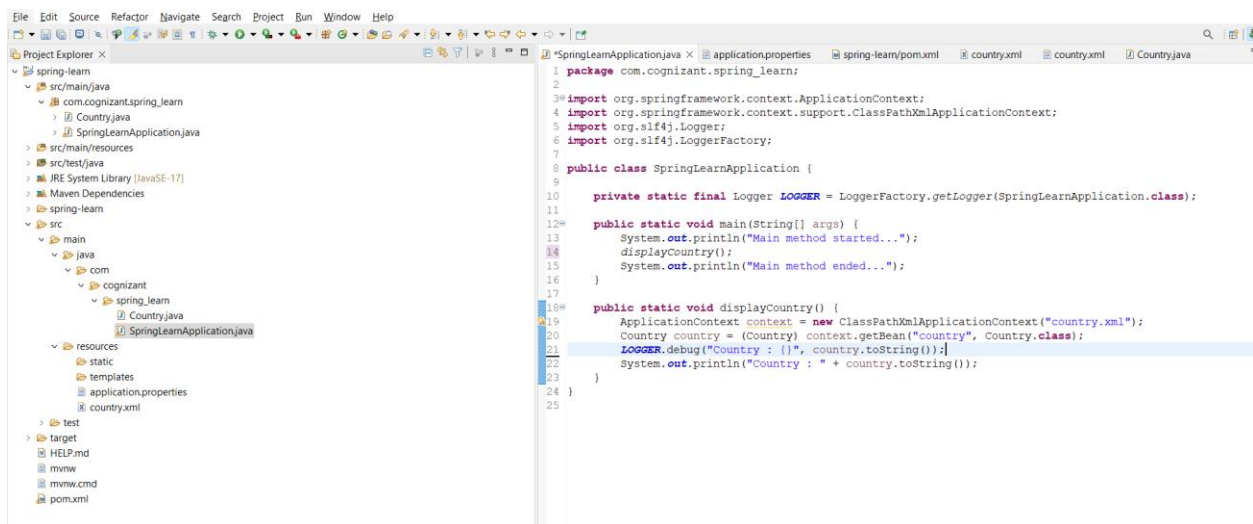


# Digital Nurture 4.0 Deep Skilling

## Week 4- HANDS ON 2

## 2. Spring Core – Load Country from Spring Configuration XML

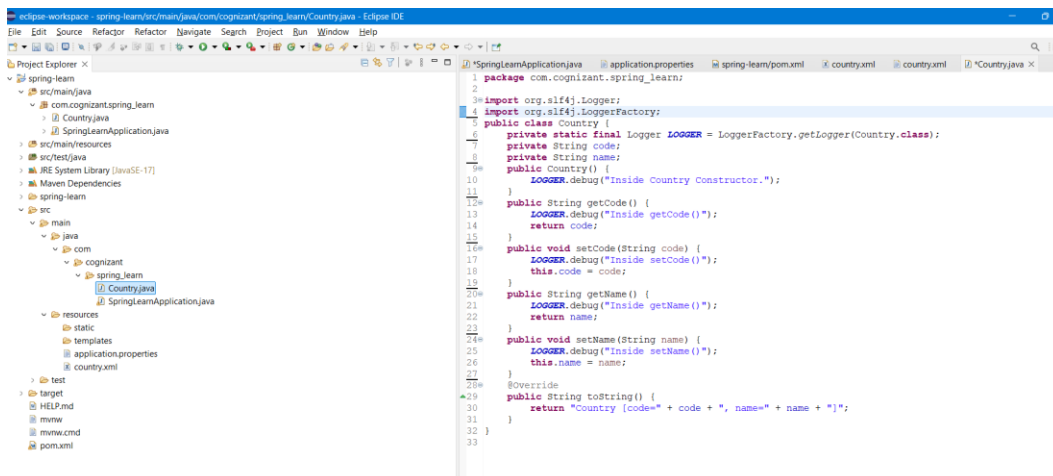
### SpringLearnApplication



The screenshot shows the Eclipse IDE with the SpringLearnApplication.java file open. The Project Explorer on the left shows the project structure: src/main/java/com/cognizant/spring\_learn, src/main/resources, src/test/java, JRE System Library [JavaSE-17], Maven Dependencies, spring-learn, src, main, java, com, cognizant, spring\_learn, Country.java, SpringLearnApplication.java, resources, static, templates, application.properties, country.xml, test, target, HELP.md, mmw, mmw.cmd, pom.xml. The main editor displays the code for SpringLearnApplication.java, which includes package, imports, and a main method that calls displayCountry().

```
1 package com.cognizant.spring_learn;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5 import org.slf4j.Logger;
6 import org.slf4j.LoggerFactory;
7
8 public class SpringLearnApplication {
9
10     private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
11
12     public static void main(String[] args) {
13         System.out.println("Main method started...");
14         displayCountry();
15         System.out.println("Main method ended...");
16     }
17
18     public static void displayCountry() {
19         ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
20         Country country = (Country) context.getBean("country", Country.class);
21         LOGGER.debug("Country : {}, country.toString()");
22         System.out.println("Country : " + country.toString());
23     }
24 }
25
```

