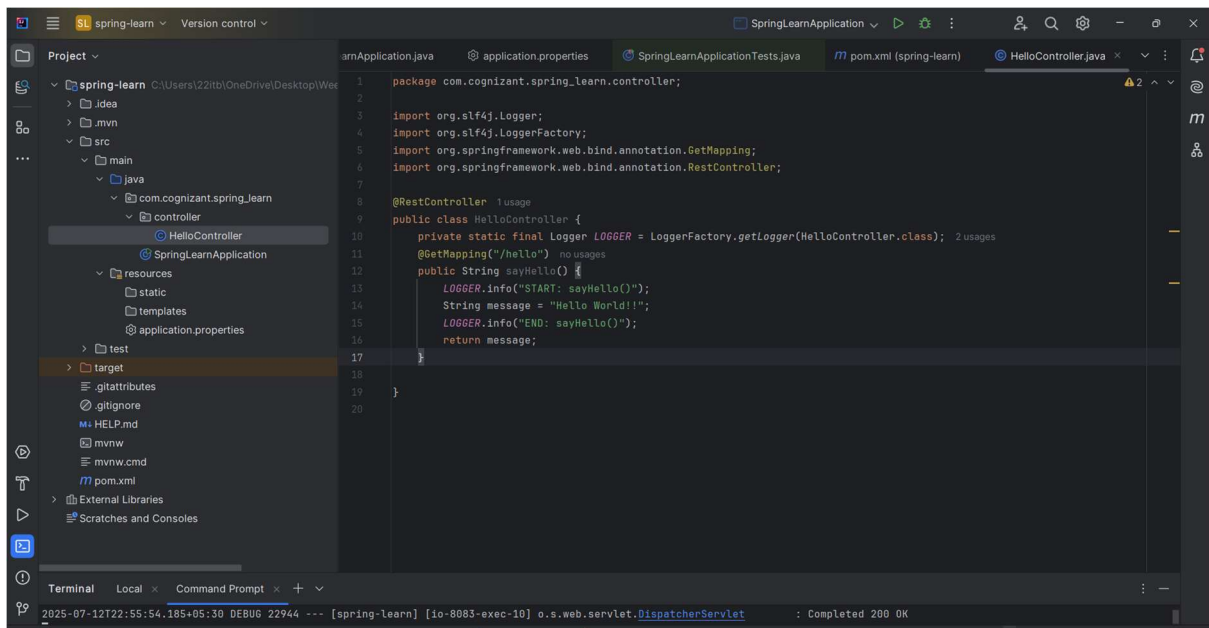


Week 4

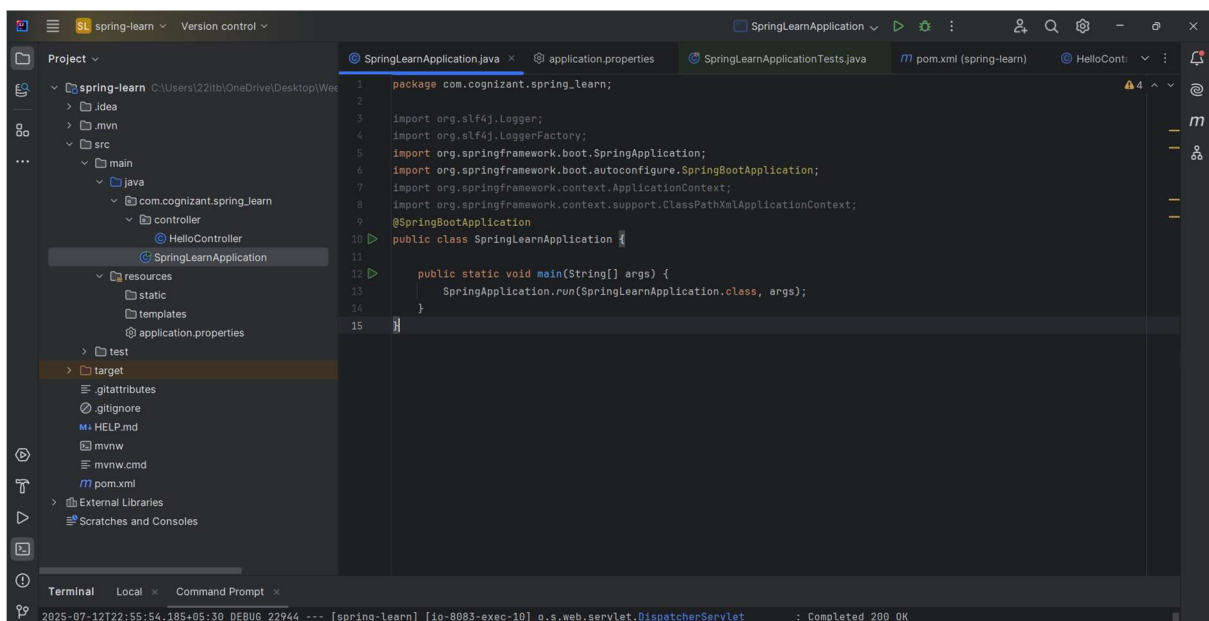
Hello World RESTful Web Service

Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework. Refer details below:

