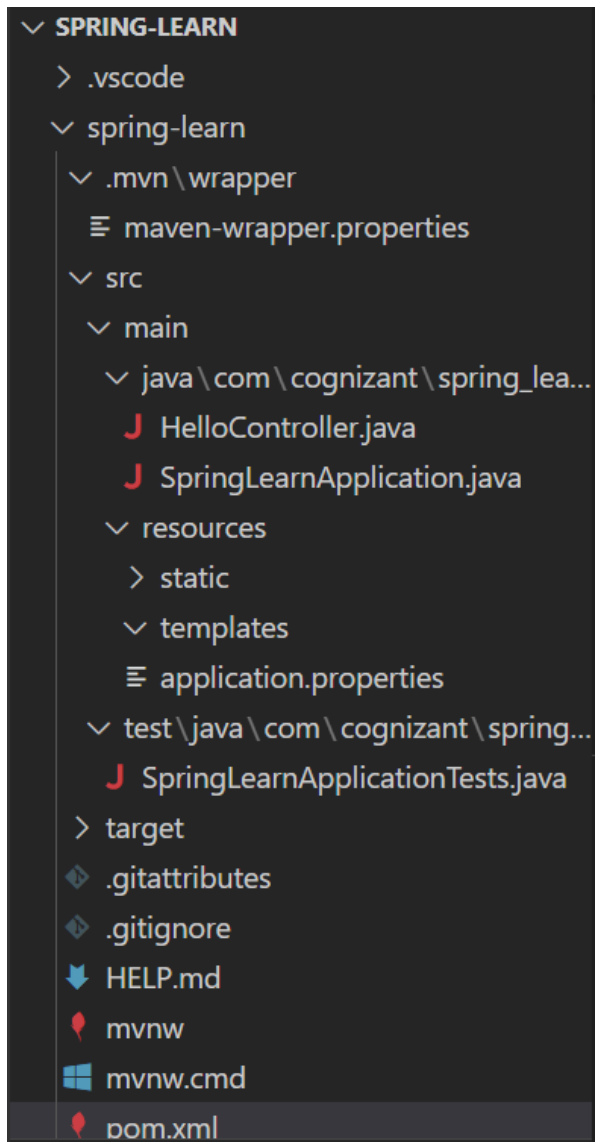


WEEK 4

Create a Spring Web Project using Maven

File structure:



application.properties:

```
1  server.port=8080
2  logging.level.org.springframework=INFO
3
```

SpringLearnApplication.java

```
SpringLearnApplicationTests.java  SpringLearnApplication.java X pom.xml HelloController
spring-learn > src > main > java > com > cognizant > spring_learn > SpringLearnAppli
1 package com.cognizant.spring_learn;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class SpringLearnApplication {
8
9     Run | Debug
10    public static void main(String[] args) {
11        System.out.println(x:"SpringLearnApplication main() started...");
12        SpringApplication.run(SpringLearnApplication.class, args);
13    }
14 }
```

SpringLearnApplicationTests.java

```
SpringLearnApplicationTests.java X SpringLearnApplication.java pom.xml
spring-learn > src > test > java > com > cognizant > spring_learn > SpringLearnAppli
1 package com.cognizant.spring_learn;
2
3 import org.junit.jupiter.api.Test;
4 import org.springframework.boot.test.context.SpringBootTest;
5
6 @SpringBootTest
7 class SpringLearnApplicationTests {
8
9     @Test
10    void contextLoads() {
11    }
12 }
13
```

HelloController.java

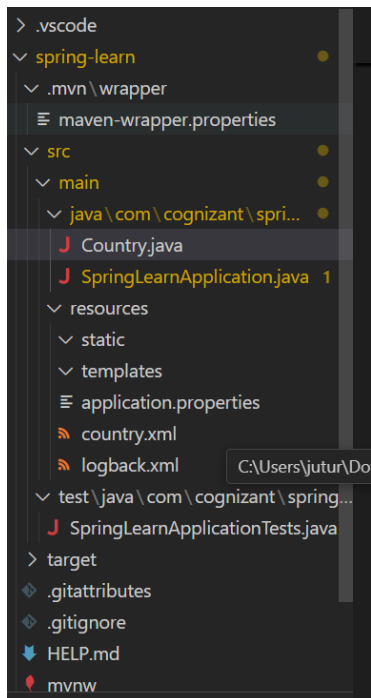
```
SpringLearnApplicationTests.java SpringLearnApplication.java pom.xml HelloController.java X
spring-learn > src > main > java > com > cognizant > spring_learn > HelloController.java
1 package com.cognizant.spring_learn;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HelloController {
8
9     http://127.0.0.1:8080/hello
10    @GetMapping("/hello")
11    public String sayHello() {
12        return "Hello from Spring Boot!";
13    }
14 }
```

