

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:

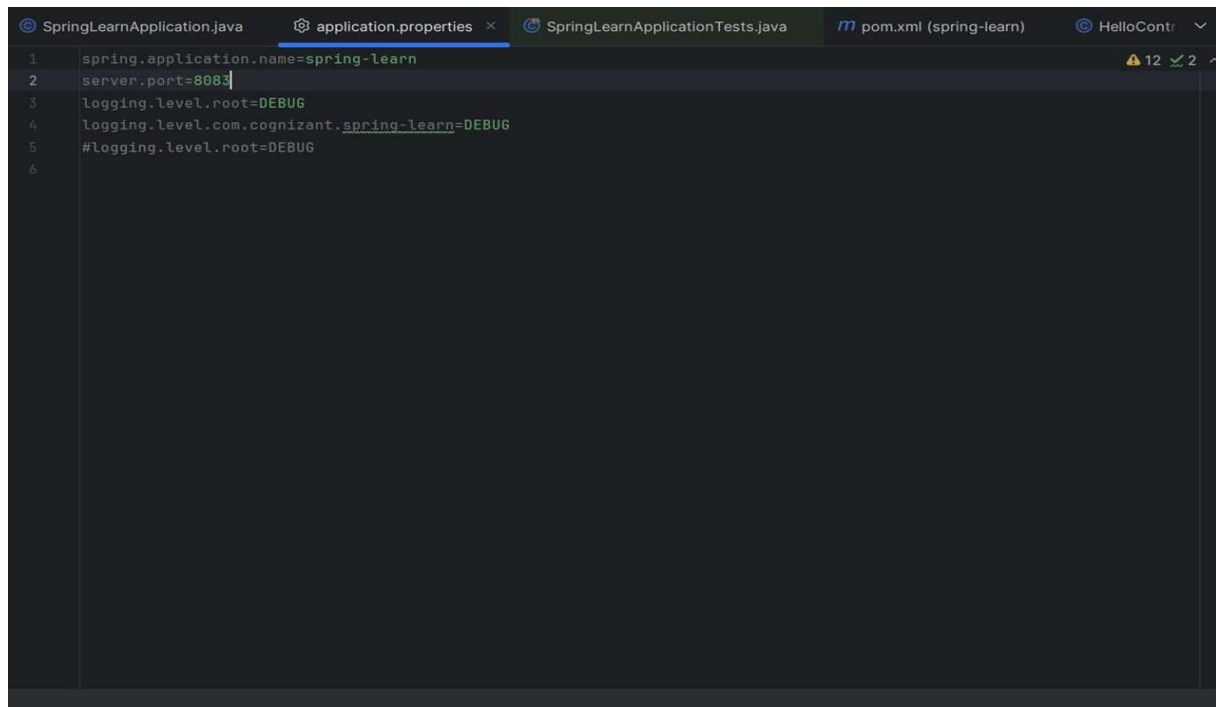
HelloController

```
arnApplication.java  application.properties  SpringLearnApplicationTests.java  pom.xml (spring-learn)  HelloController.java x
1  package com.cognizant.spring_learn.controller;
2
3  import org.slf4j.Logger;
4  import org.slf4j.LoggerFactory;
5  import org.springframework.web.bind.annotation.GetMapping;
6  import org.springframework.web.bind.annotation.RestController;
7
8  @RestController 1 usage
9  public class HelloController {
10     private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class); 2 usages
11     @GetMapping("/hello") no usages
12     public String sayHello() {
13         LOGGER.info("START: sayHello()");
14         String message = "Hello World!!";
15         LOGGER.info("END: sayHello()");
16         return message;
17     }
18
19 }
20
```

SpringLearnApplication

```
SpringLearnApplication.java x  application.properties  SpringLearnApplicationTests.java  pom.xml (spring-learn)  HelloConti
1  package com.cognizant.spring_learn;
2
3  import org.slf4j.Logger;
4  import org.slf4j.LoggerFactory;
5  import org.springframework.boot.SpringApplication;
6  import org.springframework.boot.autoconfigure.SpringBootApplication;
7  import org.springframework.context.ApplicationContext;
8  import org.springframework.context.support.ClassPathXmlApplicationContext;
9  @SpringBootApplication
10 public class SpringLearnApplication {
11
12     public static void main(String[] args) {
13         SpringApplication.run(SpringLearnApplication.class, args);
14     }
15 }
```

