**Hello World RESTful Web Service** Spring Boot task

**✅ Problem Statement**

Create a RESTful service using Spring Web Framework in the existing SpringLearn application that returns the text “Hello World!!” on calling the endpoint /hello.

**✅ Steps**

1. Created a new class named HelloController under the package com.cognizant.spring\_learn.controller.
2. Annotated the class with @RestController to enable RESTful behavior.
3. Added a method sayHello() inside HelloController, annotated with @RequestMapping(path="/hello", method=RequestMethod.GET).
4. Implemented sayHello() to return the hardcoded string Hello World!!.
5. Added log statements using LOGGER.debug() at the start and end of the sayHello() method.
6. Updated the application.properties file to set server.port=8083 so the application runs on port 8083.
7. Ran the application and accessed http://localhost:8083/hello via Chrome and Postman.
8. Verified the output response and checked HTTP headers in browser DevTools and Postman’s Headers tab.

**CODE:**

* **HelloController.java**

package com.cognizant.spring\_learn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.web.bind.annotation.\*;

@RestController

public class HelloController {

private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);

@RequestMapping(path = "/hello", method = RequestMethod.GET)

public String sayHello() {

LOGGER.debug("Start sayHello()");

String response = "Hello World!!";

LOGGER.debug("End sayHello()");

return response;