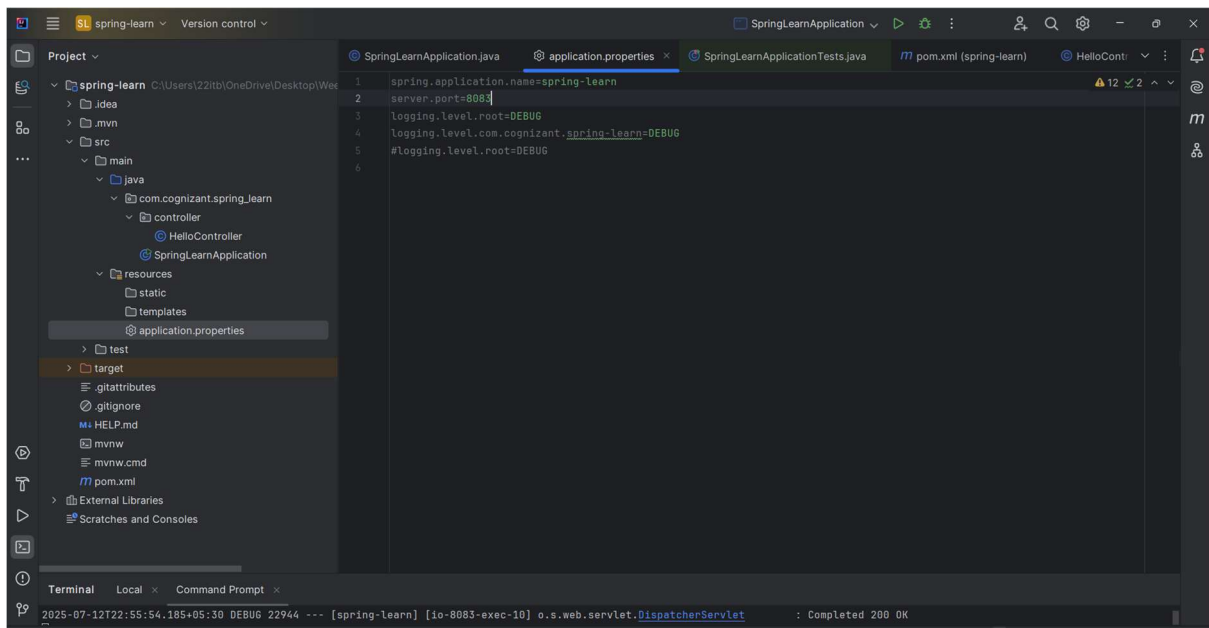
HelloController



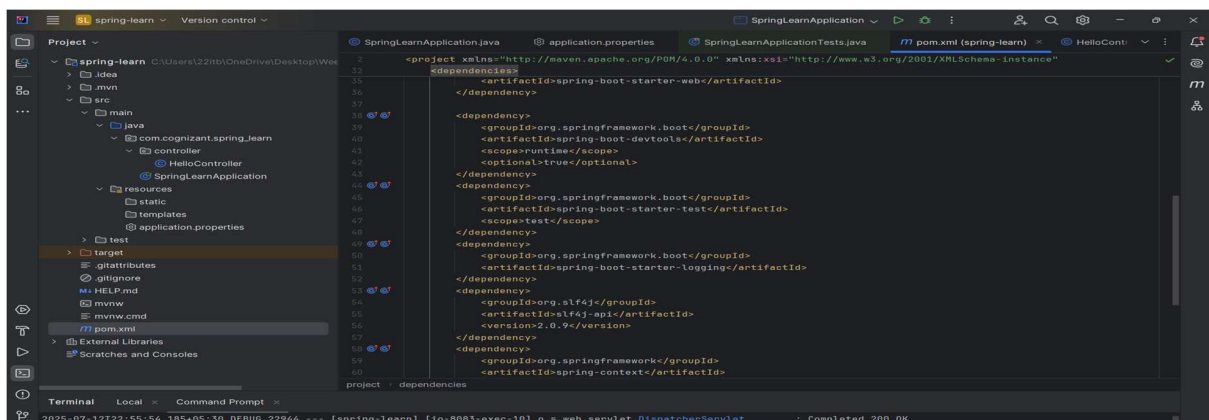
SpringLearnApplication



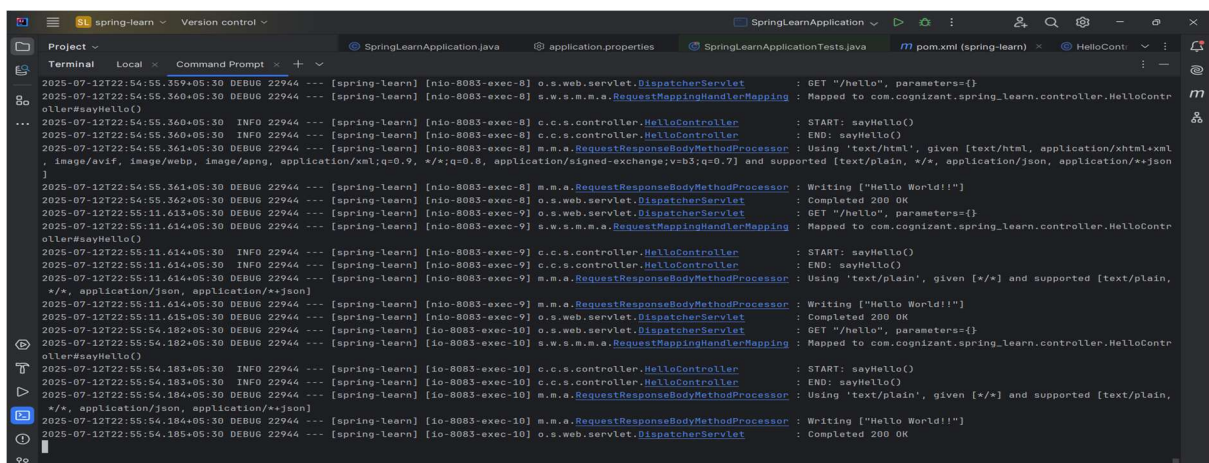
Application.property



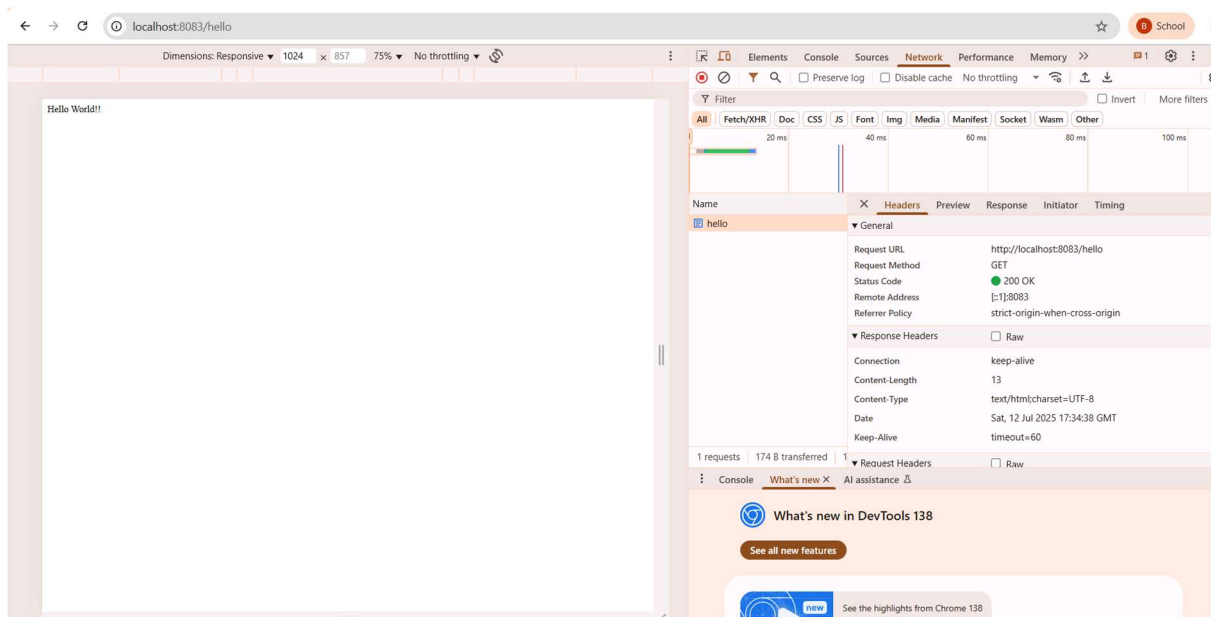
Pom.xml



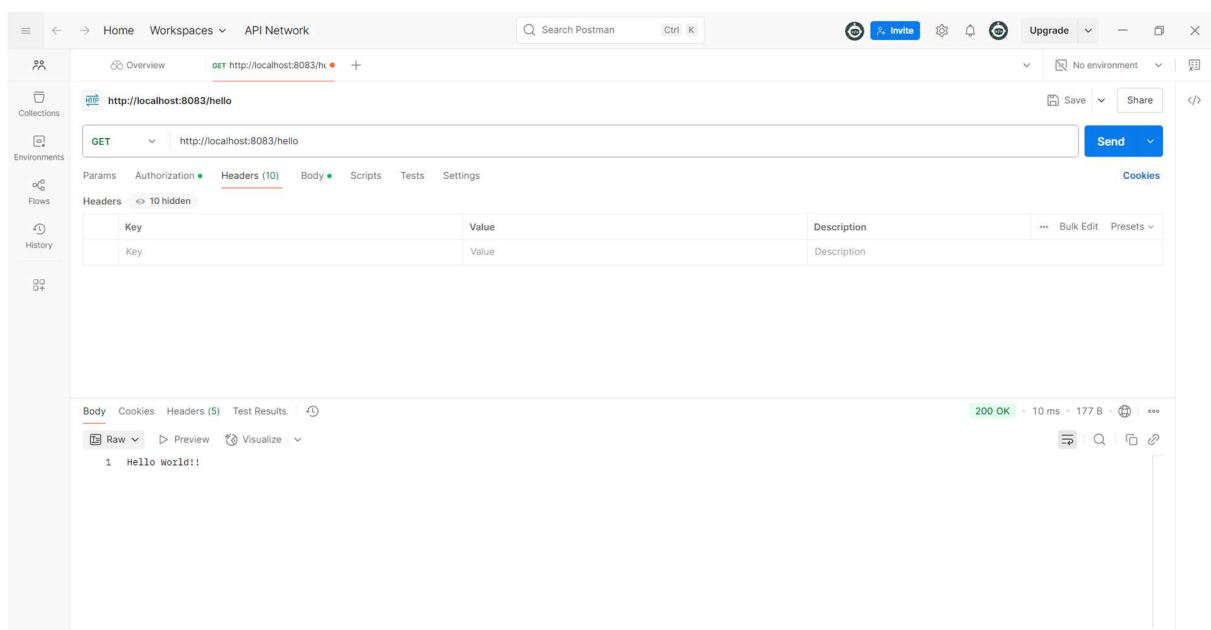
Console Output



Web Output



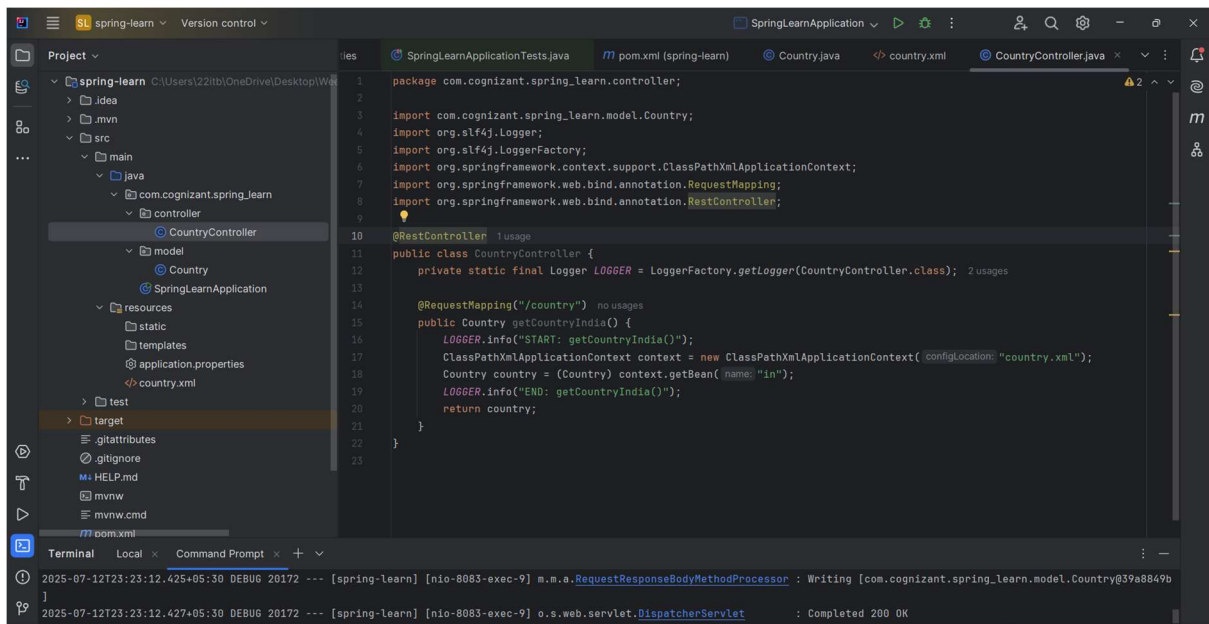
Postman output



REST - Country Web Service

Write a REST service that returns India country details in the earlier created spring learn application.

CountryController

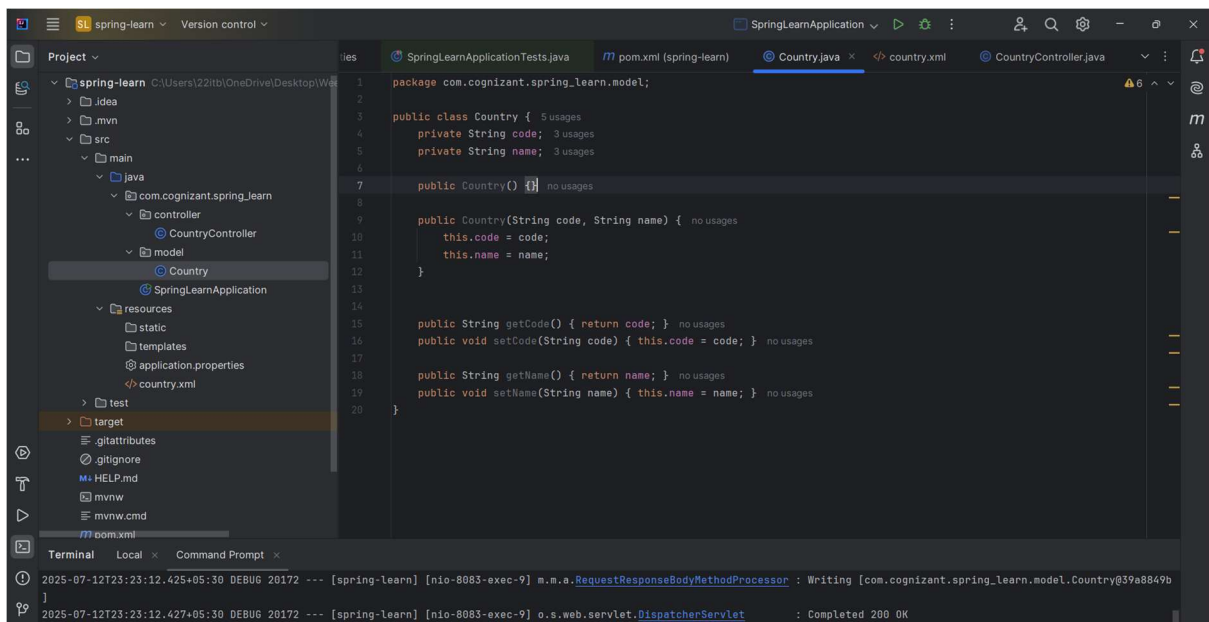


```
1 package com.cognizant.spring_learn.controller;
2
3 import com.cognizant.spring_learn.model.Country;
4 import org.slf4j.Logger;
5 import org.slf4j.LoggerFactory;
6 import org.springframework.context.support.ClassPathXmlApplicationContext;
7 import org.springframework.web.bind.annotation.RequestMapping;
8 import org.springframework.web.bind.annotation.RestController;
9
10 @RestController
11 public class CountryController {
12     private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);
13
14     @RequestMapping("/country")
15     public Country getCountryIndia() {
16         LOGGER.info("START: getCountryIndia()");
17         ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
18         Country country = (Country) context.getBean("in");
19         LOGGER.info("END: getCountryIndia()");
20         return country;
21     }
22 }
23
```

