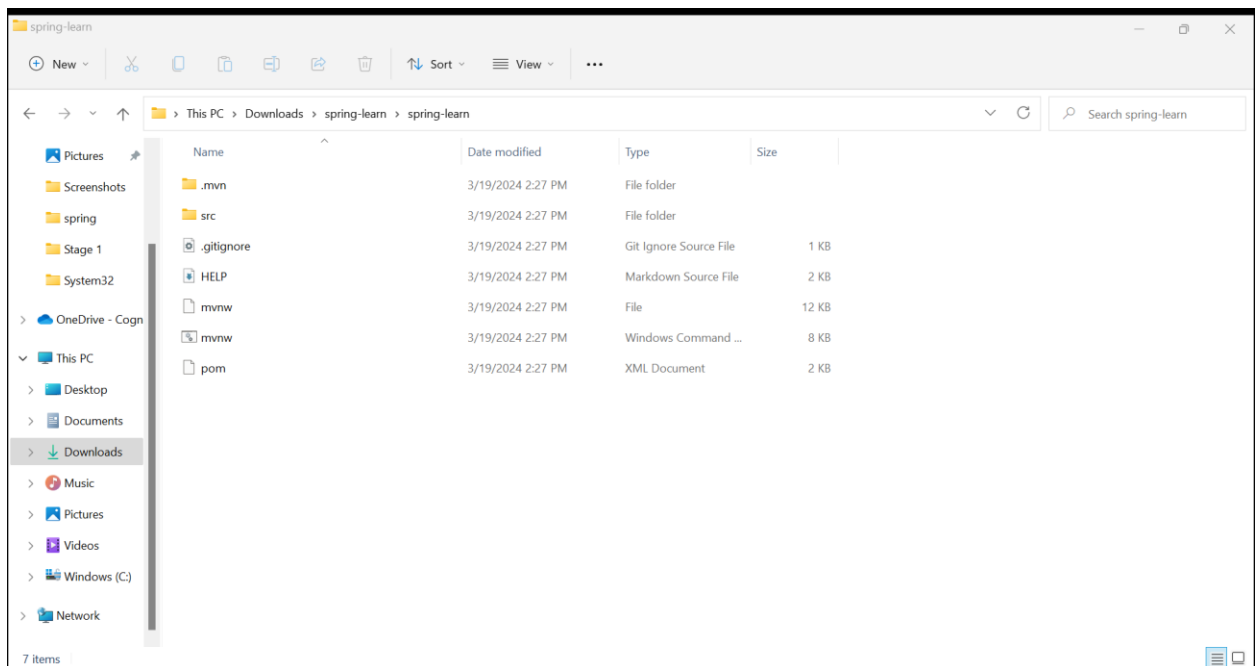
Hands on 1

Create a Spring Web Project using Maven

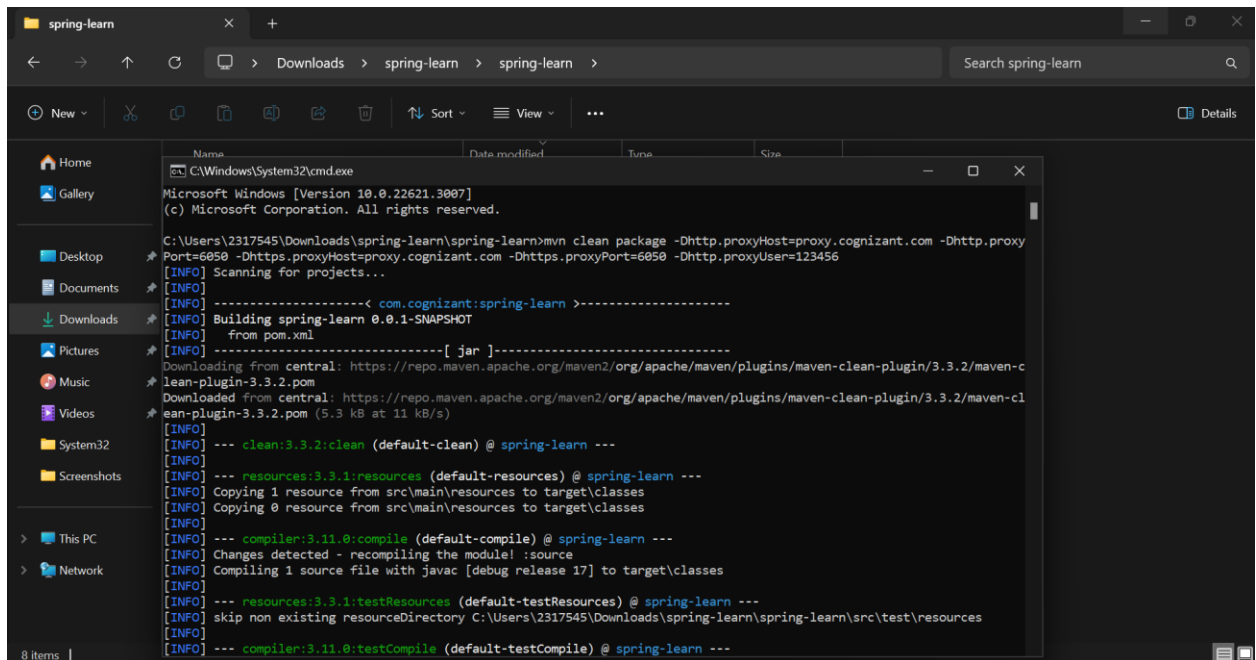
1. Created one spring learn maven project with the help of Spring Initializer