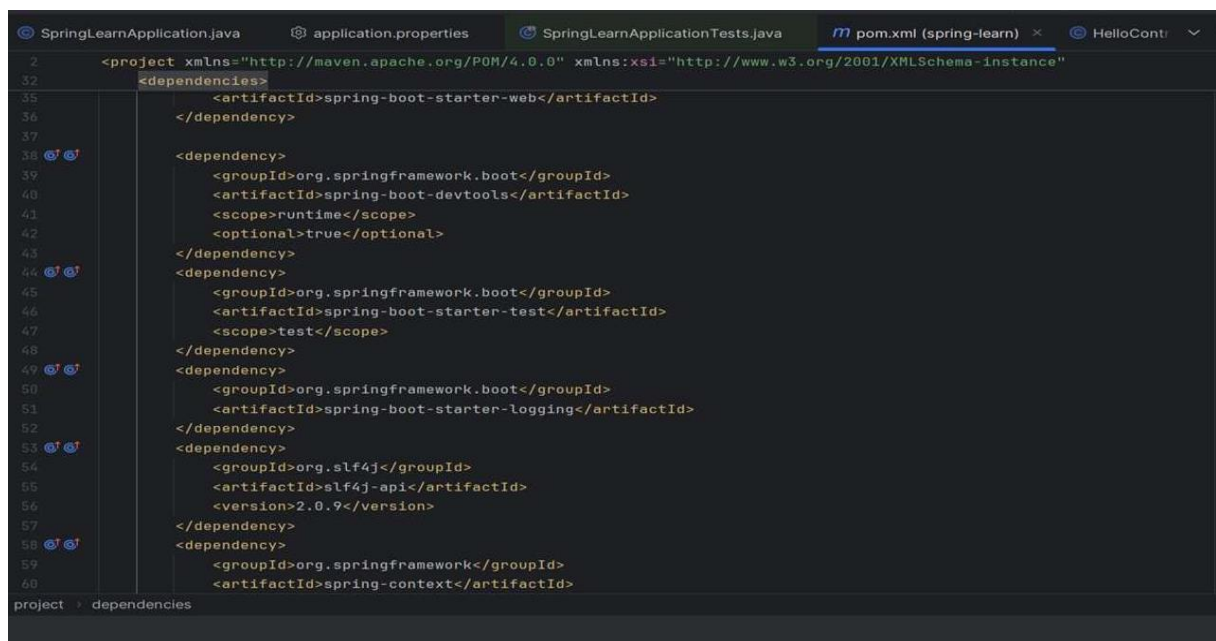
Application.properties



The screenshot shows an IDE with several tabs open: SpringLearnApplication.java, application.properties (selected), SpringLearnApplicationTests.java, pom.xml (spring-learn), and HelloContr. The application.properties file contains the following configuration:

```
1 spring.application.name=spring-learn
2 server.port=8083
3 logging.level.root=DEBUG
4 logging.level.com.cognizant.spring.learn=DEBUG
5 #logging.level.root=DEBUG
6
```