Terminal output:

```
2025-07-12T23:23:12.425+05:30 DEBUG 20172 --- [spring-learn] [nio-8083-exec-9] m.m.a.RequestMappingHandlerMapping : Writing [com.cognizant.spring_learn.model.Country@39a8849b]
2025-07-12T23:23:12.427+05:30 DEBUG 20172 --- [spring-learn] [nio-8083-exec-9] o.s.web.servlet.DispatcherServlet : Completed 200 OK
```

Model :Country

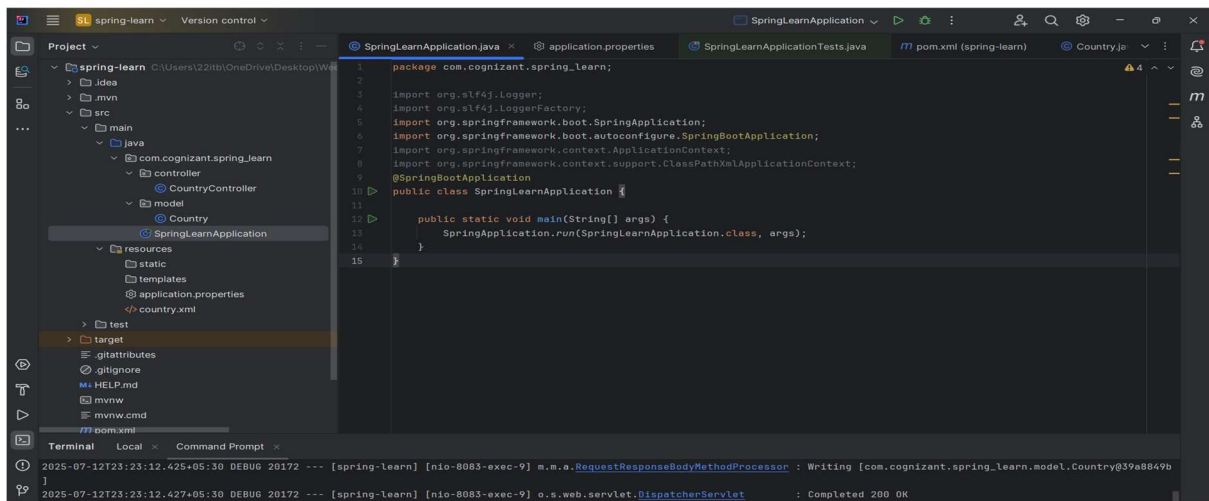


```
1 package com.cognizant.spring_learn.model;
2
3 public class Country {
4     private String code;
5     private String name;
6
7     public Country() {}
8
9     public Country(String code, String name) {
10         this.code = code;
11         this.name = name;
12     }
13
14     public String getCode() { return code; }
15     public void setCode(String code) { this.code = code; }
16
17     public String getName() { return name; }
18     public void setName(String name) { this.name = name; }
19 }
20
```

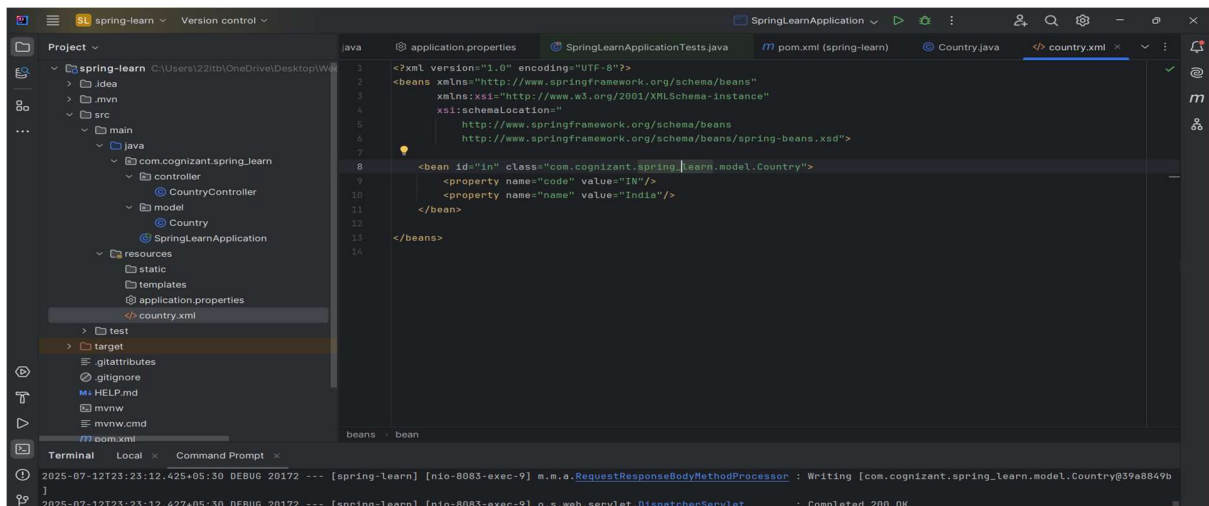
Terminal output:

```
2025-07-12T23:23:12.425+05:30 DEBUG 20172 --- [spring-learn] [nio-8083-exec-9] m.m.a.RequestMappingHandlerMapping : Writing [com.cognizant.spring_learn.model.Country@39a8849b]
2025-07-12T23:23:12.427+05:30 DEBUG 20172 --- [spring-learn] [nio-8083-exec-9] o.s.web.servlet.DispatcherServlet : Completed 200 OK
```