Output:

```
< > ↺ ⓘ localhost:8080/hello
Hello from Spring Boot!
```

Spring Core – Load Country from Spring Configuration XML

File structure:



Country.java

```
SpringLearnApplicationTests.java | SpringLearnApplication.java 1 | logback.xml | pom.xml |
spring-learn > src > main > java > com > cognizant > spring_learn > Country.java > ...
4  import org.slf4j.Logger;
5  import org.slf4j.LoggerFactory;
6
7  public class Country {
8      private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);
9
10     private String code;
11     private String name;
12
13     public Country() {
14         LOGGER.debug("Inside Country Constructor.");
15     }
16
17     public String getCode() {
18         LOGGER.debug("Inside getCode()");
19         return code;
20     }
21
22     public void setCode(String code) {
23         LOGGER.debug("Inside setCode()");
24         this.code = code;
25     }
26
27     public String getName() {
28         LOGGER.debug("Inside getName()");
29         return name;
30     }
31
32     public void setName(String name) {
33         LOGGER.debug("Inside setName()");
```

spring-learn > src > main > java > com > cognizant > spring_learn > J Country.java > ...

```
7   public class Country {
17      public String getCode() {
20      }
21
22      public void setCode(String code) {
23          LOGGER.debug("Inside setCode()");
24          this.code = code;
25      }
26
27      public String getName() {
28          LOGGER.debug("Inside getName()");
29          return name;
30      }
31
32      public void setName(String name) {
33          LOGGER.debug("Inside setName()");
34          this.name = name;
35      }
36
37      @Override
38      public String toString() {
39          return "Country [code=" + code + ", name=" + name + "]";
40      }
41  }
```

SpringLearnApplication.java

spring-learn > src > main > java > com > cognizant > spring_learn > J SpringLearnApplication.java > ...

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

Country.xml

```
spring-learn > src > main > resources > 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          http://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
```

logback.xml

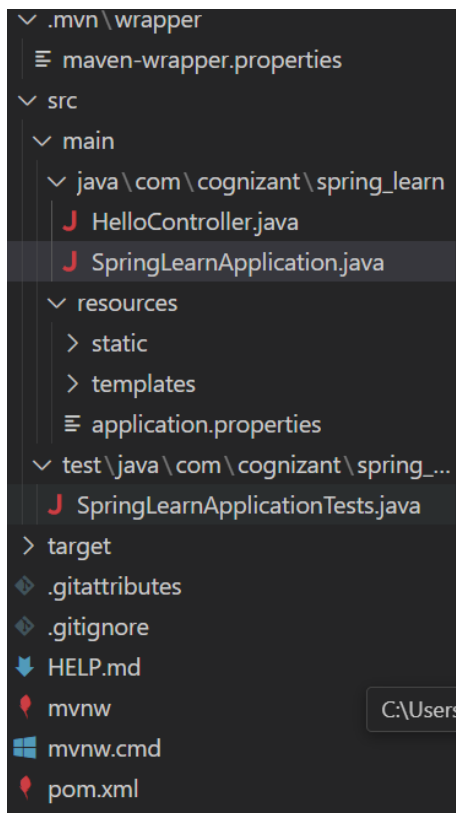
```
spring-learn > src > main > resources > logback.xml
1  <configuration>
2      <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
3          <encoder>
4              <pattern>%d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level %logger{36} - %msg%n</pattern>
5          </encoder>
6      </appender>
7
8      <root level="debug">
9          <appender-ref ref="STDOUT" />
10     </root>
11 </configuration>
12
```

Output:

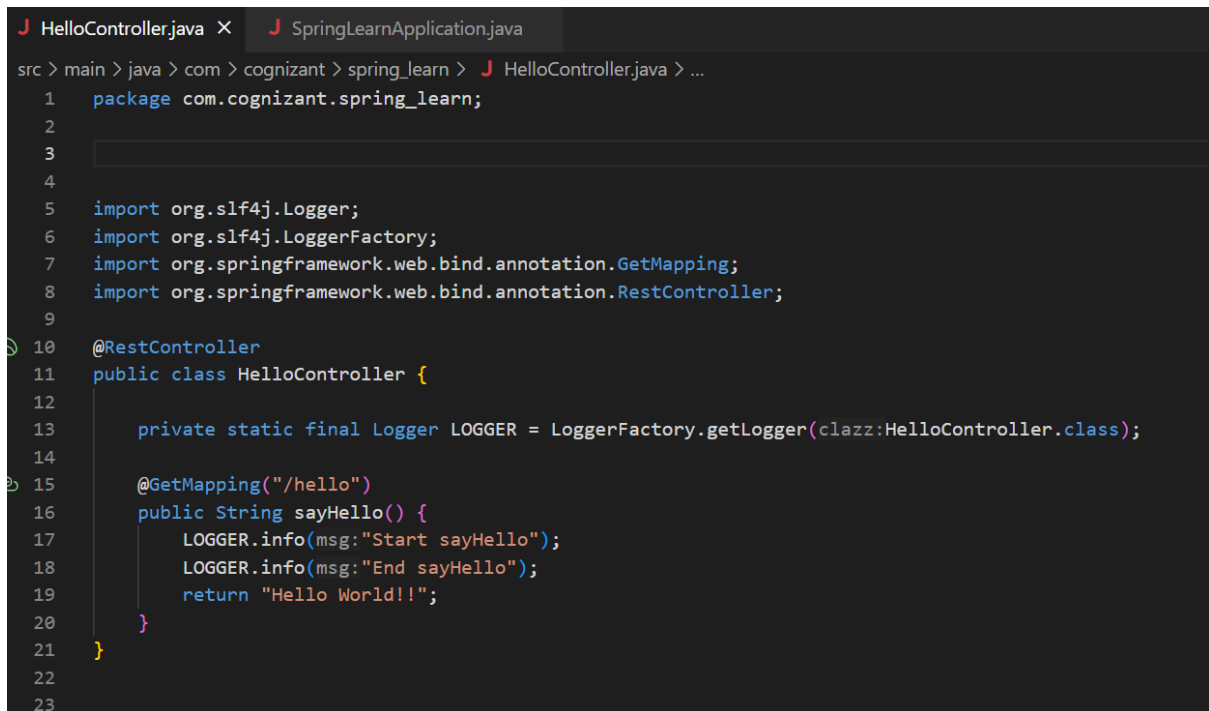
```
2025-07-11 17:08:14 [main] DEBUG c.c.s.SpringLearnApplication - Inside main()
2025-07-11 17:08:14 [main] DEBUG o.s.c.s.ClassPathXmlApplicationContext - Refreshing org.springframework.con
.support.ClassPathXmlApplicationContext@6bedbc4d
2025-07-11 17:08:14 [main] DEBUG o.s.b.f.xml.XmlBeanDefinitionReader - Loaded 1 bean definitions from class
resource [country.xml]
2025-07-11 17:08:14 [main] DEBUG o.s.b.f.s.DefaultListableBeanFactory - Creating shared instance of singleto
an 'country'
2025-07-11 17:08:14 [main] DEBUG com.cognizant.spring_learn.Country - Inside Country Constructor.
2025-07-11 17:08:14 [main] DEBUG com.cognizant.spring_learn.Country - Inside setCode()
2025-07-11 17:08:14 [main] DEBUG com.cognizant.spring_learn.Country - Inside setName()
2025-07-11 17:08:14 [main] DEBUG c.c.s.SpringLearnApplication - Country : Country [code=IN, name=India]
PS C:\Users\jutur\Downloads\spring-learn>
```

Hello World RESTful Web Service

File Structure:



HelloController.java



SpringLearnApplication.java

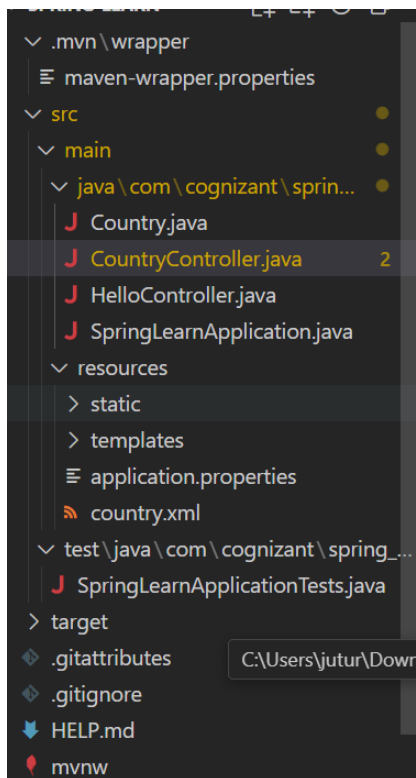
```
c > main > java > com > cognizant > spring_learn > SpringLearnApplication.java > ...
1  package com.cognizant.spring_learn;
2
3  import org.springframework.boot.SpringApplication;
4  import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6  @SpringBootApplication
7  public class SpringLearnApplication {
8      Run | Debug
9      public static void main(String[] args) {
10         SpringApplication.run(primarySource:SpringLearnApplication.class, args);
11     }
12 }
```