Pom.xml



The screenshot shows the same IDE with the pom.xml (spring-learn) tab selected. The file contains the following XML configuration:

```
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
32 <dependencies>
35     <artifactId>spring-boot-starter-web</artifactId>
36     </dependency>
37
38     <dependency>
39         <groupId>org.springframework.boot</groupId>
40         <artifactId>spring-boot-devtools</artifactId>
41         <scope>runtime</scope>
42         <optional>true</optional>
43     </dependency>
44     <dependency>
45         <groupId>org.springframework.boot</groupId>
46         <artifactId>spring-boot-starter-test</artifactId>
47         <scope>test</scope>
48     </dependency>
49     <dependency>
50         <groupId>org.springframework.boot</groupId>
51         <artifactId>spring-boot-starter-logging</artifactId>
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>org.springframework</groupId>
60         <artifactId>spring-context</artifactId>

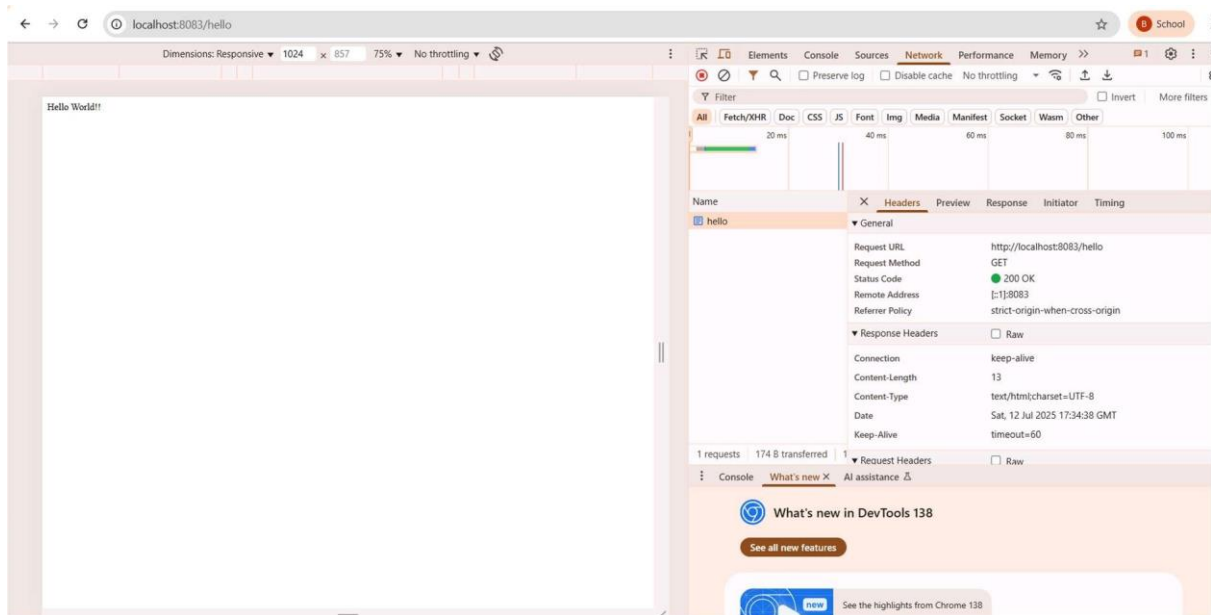
```

At the bottom of the IDE, a breadcrumb trail shows 'project > dependencies'.