2. Downloaded the project as zip and extract the zip in root folder to Eclipse Workspace



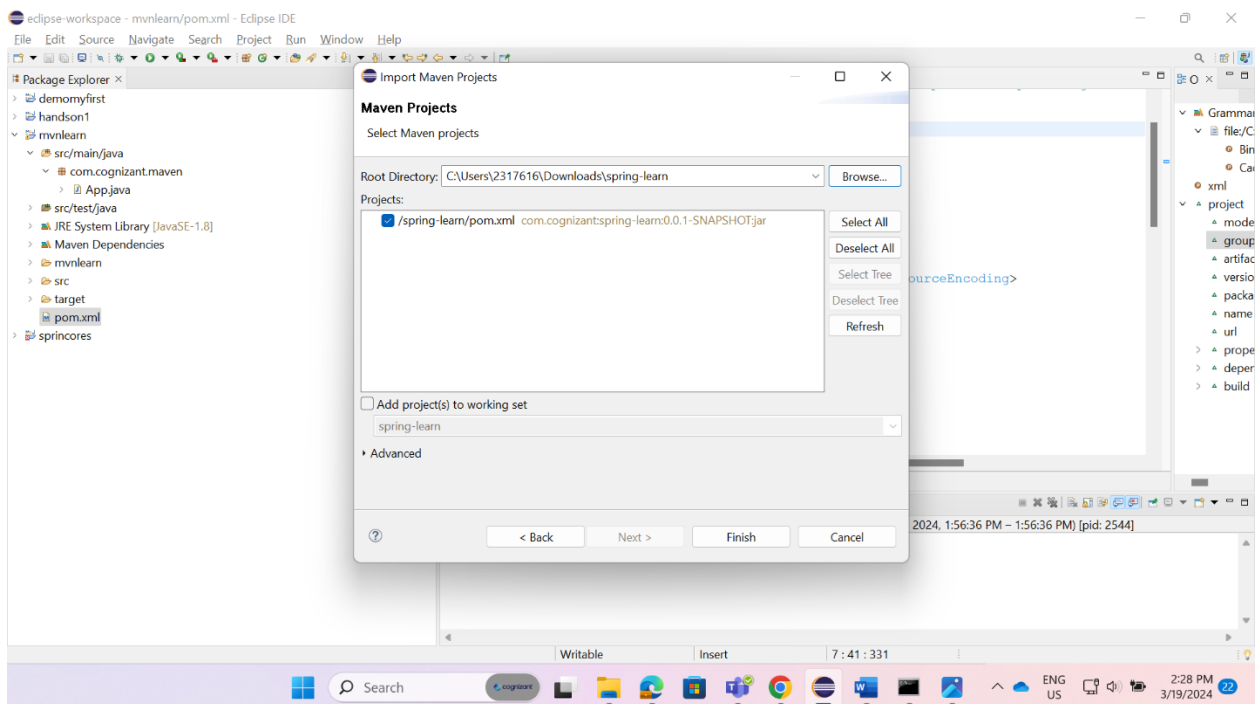
3. Build the project using given command on command line