Output:

```
← → ↺ localhost:8080/hello
Hello World!!
```


REST - Country Web Service

File structure:



Country.java



CountryController.java:

```
src > main > java > com > cognizant > spring_learn > CountryController.java > ...
1  package com.cognizant.spring_learn;
2
3
4  import com.cognizant.spring_learn.Country;
5  import org.slf4j.Logger;
6  import org.slf4j.LoggerFactory;
7  import org.springframework.context.ApplicationContext;
8  import org.springframework.context.support.ClassPathXmlApplicationContext;
9  import org.springframework.web.bind.annotation.RequestMapping;
10 import org.springframework.web.bind.annotation.RestController;
11
12 @RestController
13 public class CountryController {
14
15     private static final Logger LOGGER = LoggerFactory.getLogger(clazz:CountryController.class);
16
17     @RequestMapping("/country")
18     public Country getCountryIndia() {
19         LOGGER.info(msg:"Start getCountryIndia");
20
21         ApplicationContext context = new ClassPathXmlApplicationContext(configLocation:"country.xml");
22         Country country = (Country) context.getBean(name:"in");
23
24         LOGGER.info(msg:"End getCountryIndia");
25         return country;
26     }
27 }
28
```

Output:

```
localhost:8080/country
Pretty-print ☐
{"code":"IN","name":"India"}
```

REST - Get country based on country code

Country.java

```
src > main > java > com > cognizant > spring_learn > J Country.java > ...
1  package com.cognizant.spring_learn;
2
3
4  public class Country {
5      private String code;
6      private String name;
7
8      // Getters and Setters
9      public String getCode() {
10         return code;
11     }
12     public void setCode(String code) {
13         this.code = code;
14     }
15     public String getName() {
16         return name;
17     }
18     public void setName(String name) {
19         this.name = name;
20     }
21 }
22
```

CountryController.java

```
src > main > java > com > cognizant > spring_learn > J CountryController.java > ...
1  package com.cognizant.spring_learn;
2
3
4
5
6  import com.cognizant.spring_learn.Country;
7  import com.cognizant.spring_learn.CountryService;
8  import org.slf4j.Logger;
9  import org.slf4j.LoggerFactory;
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.web.bind.annotation.*;
12
13 @RestController
14 public class CountryController {
15
16     private static final Logger LOGGER = LoggerFactory.getLogger(clazz:CountryController.class);
17
18     @Autowired
19     private CountryService countryService;
20
21     @GetMapping("/countries/{code}")
22     public Country getCountry(@PathVariable String code) {
23         LOGGER.info(format:"Start getCountry - code: {}", code);
24         Country country = countryService.getCountry(code);
25         LOGGER.info(msg:"End getCountry");
26         return country;
27     }
28 }
29
```

CountryService.java

```
J SpringLearnApplication.java  country.xml  J CountryController.java 2  J CountryService.java 3 X  J Country.java  >
src > main > java > com > cognizant > spring_learn > J CountryService.java > ...
1  package com.cognizant.spring_learn;
2
3
4
5  import com.cognizant.spring_learn.Country;
6  import org.springframework.context.ApplicationContext;
7  import org.springframework.context.support.ClassPathXmlApplicationContext;
8  import org.springframework.stereotype.Service;
9
10 import java.util.List;
11
12 @Service
13 public class CountryService {
14
15     public Country getCountry(String code) {
16         ApplicationContext context = new ClassPathXmlApplicationContext(configLocation:"country.xml");
17         List<Country> countries = context.getBean(name:"countryList", requiredType:List.class);
18
19         return countries.stream()
20             .filter(c -> c.getCode().equalsIgnoreCase(code))
21             .findFirst()
22             .orElse(other:null);
23     }
24 }
25
```

Output:

```
pretty print
{"code":"IN","name":"India"}
```