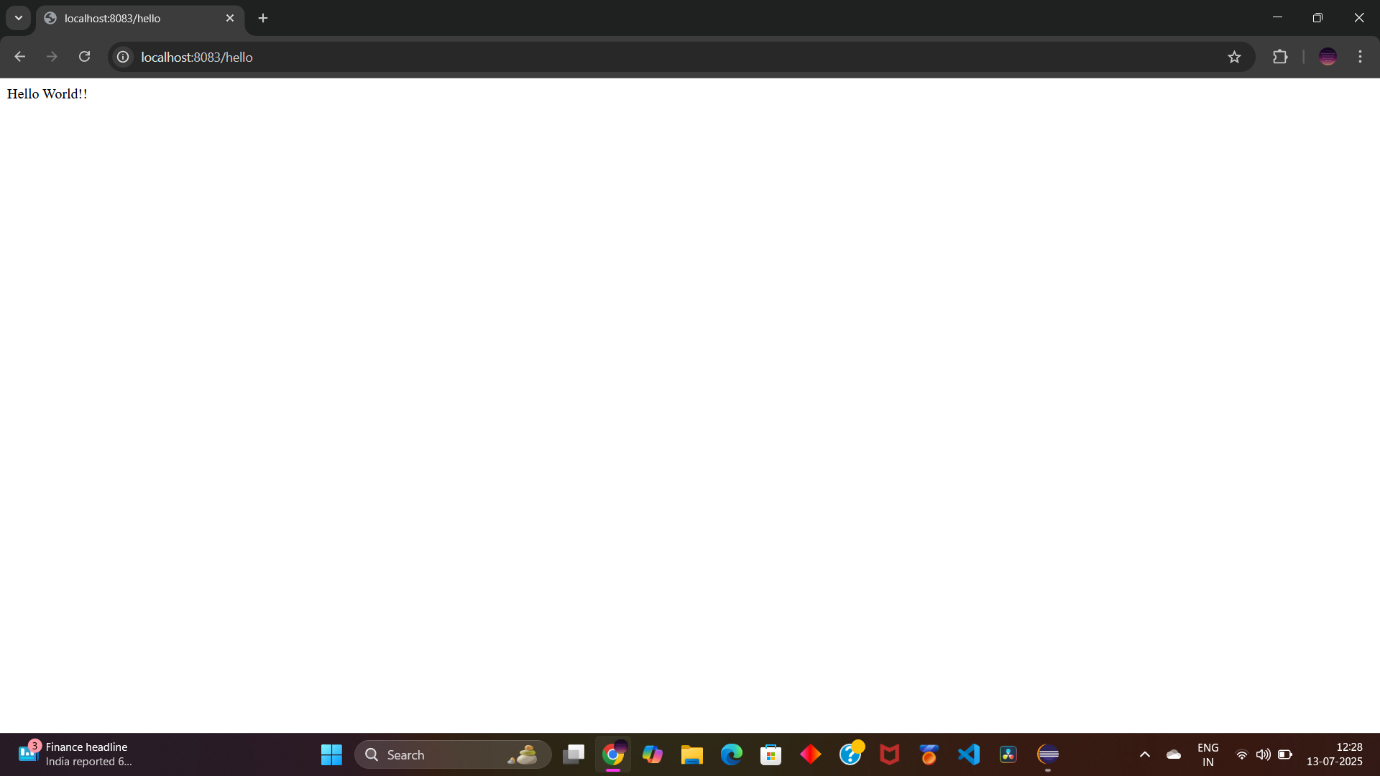
}

}

* **src/main/resources/application.properties**

**server.port=8083**

**✅ Output:**

****

*Accessed via:* [*http://localhost:8083/hello*](http://localhost:8083/hello)

**REST - Country Web Service:**

✅ Problem Statement

\*\*Create a REST endpoint\*\* in Spring Boot that returns a `Country` bean (India) loaded from an XML configuration file.

---

## Steps:

1. Create `Country` model class.

2. Create `country.xml` Spring config with a bean.

3. Create `CountryController` to return the bean via REST.

4. Load the bean using `ClassPathXmlApplicationContext`.

5. Return it as JSON from controller.

---

## Code

### 1. `Country.java` (Model)

`src/main/java/com/cognizant/spring\_learn/Country.java`

```java

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = name;

}

public String getCode() { return code; }

public void setCode(String code) { this.code = code; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

}

```

---

### 2. `country.xml`

`src/main/resources/country.xml`

```xml

<?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

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="country" class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="IN" />

<property name="name" value="India" />

</bean>

</beans>