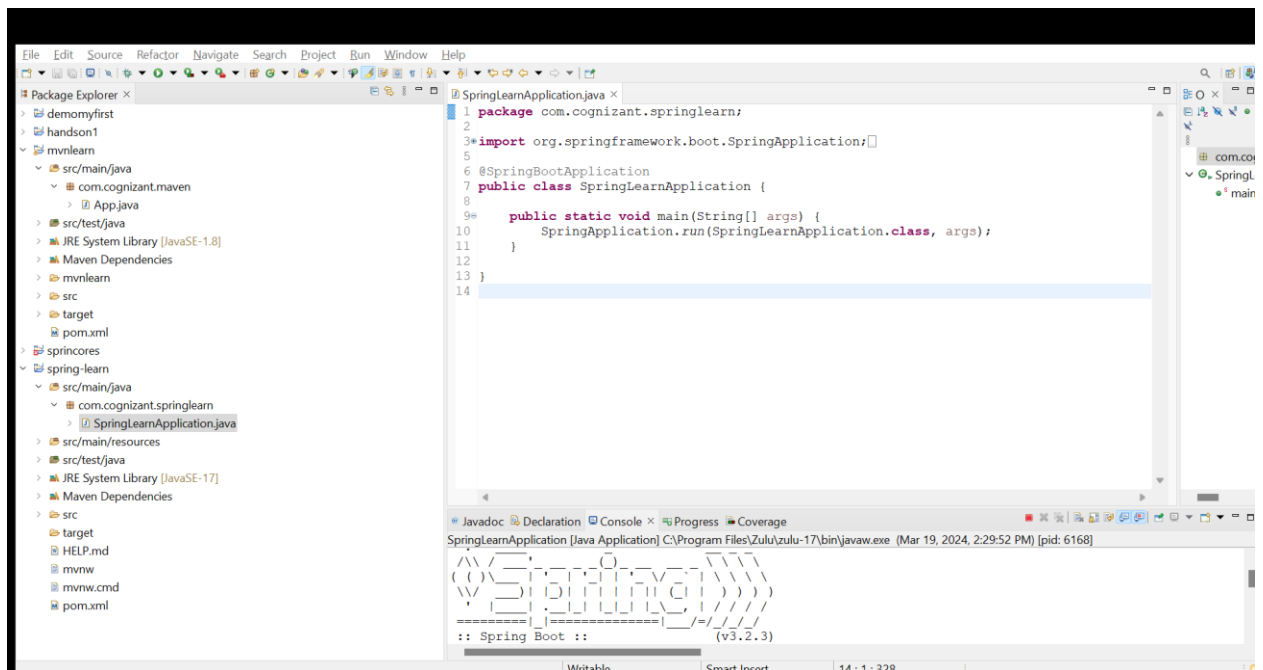
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\2317545\Downloads\spring-learn\spring-learn>mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxy
Port=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.cognizant:spring-learn >-----
[INFO] Building spring-learn 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.3.2/maven-c
lean-plugin-3.3.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.3.2/maven-cl
ean-plugin-3.3.2.pom (5.3 kB at 11 kB/s)
[INFO]
[INFO] --- clean:3.3.2:clean (default-clean) @ spring-learn ---
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ spring-learn ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ spring-learn ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 1 source file with javac [debug release 17] to target/classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ spring-learn ---
[INFO] skip non existing resourceDirectory C:\Users\2317545\Downloads\spring-learn\spring-learn\src\test\resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ spring-learn ---
```

4. Imported the project in Eclipse IDE



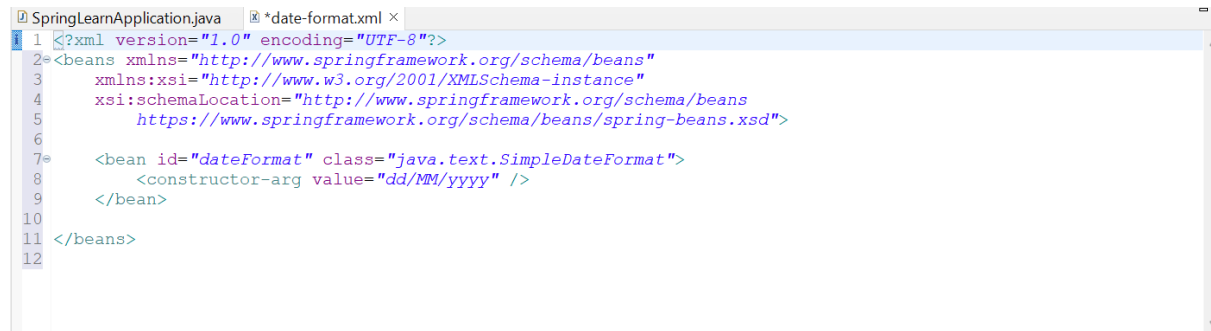
5. Run the SpringLearnApplication class.



Hands on 2

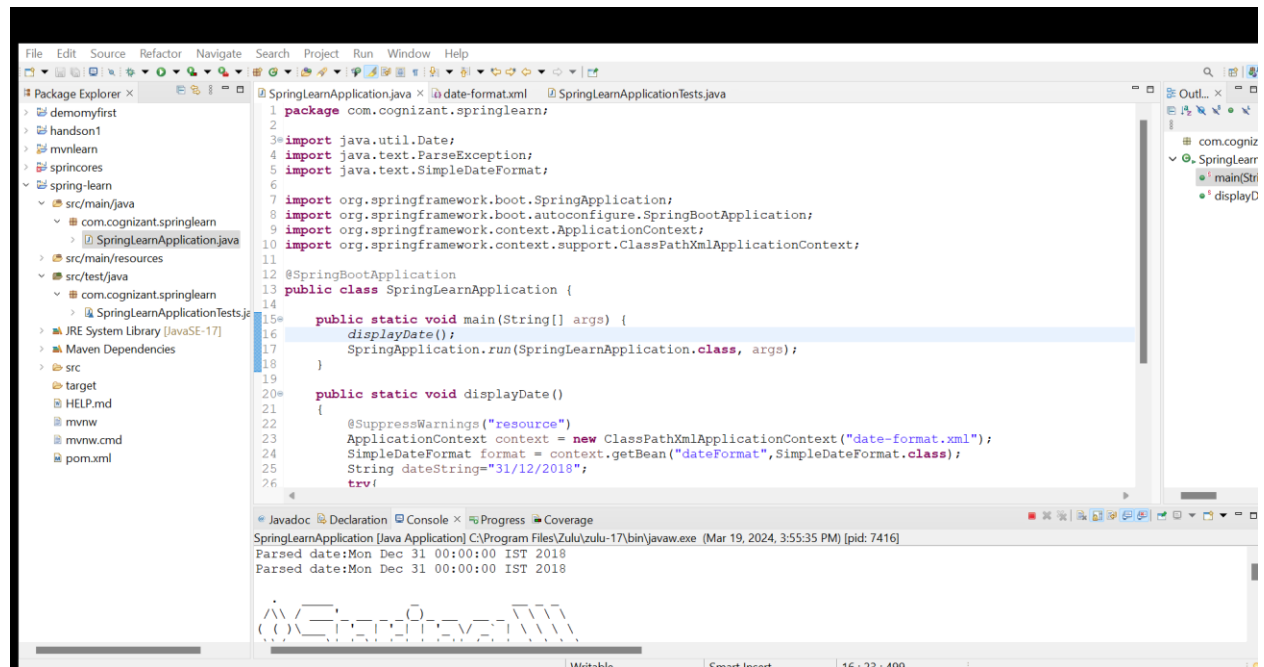
Spring Core – Load SimpleDateFormat from Spring Configuration XML

1. Created the date-format.xml file.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans
5                           https://www.springframework.org/schema/beans/spring-beans.xsd">
6
7     <bean id="dateFormat" class="java.text.SimpleDateFormat">
8         <constructor-arg value="dd/MM/yyyy" />
9     </bean>
10
11 </beans>
12
```