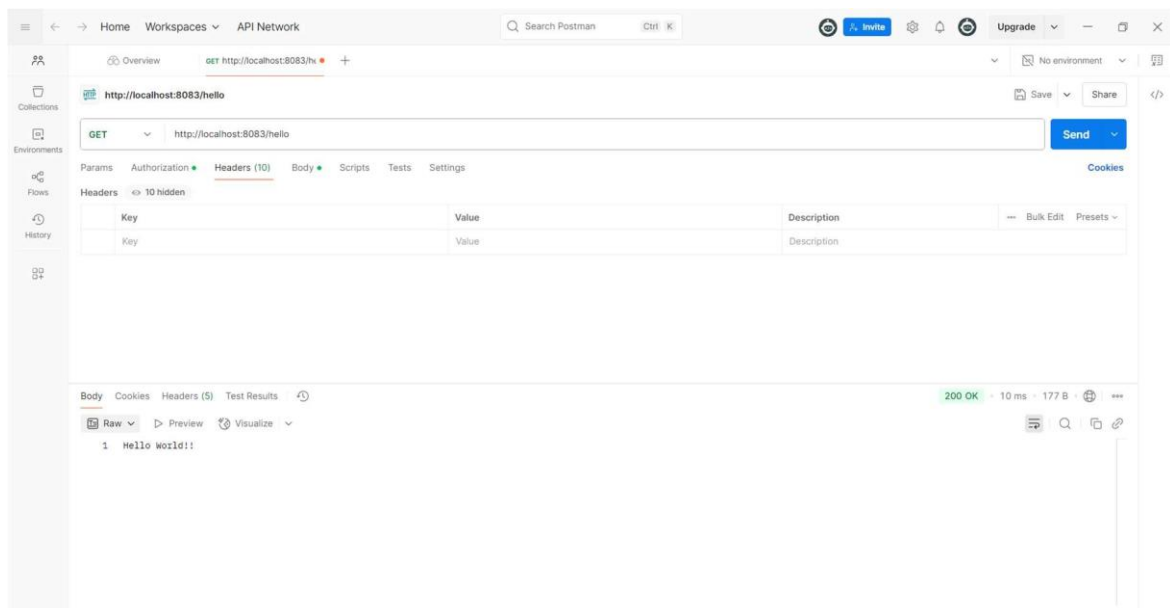
Console Output

```
Project ▾
Terminal Local × Command Prompt × + ▾
SpringLearnApplication.java application.properties SpringLearnApplicationTests.java pom.xml (spring-learn) × HelloConti...
2025-07-12T22:54:55.359+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-8] o.s.web.servlet.DispatcherServlet : GET "/hello", parameters={}
2025-07-12T22:54:55.360+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-8] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped to com.cognizant.spring_learn.controller.HelloContr
oller#sayHello()
2025-07-12T22:54:55.360+05:30 INFO 22944 --- [spring-learn] [nio-8083-exec-8] c.c.s.controller.HelloController : START: sayHello()
2025-07-12T22:54:55.360+05:30 INFO 22944 --- [spring-learn] [nio-8083-exec-8] c.c.s.controller.HelloController : END: sayHello()
2025-07-12T22:54:55.361+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-8] m.m.a.ResponseBodyMethodProcessor : Using 'text/html', 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 [text/plain, */*, application/json, application/*json
]
2025-07-12T22:54:55.361+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-8] m.m.a.ResponseBodyMethodProcessor : Writing ["Hello World!!"]
2025-07-12T22:54:55.362+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-8] o.s.web.servlet.DispatcherServlet : Completed 200 OK
2025-07-12T22:55:11.613+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-9] o.s.web.servlet.DispatcherServlet : GET "/hello", parameters={}
2025-07-12T22:55:11.614+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-9] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped to com.cognizant.spring_learn.controller.HelloContr
oller#sayHello()
2025-07-12T22:55:11.614+05:30 INFO 22944 --- [spring-learn] [nio-8083-exec-9] c.c.s.controller.HelloController : START: sayHello()
2025-07-12T22:55:11.614+05:30 INFO 22944 --- [spring-learn] [nio-8083-exec-9] c.c.s.controller.HelloController : END: sayHello()
2025-07-12T22:55:11.614+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-9] m.m.a.ResponseBodyMethodProcessor : Using 'text/plain', given [*/] and supported [text/plain,
*/*, application/json, application/*json]
2025-07-12T22:55:11.614+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-9] m.m.a.ResponseBodyMethodProcessor : Writing ["Hello World!!"]
2025-07-12T22:55:11.615+05:30 DEBUG 22944 --- [spring-learn] [nio-8083-exec-9] o.s.web.servlet.DispatcherServlet : Completed 200 OK
2025-07-12T22:55:54.182+05:30 DEBUG 22944 --- [spring-learn] [io-8083-exec-10] o.s.web.servlet.DispatcherServlet : GET "/hello", parameters={}
2025-07-12T22:55:54.182+05:30 DEBUG 22944 --- [spring-learn] [io-8083-exec-10] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped to com.cognizant.spring_learn.controller.HelloContr
oller#sayHello()
2025-07-12T22:55:54.183+05:30 INFO 22944 --- [spring-learn] [io-8083-exec-10] c.c.s.controller.HelloController : START: sayHello()
2025-07-12T22:55:54.183+05:30 INFO 22944 --- [spring-learn] [io-8083-exec-10] c.c.s.controller.HelloController : END: sayHello()
2025-07-12T22:55:54.184+05:30 DEBUG 22944 --- [spring-learn] [io-8083-exec-10] m.m.a.ResponseBodyMethodProcessor : Using 'text/plain', given [*/] and supported [text/plain,
*/*, application/json, application/*json]
2025-07-12T22:55:54.184+05:30 DEBUG 22944 --- [spring-learn] [io-8083-exec-10] m.m.a.ResponseBodyMethodProcessor : Writing ["Hello World!!"]
2025-07-12T22:55:54.185+05:30 DEBUG 22944 --- [spring-learn] [io-8083-exec-10] o.s.web.servlet.DispatcherServlet : Completed 200 OK
```

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

```
SpringLearnApplicationTests.java pom.xml (spring-learn) Country.java country.xml CountryController.java x v ...
package com.cognizant.spring_learn.controller;

import com.cognizant.spring_learn.model.Country;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController 1 usage
public class CountryController {
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class); 2 usages

    @RequestMapping("/country") no usages
    public Country getCountryIndia() {
        LOGGER.info("START: getCountryIndia()");
        ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = (Country) context.getBean("in");
        LOGGER.info("END: getCountryIndia()");
        return country;
    }
}
```

