Hands-on: REST - Country Web Service

# 1. Objective

To create a REST endpoint in the existing Spring Boot application that returns India's country details by reading from a Spring XML configuration file.

# 2. Implementation Steps

• Create a Spring XML file named country.xml under src/main/resources.

• Define a bean with id='country' and values for 'code' and 'name'.

• Create a POJO class 'Country.java' with fields, constructor, getters, and setters.

• Create a controller class 'CountryController.java' with endpoint '/country'.

• Load the bean from XML using ClassPathXmlApplicationContext and return it from the method.

# 3. country.xml

<bean id="country" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
</bean>

# 4. Country.java

public class Country {  
 private String code;  
 private String name;  
  
 public Country() {}  
  
 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;  
 }  
}

# 5. CountryController.java

@RestController  
public class CountryController {  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country");  
 return country;  
 }  
}

# 6. Output & Testing

• Run the application and hit: http://localhost:8083/country

• Output:



• You can test this in Chrome browser or Postman.

# 7. HTTP Headers Inspection

• In Chrome Developer Tools > Network tab, inspect the /country request.

• Check Response Headers like Content-Type: application/json.

• In Postman, use GET on /country and inspect Headers tab to see response headers.

# 8. What Happens Internally

• Spring automatically calls the controller method mapped to /country.

• It loads the Spring XML configuration and fetches the Country bean.

• Spring Boot uses Jackson (built-in) to convert the Country object to JSON.

# 9. Conclusion

This hands-on demonstrates how to return a bean loaded from Spring XML config as a JSON response using Spring Boot REST API. It also covers practical testing and header inspection using browser tools and Postman.