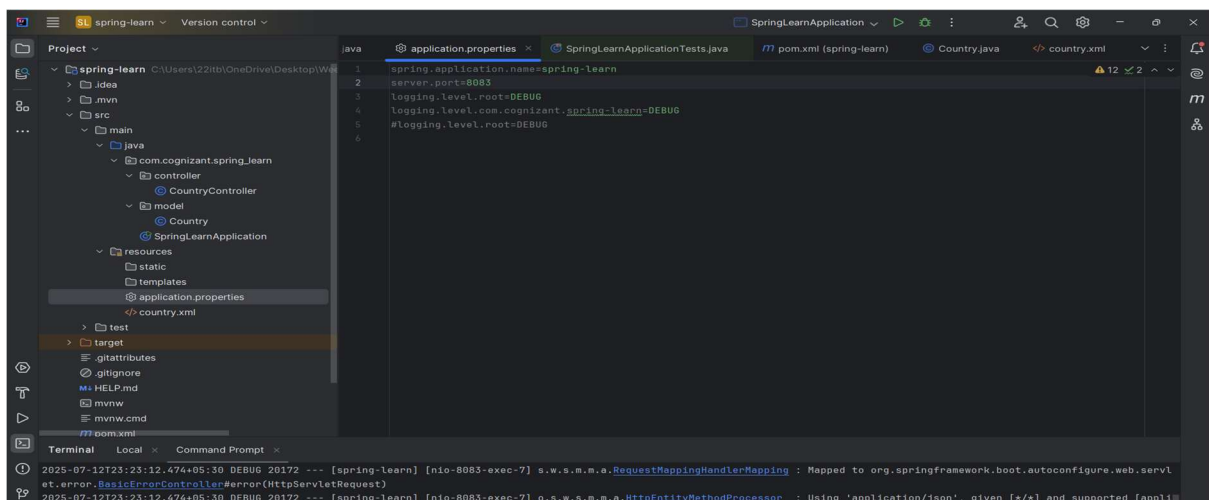
SpringLearnApplication



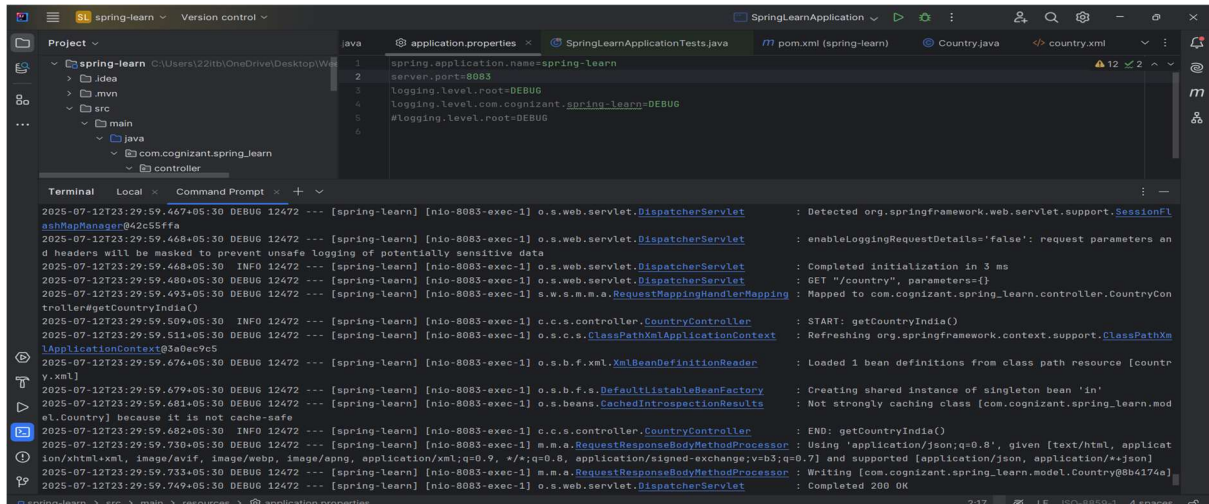
Country.xml



Application.property



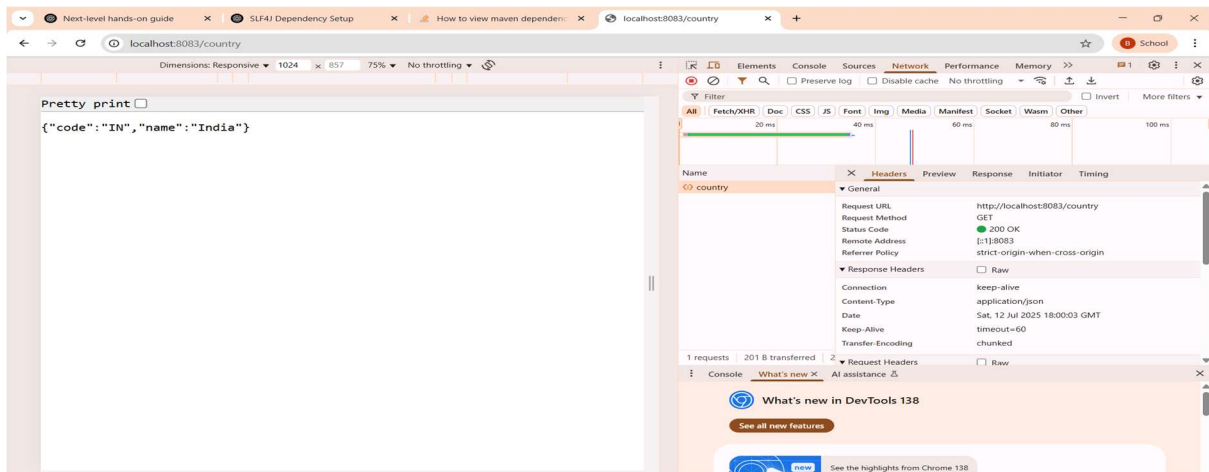
Console Output



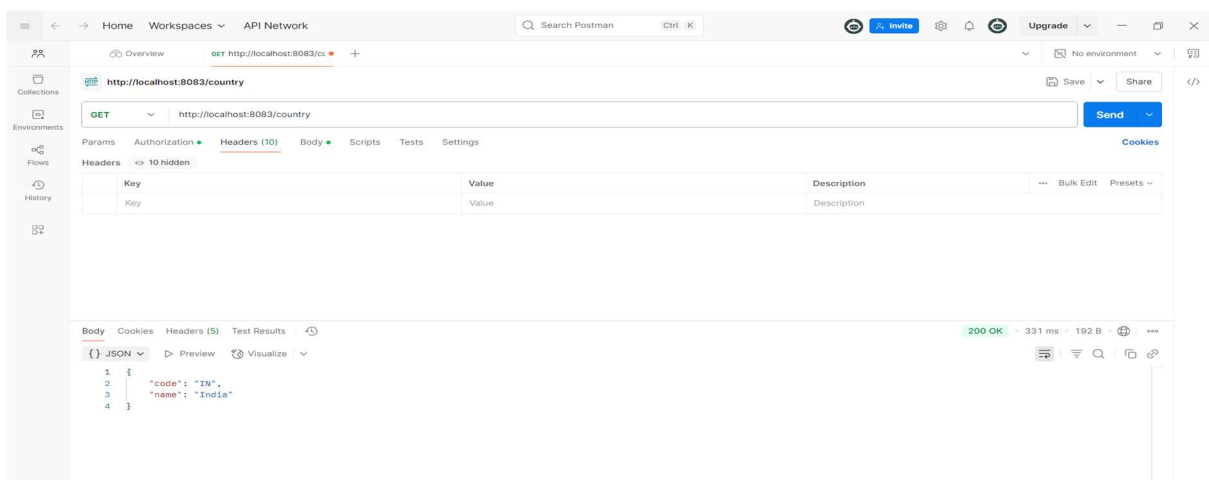
The screenshot shows the console output of a Spring Boot application. The logs indicate that the application is running on port 8083, with logging level set to DEBUG. The application context is refreshed, and the CountryController is started. The application is also loaded with 1 bean definition from the class path resource. The logs show the application is running successfully.

```
2025-07-12T23:29:59.467:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : Detected org.springframework.web.servlet.support.SessionFl
ashMapManager@42c55ffa
2025-07-12T23:29:59.468:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : enableLoggingRequestDetails='false': request parameters an
d headers will be masked to prevent unsafe logging of potentially sensitive data
2025-07-12T23:29:59.480:05:30 INFO 12472 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
2025-07-12T23:29:59.493:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : GET "/country", parameters={}
2025-07-12T23:29:59.509:05:30 INFO 12472 --- [spring-learn] [nio-8083-exec-1] s.w.s.m.a.RequestMappingHandlerMapping : Mapped to com.cognizant.spring_learn.controller.CountryCon
troller#getCountrYIndia()
2025-07-12T23:29:59.511:05:30 INFO 12472 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.CountryController : START: getCountrYIndia()
2025-07-12T23:29:59.511:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] o.s.c.s.ClassPathXmlApplicationContext : Refreshing org.springframework.context.support.ClassPathXml