2. Created new method `displayDate()` in `SpringLearnApplication.java`. In `displayDate()` method create the `ApplicationContext`.
3. Get the `dateFormat` using `getBean()` method. Using the format variable try to parse string '31/12/2018' to `Date` class and display the result using `System.out.println`.
4. Run the application as 'Java Application' and checked the result in console log output.



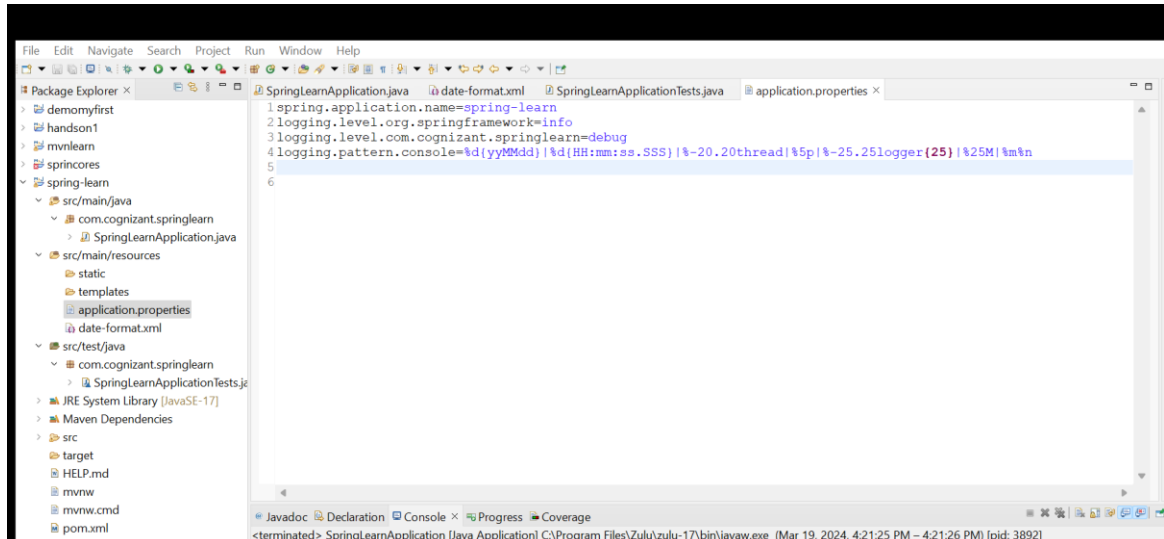
```
1 package com.cognizant.springlearn;
2
3 import java.util.Date;
4 import java.text.ParseException;
5 import java.text.SimpleDateFormat;
6
7 import org.springframework.boot.SpringApplication;
8 import org.springframework.boot.autoconfigure.SpringBootApplication;
9 import org.springframework.context.ApplicationContext;
10 import org.springframework.context.support.ClassPathXmlApplicationContext;
11
12 @SpringBootApplication
13 public class SpringLearnApplication {
14
15     public static void main(String[] args) {
16         displayDate();
17         SpringApplication.run(SpringLearnApplication.class, args);
18     }
19
20     public static void displayDate()
21     {
22         @SuppressWarnings("resource")
23         ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
24         SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);
25         String dateString="31/12/2018";
26         try{
27
28         }
29     }
30 }
```

SpringLearnApplication [Java Application] C:\Program Files\Zulu\zulu-17\bin\javaw.exe (Mar 19, 2024, 3:55:35 PM) [pid: 7416]
Parsed date: Mon Dec 31 00:00:00 IST 2018
Parsed date: Mon Dec 31 00:00:00 IST 2018