Model :Country

```
ies SpringLearnApplicationTests.java pom.xml (spring-learn) Country.java x country.xml CountryController.java v
1 package com.cognizant.spring_learn.model;
2
3 public class Country { 5 usages
4     private String code; 3 usages
5     private String name; 3 usages
6
7     public Country() {} no usages
8
9     public Country(String code, String name) { no usages
10         this.code = code;
11         this.name = name;
12     }
13
14
15     public String getCode() { return code; } no usages
16     public void setCode(String code) { this.code = code; } no usages
17
18     public String getName() { return name; } no usages
19     public void setName(String name) { this.name = name; } no usages
20 }
```

SpringLearnApplication

```
SpringLearnApplication.java  application.properties  SpringLearnApplicationTests.java  pom.xml (spring-learn)  Country.java

1  package com.cognizant.spring_learn;
2
3  import org.slf4j.Logger;
4  import org.slf4j.LoggerFactory;
5  import org.springframework.boot.SpringApplication;
6  import org.springframework.boot.autoconfigure.SpringBootApplication;
7  import org.springframework.context.ApplicationContext;
8  import org.springframework.context.support.ClassPathXmlApplicationContext;
9  @SpringBootApplication
10 public class SpringLearnApplication {
11
12     public static void main(String[] args) {
13         SpringApplication.run(SpringLearnApplication.class, args);
14     }
15 }
```

Country.xml

```
java  application.properties  SpringLearnApplicationTests.java  pom.xml (spring-learn)  Country.java  country.xml

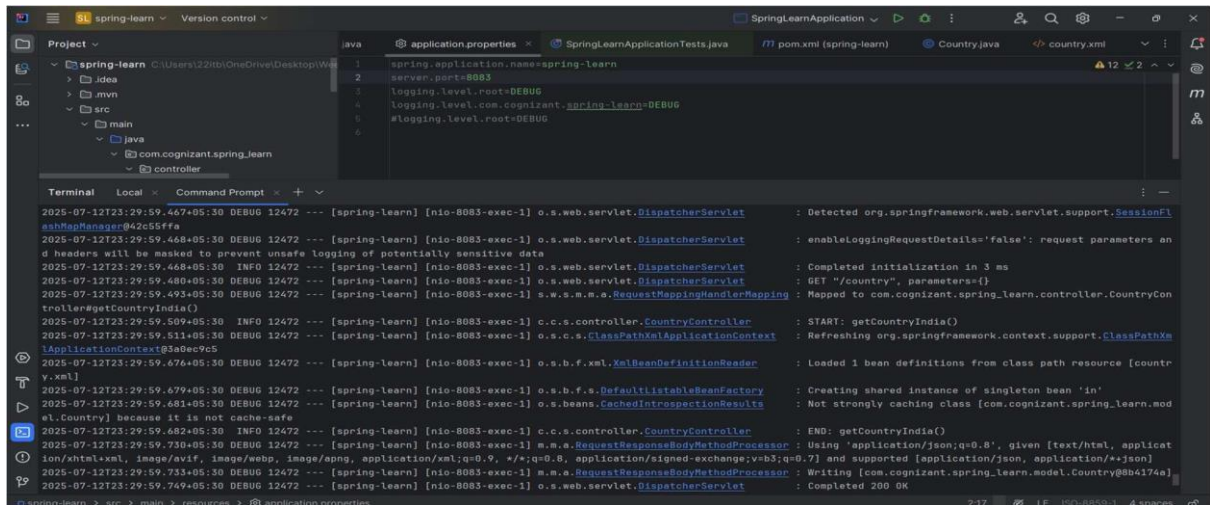
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4        xsi:schemaLocation="
5            http://www.springframework.org/schema/beans
6            http://www.springframework.org/schema/beans/spring-beans.xsd">
7
8      <bean id="in" class="com.cognizant.spring_learn.model.Country">
9          <property name="code" value="IN"/>
10         <property name="name" value="India"/>
11     </bean>
12
13 </beans>
14
```

Application.property

```
java  application.properties  SpringLearnApplicationTests.java  pom.xml (spring-learn)  Country.java  country.xml

1  spring.application.name=spring-learn
2  server.port=8083
3  logging.level.root=DEBUG
4  logging.level.com.cognizant.spring_learn=DEBUG
5  #logging.level.root=DEBUG
6
```