ApplicationContext@5a0ec9c5
2025-07-12T23:29:59.676:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] o.s.b.f.xml.XmlBeanDefinitionReader : Loaded 1 bean definitions from class path resource [countr
y.xml]
2025-07-12T23:29:59.679:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] o.s.b.f.s.DefaultListableBeanFactory : Creating shared instance of singleton bean 'in
'
2025-07-12T23:29:59.681:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] o.s.beans.CachedIntrospectionResults : Not strongly caching class [com.cognizant.spring_learn.mod
el.Country] because it is not cache-safe
2025-07-12T23:29:59.682:05:30 INFO 12472 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.CountryController : END: getCountrYIndia()
2025-07-12T23:29:59.730:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] m.m.a.RequestResponseBodyMethodProcessor : Using 'application/json;q=0.8', given [text/html, applicat
ion/xhtml+xml, image/avif, image/webp, image/apng, application/xml;q=0.9, */*;q=0.8, application/signed-exchange;v=b3;q=0.7] and supported [application/json, application/*+json]
2025-07-12T23:29:59.733:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] m.m.a.RequestResponseBodyMethodProcessor : Writing [com.cognizant.spring_learn.model.Country@8b4174a]
2025-07-12T23:29:59.749:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : Completed 200 OK
```

Web output



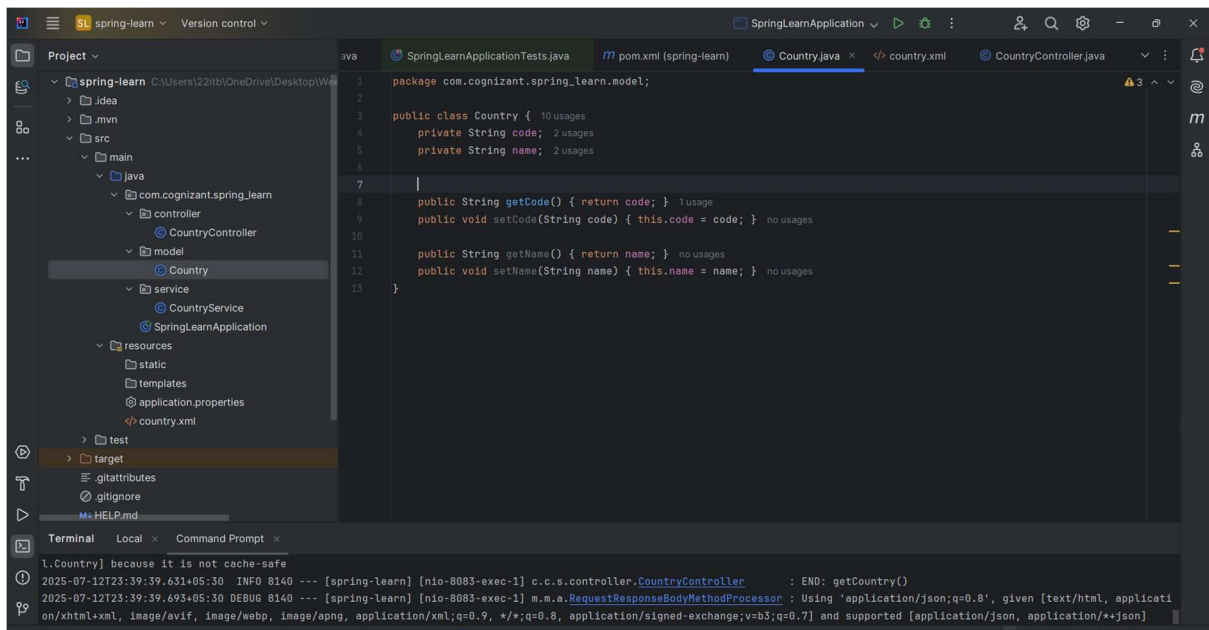
Postman output



REST - Get country based on country code

Write a REST service that returns a specific country based on country code. The country code should be case insensitive