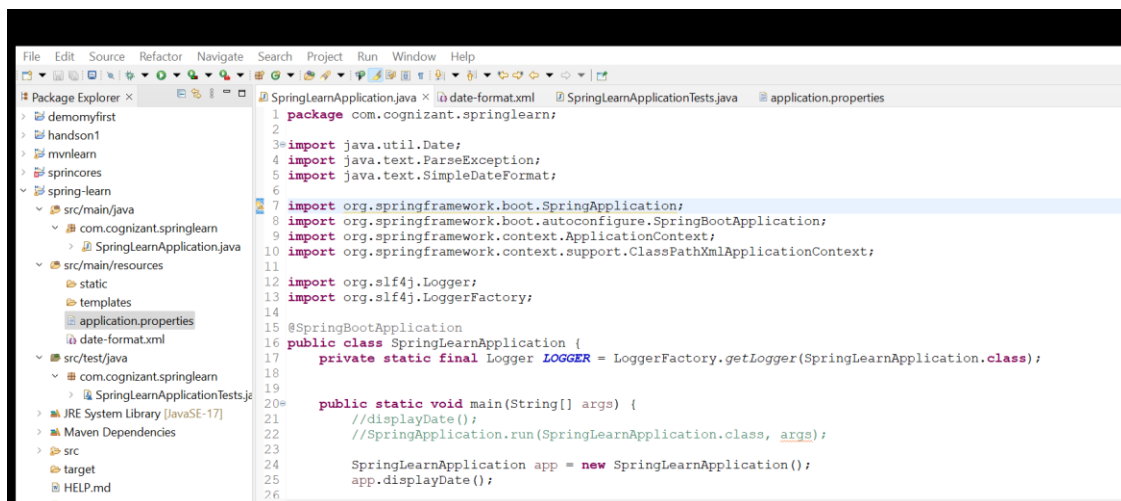
Hands on 3

Spring Core - Incorporate Logging

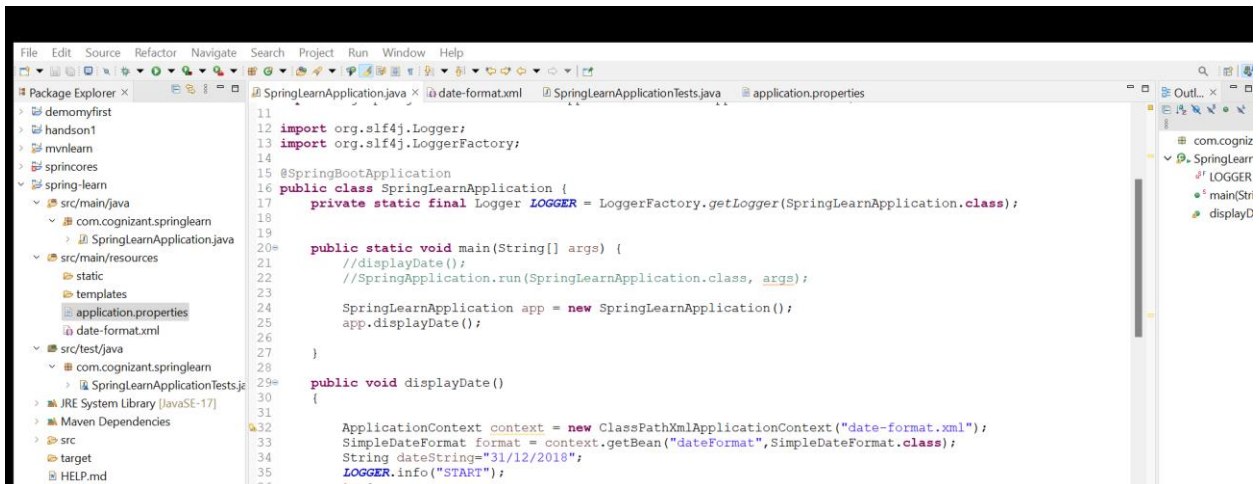
1. Created **application.properties** in **src/main/resources** folder



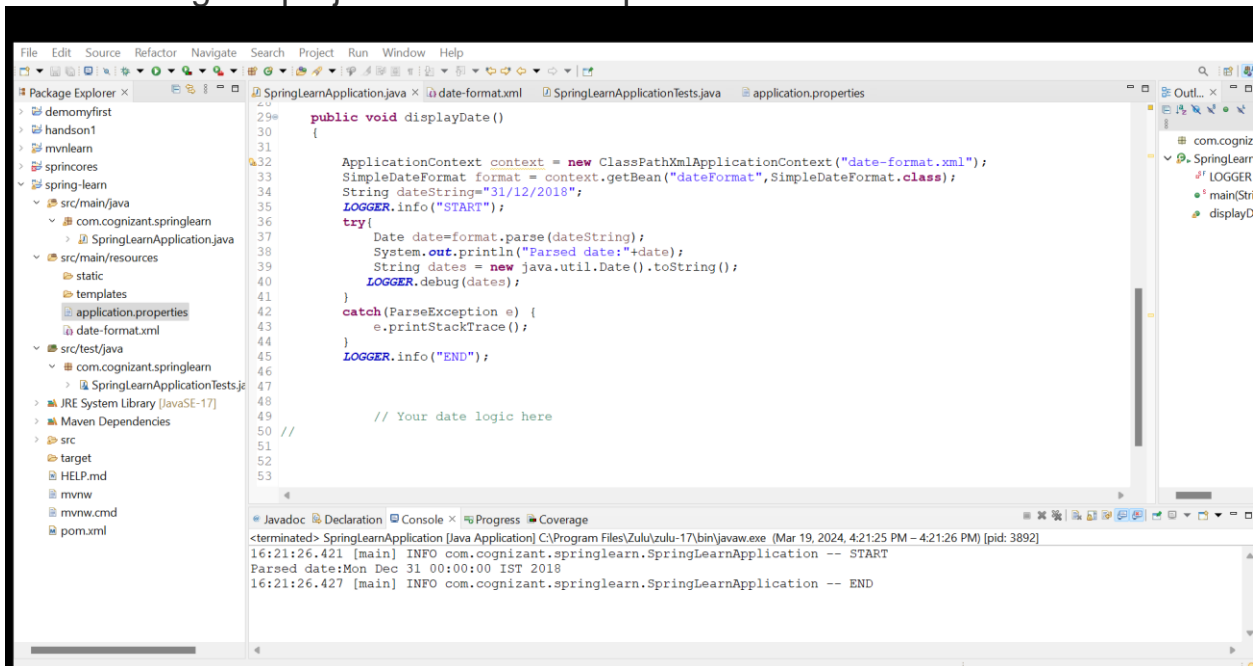
2. In **SpringLearnApplication.java** include the following imports