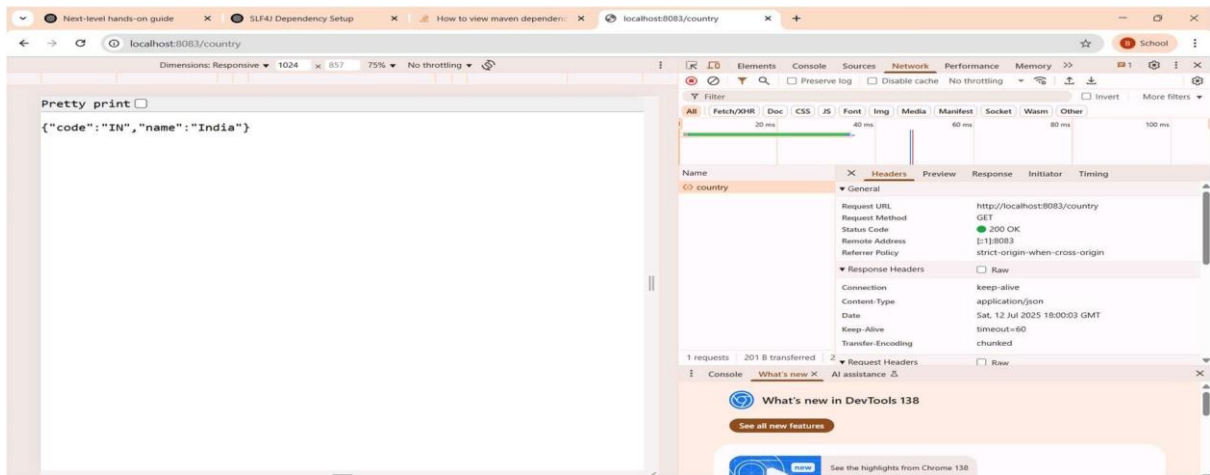

Console Output



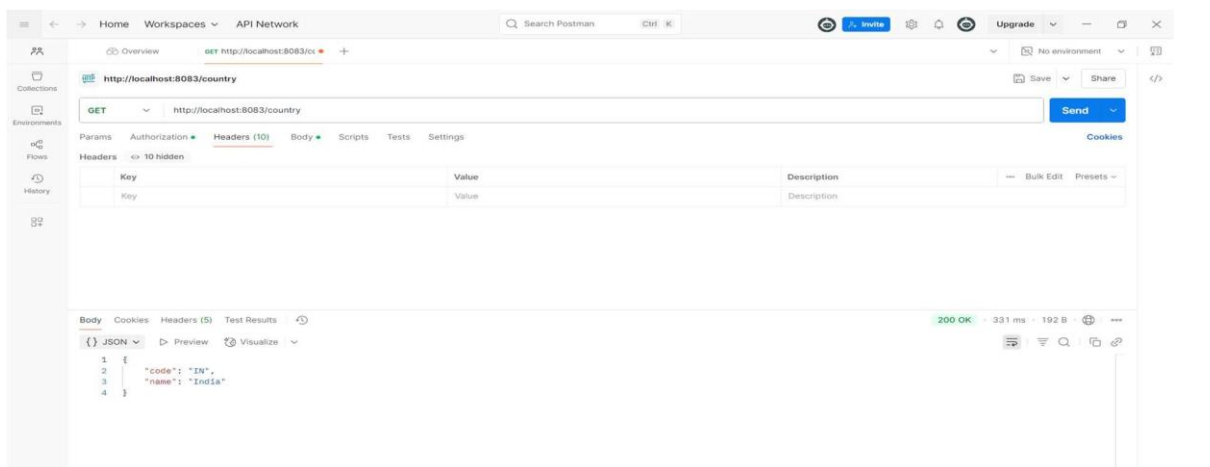
The screenshot shows the console output of a Spring Boot application. The logs are from a [spring-learn] application running on port 8083. The logs show the application starting up, including the detection of org.springframework.web.servlet.support.SessionFlashMapManager, the completion of initialization in 3 ms, the GET request for "/country", the mapping of the request to com.cognizant.spring_learn.controller.CountryController, the start of the getCountryIndia() method, the refreshing of the org.springframework.context.support.ClassPathXmlApplicationContext, the loading of 1 bean definitions from class path resource [country.xml], the creation of a shared instance of singleton bean 'in', the not strongly caching class [com.cognizant.spring_learn.model.Country] because it is not cache-safe, the end of the getCountryIndia() method, the writing of the response body, and the completion of the 200 OK response.

```
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.SessionFlashMapManager@42c555ff
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 and 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={}
2025-07-12T23:29:59.493:05:30 DEBUG 12472 --- [spring-learn] [nio-8083-exec-1] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped to com.cognizant.spring_learn.controller.CountryController::getCountryIndia()
2025-07-12T23:29:59.509:05:30 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.ClassPathXmlApplicationContext@3a8dec9c
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 [country.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.model.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,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]
2025-07-12T23:29:59.749:05:30 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

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

Service :CountryService