### Country.java



The screenshot shows the Eclipse IDE with the Country.java file open. The Project Explorer on the left shows the project structure: src/main/java/com/cognizant/spring\_learn, src/main/resources, src/test/java, JRE System Library [JavaSE-17], Maven Dependencies, spring-learn, src, main, java, com, cognizant, spring\_learn, Country.java, SpringLearnApplication.java, resources, static, templates, application.properties, country.xml, test, target, HELP.md, mmw, mmw.cmd, pom.xml. The main editor displays the code for Country.java, which includes package, imports, private fields, constructor, getters, setters, and toString method.

```
1 package com.cognizant.spring_learn;
2
3 import org.slf4j.Logger;
4 import org.slf4j.LoggerFactory;
5
6 public class Country {
7     private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);
8     private String code;
9     private String name;
10     public Country() {
11         LOGGER.debug("Inside Country Constructor.");
12     }
13     public String getCode() {
14         LOGGER.debug("Inside getCode()");
15         return code;
16     }
17     public void setCode(String code) {
18         LOGGER.debug("Inside setCode()");
19         this.code = code;
20     }
21     public String getName() {
22         LOGGER.debug("Inside getName()");
23         return name;
24     }
25     public void setName(String name) {
26         LOGGER.debug("Inside setName()");
27         this.name = name;
28     }
29     @Override
30     public String toString() {
31         return "Country [code=" + code + ", name=" + name + "];"
32     }
33 }

```

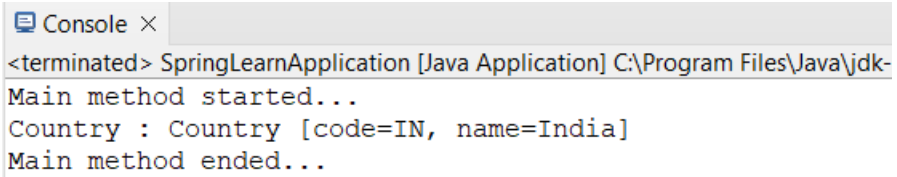
### Country.xml

```
*SpringLearnApplication.java  application.properties  spring-learn/pom.xml  country.xml  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="
5         http://www.springframework.org/schema/beans
6         https://www.springframework.org/schema/beans/spring-beans.xsd">
7
8     <bean id="country" class="com.cognizant.spring_learn.Country">
9         <property name="code" value="IN" />
10        <property name="name" value="India" />
11    </bean>
12
13</beans>
14
```

## Pom.xml dependencies

```
*SpringLearnApplication.java ×  application.properties  *spring-learn/pom.xml ×  country.xml  country.xml  *
30 <java.version>17</java.version>
31 </properties>
32 <dependencies>
33     <dependency>
34         <groupId>org.springframework.boot</groupId>
35         <artifactId>spring-boot-starter-web</artifactId>
36     </dependency>
37     <dependency>
38         <groupId>org.springframework.boot</groupId>
39         <artifactId>spring-boot-devtools</artifactId>
40         <scope>runtime</scope>
41         <optional>true</optional>
42     </dependency>
43     <dependency>
44         <groupId>org.springframework.boot</groupId>
45         <artifactId>spring-boot-starter-test</artifactId>
46         <scope>test</scope>
47     </dependency>
48     <dependency>
49         <groupId>org.springframework</groupId>
50         <artifactId>spring-context</artifactId>
51         <version>6.1.6</version>
52     </dependency>
53     <dependency>
54         <groupId>org.slf4j</groupId>
55         <artifactId>slf4j-api</artifactId>
56         <version>2.0.9</version>
57     </dependency>
58     <dependency>
59         <groupId>ch.qos.logback</groupId>
60         <artifactId>logback-classic</artifactId>
61         <version>1.4.11</version>
62     </dependency>
63 </dependencies>
64
```

## Output :



The screenshot shows a console window titled "Console" with a close button. The text inside the console is as follows:

```
<terminated> SpringLearnApplication [Java Application] C:\Program Files\Java\jdk-  
Main method started...  
Country : Country [code=IN, name=India]  
Main method ended...
```