Model : Country

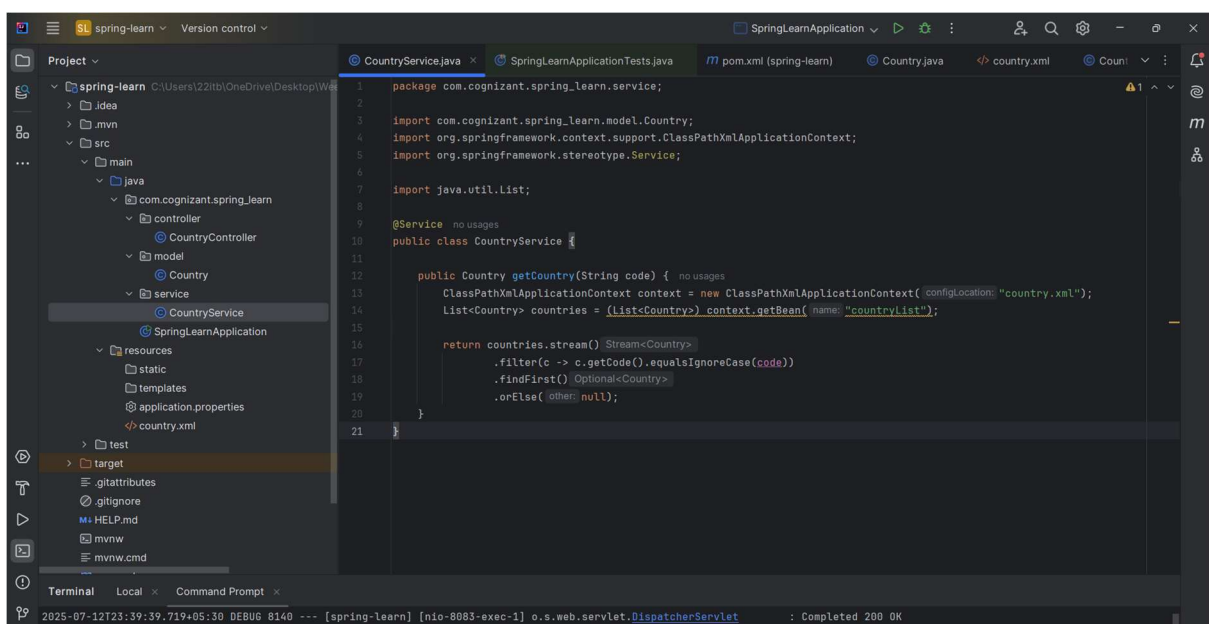


The screenshot shows an IDE with the 'Country.java' file open. The code defines a simple model class with two attributes: 'code' and 'name', both of type 'String'. It includes getter and setter methods for both attributes. The package is 'com.cognizant.spring_learn.model'.

```
1 package com.cognizant.spring_learn.model;
2
3 public class Country {
4     private String code;
5     private String name;
6
7     public String getCode() { return code; }
8     public void setCode(String code) { this.code = code; }
9
10    public String getName() { return name; }
11    public void setName(String name) { this.name = name; }
12 }
13
```

The terminal at the bottom shows a successful HTTP GET request to the '/getCountry' endpoint, returning a JSON response for the country with code '08'.

Service :CountryService

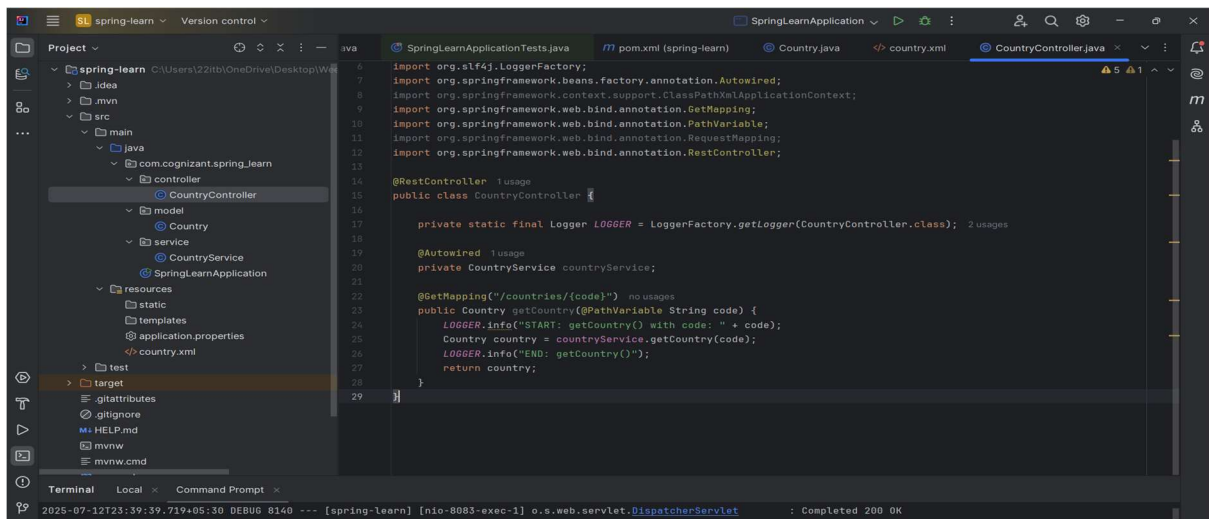


The screenshot shows the 'CountryService.java' file in the IDE. The class is annotated with '@Service' and implements the 'getCountry' method. This method uses a 'ClassPathXmlApplicationContext' to load a list of countries from 'country.xml' and returns the first country whose code matches the provided 'code' (case-insensitive). The package is 'com.cognizant.spring_learn.service'.

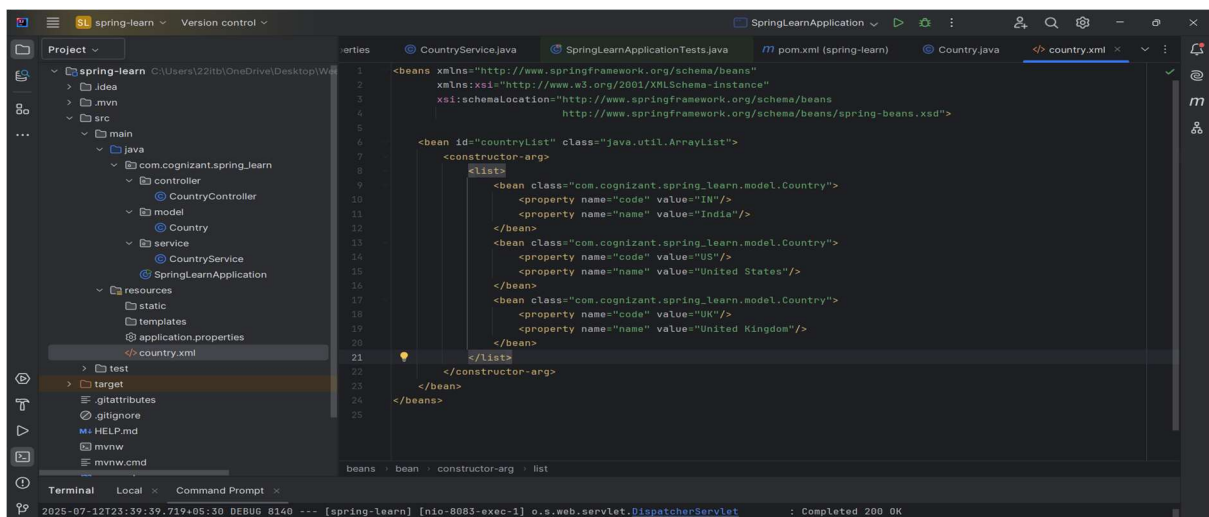
```
1 package com.cognizant.spring_learn.service;
2
3 import com.cognizant.spring_learn.model.Country;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5 import org.springframework.stereotype.Service;
6
7 import java.util.List;
8
9 @Service
10 public class CountryService {
11
12     public Country getCountry(String code) {
13         ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
14         List<Country> countries = (List<Country>) context.getBean("countryList");
15
16         return countries.stream().filter(c -> c.getCode().equalsIgnoreCase(code))
17             .findFirst().orElse(null);
18     }
19 }
20
```

The terminal at the bottom shows the application running successfully, with a 'Completed 200 OK' status for the GET request.