```
CountryService.java  SpringLearnApplicationTests.java  pom.xml (spring-learn)  Country.java  country.xml  Counti
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)).findFirst().orElse(null);
17     }
18 }
19
20
21 }
```


Controller: CountryController

```
1  import org.slf4j.LoggerFactory;
2  import org.springframework.beans.factory.annotation.Autowired;
3  import org.springframework.context.support.ClassPathXmlApplicationContext;
4  import org.springframework.web.bind.annotation.GetMapping;
5  import org.springframework.web.bind.annotation.PathVariable;
6  import org.springframework.web.bind.annotation.RequestMapping;
7  import org.springframework.web.bind.annotation.RestController;
8
9  @RestController
10 public class CountryController {
11
12     private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);
13
14     @Autowired
15     private CountryService countryService;
16
17     @GetMapping("/countries/{code}")
18     public Country getCountry(@PathVariable String code) {
19         LOGGER.info("START: getCountry() with code: " + code);
20         Country country = countryService.getCountry(code);
21         LOGGER.info("END: getCountry()");
22         return country;
23     }
24 }
```

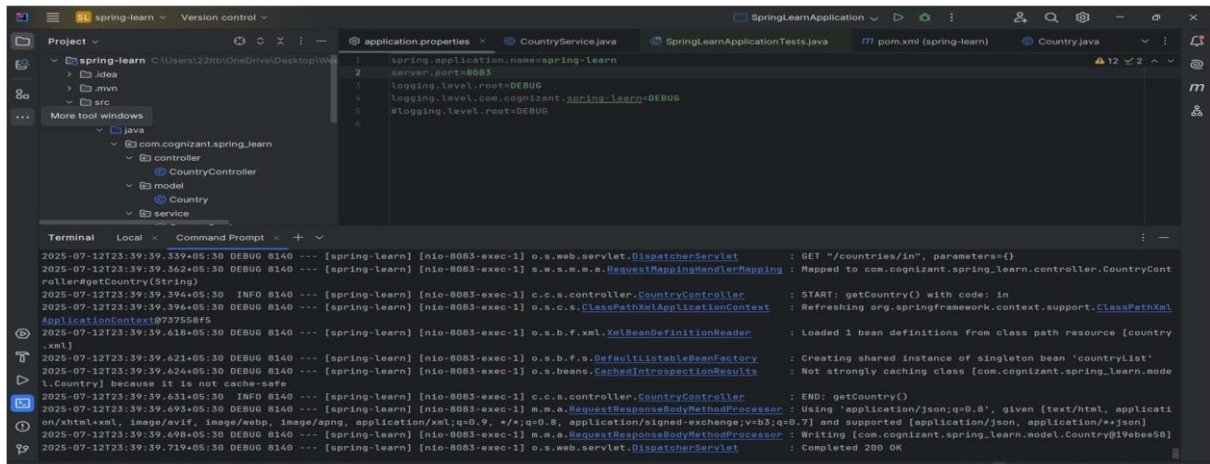
Country.xml

```
1  <beans xmlns="http://www.springframework.org/schema/beans"
2  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  xsi:schemaLocation="http://www.springframework.org/schema/beans
4  http://www.springframework.org/schema/beans/spring-beans.xsd">
5
6      <bean id="countryList" class="java.util.ArrayList">
7          <constructor-arg>
8              <list>
9                  <bean class="com.cognizant.spring_learn.model.Country">
10                     <property name="code" value="IN"/>
11                     <property name="name" value="India"/>
12                 </bean>
13                 <bean class="com.cognizant.spring_learn.model.Country">
14                     <property name="code" value="US"/>
15                     <property name="name" value="United States"/>
16                 </bean>
17                 <bean class="com.cognizant.spring_learn.model.Country">
18                     <property name="code" value="UK"/>
19                     <property name="name" value="United Kingdom"/>
20                 </bean>
21             </list>
22         </constructor-arg>
23     </bean>
24 </beans>
```

Application.xml