3. Included the static variable in **SpringLearnApplication.java**:



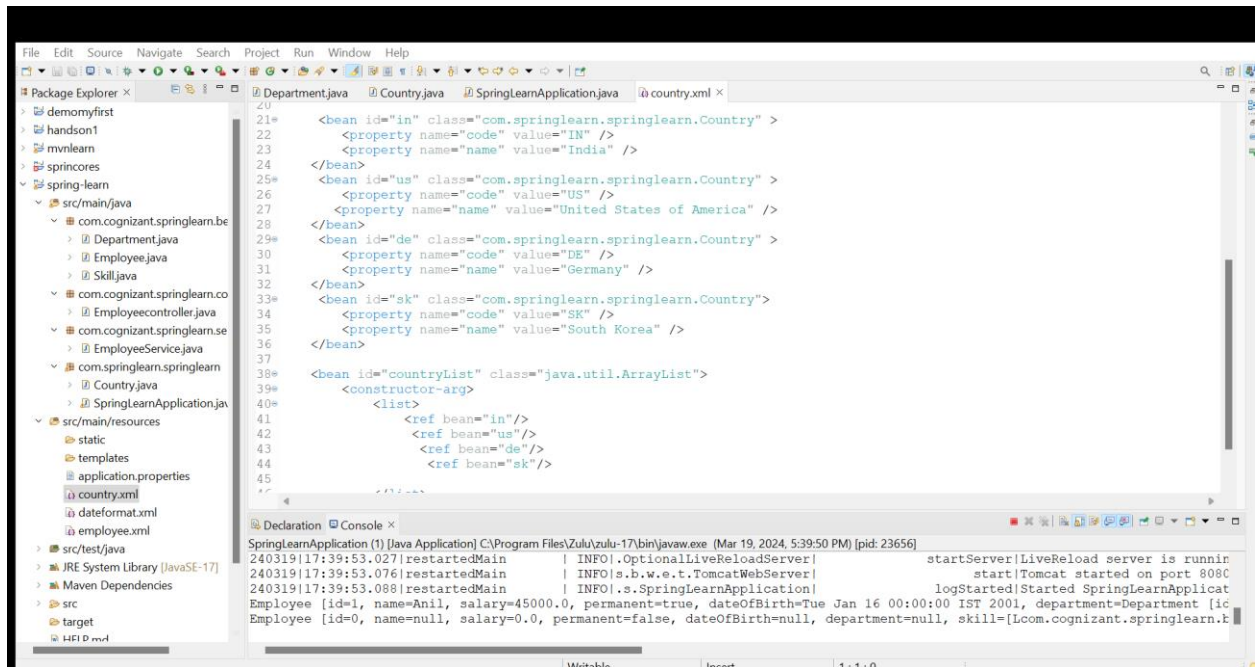
4. After running the project shown the output.