```

---

### 3. `CountryController.java`

`src/main/java/com/cognizant/spring\_learn/controller/CountryController.java`

```java

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.model.Country;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@RestController

public class CountryController {

private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START");

ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = (Country) context.getBean("country");

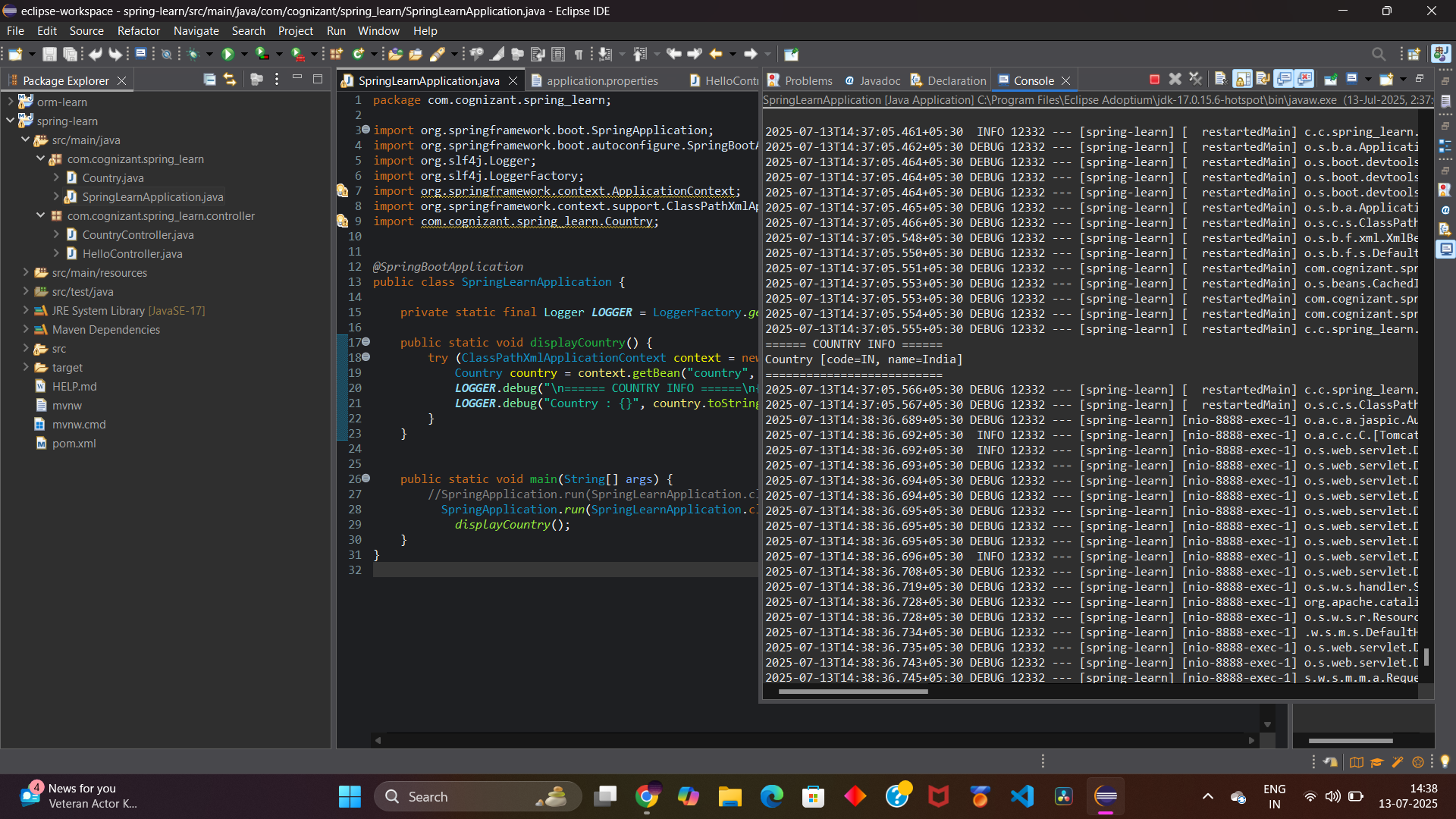
LOGGER.info("END");

return country;

}

}

## Sample Output (JSON)



**REST - Get country based on country code**

**✅ Problem Statement**

Build a Spring Boot REST endpoint /countries/{code} that returns country details based on the given code (case-insensitive), using a list of country beans defined in country.xml.

**✅ Folder & Package Structure Used**

src/

├── main/

│ ├── java/

│ │ └── com.cognizant.spring\_learn/

│ │ ├── Country.java

│ │ ├── SpringLearnApplication.java

│ │ ├── controller/

│ │ │ └── CountryController.java

│ │ └── service/

│ │ └── CountryService.java

│ └── resources/

│ └── country.xml

**✅ Implementation Steps**

1. **Define the Country POJO**
   * Contains code and name fields.
   * Located in: com.cognizant.spring\_learn.Country
2. **Create country.xml with country beans**
   * Located in: src/main/resources/country.xml
   * Defines a list of countries using Spring <bean> configuration.
3. **Create CountryService**
   * Loads countryList bean from country.xml.
   * Searches the country list using case-insensitive match.
   * Returns the matched Country object.
4. **Create REST Controller**
   * Maps @GetMapping("/countries/{code}").
   * Gets the code using @PathVariable.
   * Calls countryService.getCountry(code).

**✅ Key Code Snippets**

**Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

// Getters, Setters, Constructor

}

**country.xml**

<?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

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.cognizant.spring\_learn.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

<bean class="com.cognizant.spring\_learn.Country">

<property name="code" value="US"/>

<property name="name" value="United States"/>

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**CountryService.java**

@Service

public class CountryService {

public Country getCountry(String code) {

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

@SuppressWarnings("unchecked")

List<Country> countryList = (List<Country>) context.getBean("countryList");

return countryList.stream()

.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null);

}

}

**CountryController.java**

@RestController

public class CountryController {

@Autowired

private CountryService countryService;

@GetMapping("/countries/{code}")

public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

}

**✅ Sample Request**

GET http://localhost:8083/countries/in

**✅ Expected Output**

{

"code": "IN",

"name": "India"

}