```
1  spring.application.name=spring-learn
2  server.port=8083
3  logging.level.root=DEBUG
4  logging.level.com.cognizant.spring_learn=DEBUG
5  #logging.level.root=DEBUG
6
```

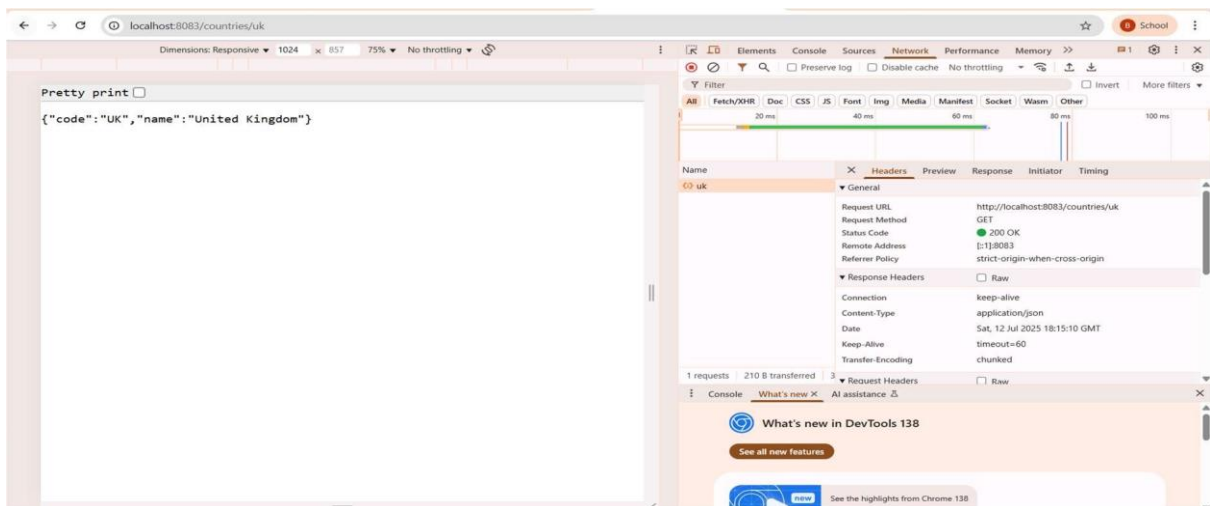
Console Output



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:

- 2025-07-12T23:39:39.339+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : GET "/countries/in", parameters={}
- 2025-07-12T23:39:39.362+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped to com.cognizant.spring_learn.controller.CountryController::getCountry(String)
- 2025-07-12T23:39:39.394+05:30 INFO 8140 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.CountryController : START: getCountry() with code: in
- 2025-07-12T23:39:39.396+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] o.s.c.s.ClassPathXmlApplicationContext : Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@9737558f5
- 2025-07-12T23:39:39.618+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] o.s.b.f.xml.XmlBeanDefinitionReader : Loaded 1 bean definitions from class path resource [country.xml]
- 2025-07-12T23:39:39.621+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] o.s.b.f.s.DefaultListableBeanFactory : Creating shared instance of singleton bean 'countryList'
- 2025-07-12T23:39:39.624+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] o.s.beans.CachedIntrospectionResults : Not strongly caching class [com.cognizant.spring_learn.model.Country] because it is not cache-safe
- 2025-07-12T23:39:39.631+05:30 INFO 8140 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.CountryController : END: getCountry()
- 2025-07-12T23:39:39.693+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] m.m.a.RequestMappingHandlerMethod : 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=b2;q=0.7] and supported [application/json, application/*+json]
- 2025-07-12T23:39:39.698+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] m.m.a.RequestMappingHandlerMethod : Writing [com.cognizant.spring_learn.model.Country@19abee58]
- 2025-07-12T23:39:39.719+05:30 DEBUG 8140 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : Completed 200 OK

Web Output



Postman output