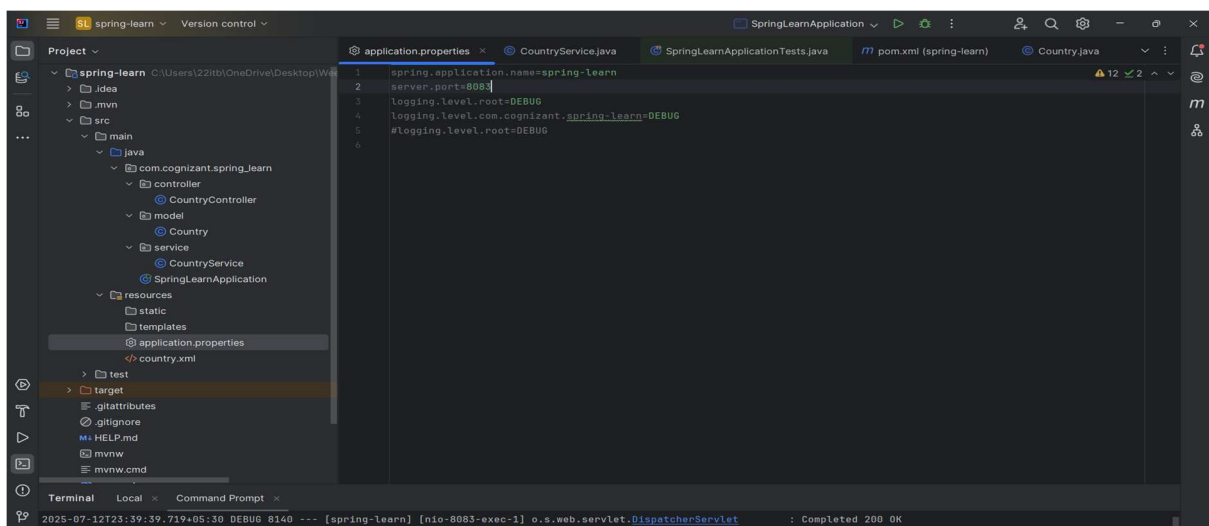
Controller: CountryController



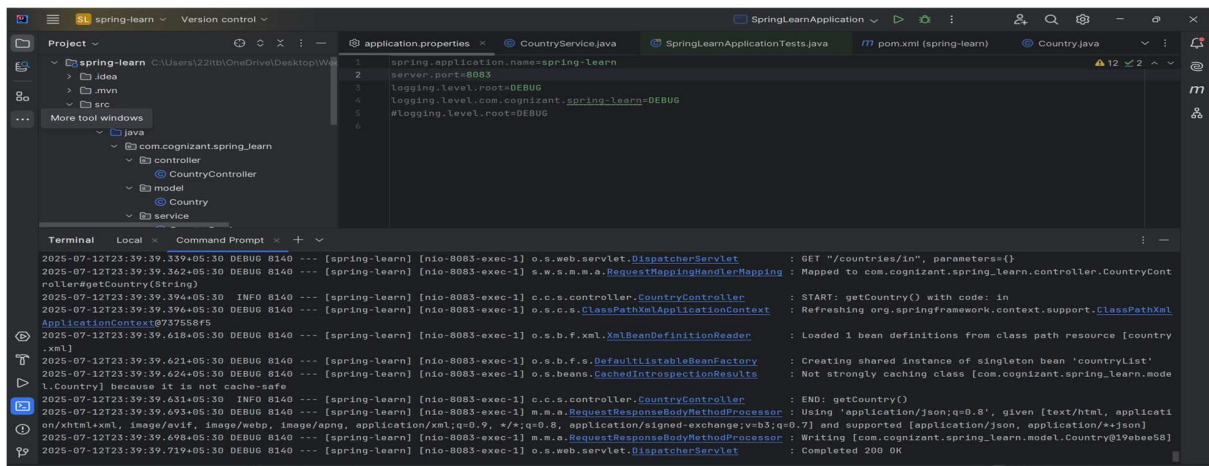
Country.xml



Application.xml



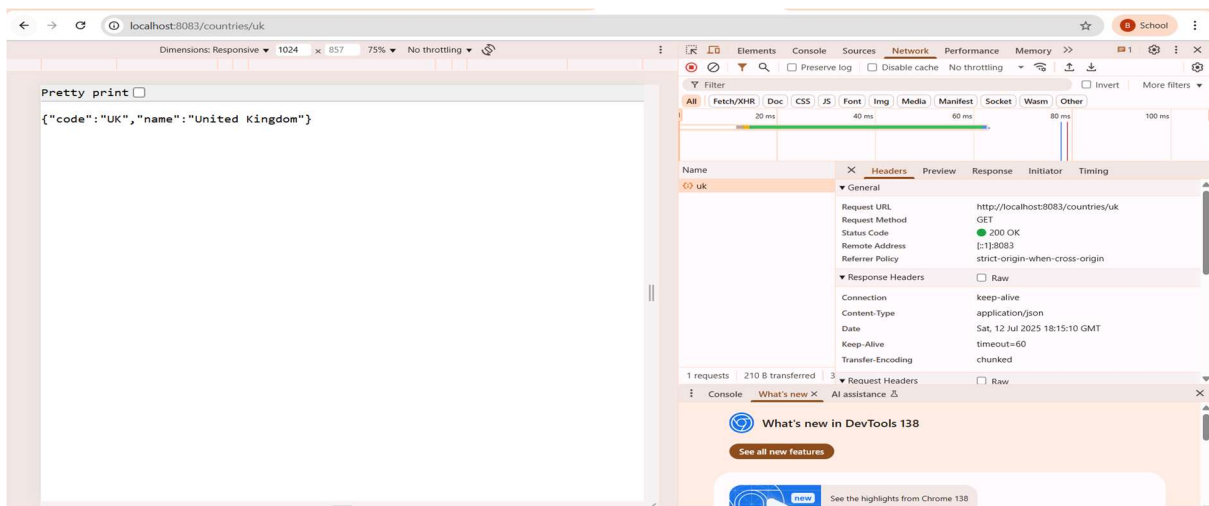
Console Output



The screenshot shows the console output of a Spring Boot application. The logs indicate that the application is running on port 8083 with logging level DEBUG. The logs show the following sequence of events:

- GET "/countries/in", parameters={}
- Mapped to com.cognizant.spring_learn.controller.CountryController
- START: getCountry() with code: in
- Refreshing org.springframework.context.support.ClassPathXmlApplicationContext
- Loaded 1 bean definitions from class path resource [country.xml]
- Creating shared instance of singleton bean 'countryList'
- Not strongly caching class [com.cognizant.spring_learn.model.Country]
- END: getCountry()
- Using 'application/json;q=0.8', given [text/html, application/xhtml+xml, image/avif, image/webp, image/apng, application/xml;q=0.9, */*;q=0.8, application/signed-exchange;v=b3;q=0.7] and supported [application/json, application/*+json]
- Writing [com.cognizant.spring_learn.model.Country@19ebae58]
- Completed 200 OK

Web Output



Postman output