Hands on 4

Spring Core – Load Country from Spring Configuration XML

1. Created the country.xml file and included the requirement code in that.

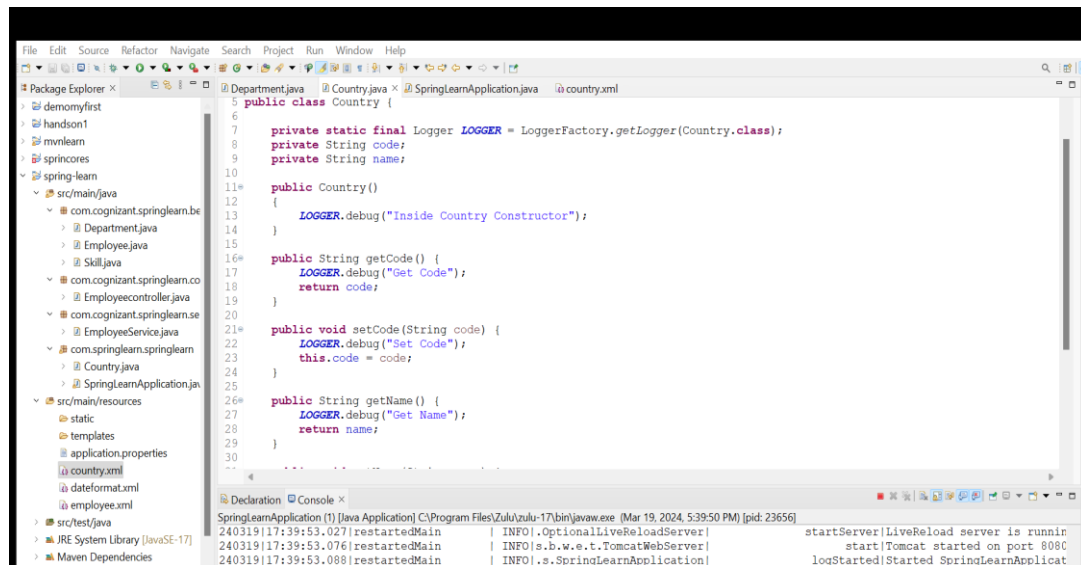


The screenshot shows an IDE with the `country.xml` file open. The XML content defines four `Country` beans and a `countryList` bean.

```
<?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/schemas/beans/spring-beans-4.3.xsd">
    <bean id="in" class="com.springlearn.springlearn.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>
    <bean id="us" class="com.springlearn.springlearn.Country">
        <property name="code" value="US"/>
        <property name="name" value="United States of America"/>
    </bean>
    <bean id="de" class="com.springlearn.springlearn.Country">
        <property name="code" value="DE"/>
        <property name="name" value="Germany"/>
    </bean>
    <bean id="sk" class="com.springlearn.springlearn.Country">
        <property name="code" value="SK"/>
        <property name="name" value="South Korea"/>
    </bean>
    <bean id="countryList" class="java.util.ArrayList">
        <constructor-arg>
            <list>
                <ref bean="in"/>
                <ref bean="us"/>
                <ref bean="de"/>
                <ref bean="sk"/>
            </list>
        </constructor-arg>
    </bean>
</beans>
```

The console output shows the application running successfully, with logs for the Spring application and Tomcat server.

2. Create **Country** class with following aspects:



The screenshot shows the `Country.java` class in the IDE. The code defines a `Country` class with a logger, private fields for `code` and `name`, and public methods for `getCode`, `setCode`, and `getName`.

```
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("Get Code");
        return code;
    }

    public void setCode(String code) {
        LOGGER.debug("Set Code");
        this.code = code;
    }

    public String getName() {
        LOGGER.debug("Get Name");
        return name;
    }
}
```

The console output shows the application running successfully, with logs for the Spring application and Tomcat server.

- Created a method `displayCountry()` in `SpringLearnApplication.java`, which will read the country bean from spring configuration file and display the country details.
- Invoked `displayCountry()` method in `main()` method of `SpringLearnApplication.java`.

```
37 // try {
38 //     Date date = format.parse("31/12/2018");
39 //     System.out.println(date);
40 // } catch (Exception e) {
41 //     e.printStackTrace();
42 // }
43 // }
44
45 public static void displayCountry()
46 {
47     ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
48
49     List<Country> countries=(List<Country>) context.getBean("countryList");
50     for(Country country : countries) {
51         LOGGER.debug("Country : {}", country.toString());
52     }
53 }
54
55 /*public static void displayDate()
56 {
57     LOGGER.info("START");
58     ApplicationContext context = new ClassPathXmlApplicationContext("dateformat.xml");
59     SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);
60     try {
61         Date date = format.parse("31/12/2018");
62         LOGGER.debug(date.toString());
63         //System.out.println(date);
64     } catch (Exception e) {
65         e.printStackTrace();
66     }
67     LOGGER.info("END");
68 } */
69
70 public static void main(String[] args) {
71
72
```

- Run the application and shown the output.

```
55
56
57
58 private static void displayCountries() {
59     ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
60     ArrayList<Country> countryList = context.getBean("countryList", ArrayList.class);
61
62     LOGGER.debug("List of Countries:");
63     for (Country country : countryList) {
64         LOGGER.debug(country.toString());
65     }
66 }
67
68 public static void main(String[] args) {
69
70
```

Console

SpringLearnApplication [Java Application] C:\Program Files\Zulu\zulu-17\bin\javaw.exe (Mar 20, 2024, 12:01:03 PM) [pid: 19616]

Description:

Web server failed to start. Port 8080 was already in use.

Action:

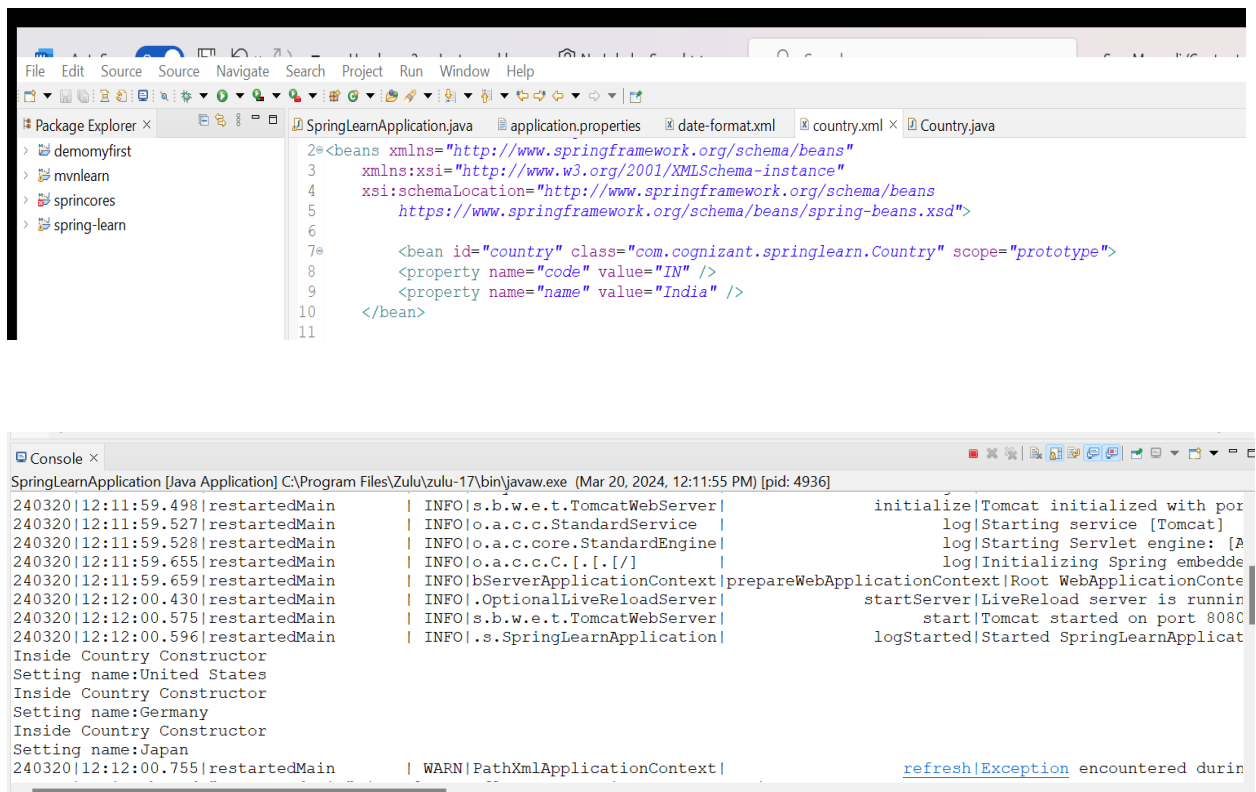
Identify and stop the process that's listening on port 8080 or configure this application to listen on another port.

Inside Country Constructor
Setting name:India
Inside Country Constructor
Setting name:United States
Inside Country Constructor
Setting name:Germany
Inside Country Constructor
Setting name:Japan
240320|12:03:51.109|restartedMain | WARN|PathXmlApplicationContext| refresh|Exception encountered durin

Hands on 5

Spring Core – Demonstration of Singleton Scope and Prototype Scope

1. Run the application
2. Constructor will be called twice, which means that two instances of country is created.



The screenshot displays an IDE with two windows. The top window shows the XML configuration for a Spring application, and the bottom window shows the console output.

XML Configuration (country.xml):

```
1<?xml version="1.0" encoding="UTF-8"?>
2<beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://www.springframework.org/schema/beans
5                          https://www.springframework.org/schema/beans/spring-beans.xsd">
6
7      <bean id="country" class="com.cognizant.springlearn.Country" scope="prototype">
8          <property name="code" value="IN" />
9          <property name="name" value="India" />
10     </bean>
11</beans>
```

Console Output:

```
SpringLearnApplication [Java Application] C:\Program Files\Zulu\zulu-17\bin\javaw.exe (Mar 20, 2024, 12:11:55 PM) [pid: 4936]
240320|12:11:59.498|restartedMain|INFO|s.b.w.e.t.TomcatWebServer|initialize|Tomcat initialized with port 8080
240320|12:11:59.527|restartedMain|INFO|o.a.c.c.StandardService|log|Starting service [Tomcat]
240320|12:11:59.528|restartedMain|INFO|o.a.c.core.StandardEngine|log|Starting Servlet engine: [Apache/2.4.18 (Ubuntu)]
240320|12:11:59.655|restartedMain|INFO|o.a.c.c.C.[.[./]|log|Initializing Spring embedded web application
240320|12:11:59.659|restartedMain|INFO|b.ServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext
240320|12:12:00.430|restartedMain|INFO|.OptionalLiveReloadServer|startServer|LiveReload server is running on port 8080
240320|12:12:00.575|restartedMain|INFO|s.b.w.e.t.TomcatWebServer|start|Tomcat started on port 8080
240320|12:12:00.596|restartedMain|INFO|.s.SpringLearnApplication|logStarted|Started SpringLearnApplication
Inside Country Constructor
Setting name:United States
Inside Country Constructor
Setting name:Germany
Inside Country Constructor
Setting name:Japan
240320|12:12:00.755|restartedMain|WARN|PathXmlApplicationContext|refresh|Exception encountered during refresh
```

Hands on 6

Spring Core – Load list of countries from Spring Configuration XML

1. Created a separate bean for each of the four country in country.xml.
2. Created an ArrayList of Country in country.xml.

