REST – Country Web Service

# Objective:

Create a Spring Boot REST controller that returns India’s country details from an XML bean configuration.

# 1. Sample Request and Response

URL: http://localhost:8083/country

HTTP Method: GET

Sample Response:

{  
 "code": "IN",  
 "name": "India"  
}

# 2. Controller Method

File: CountryController.java

Package: com.cognizant.spring\_learn.controller

Annotation: @RequestMapping("/country")

Method Name: getCountryIndia()

What happens in the controller method:

- The method is triggered when a GET request is made to /country.  
- It uses Spring’s ApplicationContext to load the XML configuration file country.xml.  
- It retrieves a bean named 'in', which represents the Country object for India.  
- The object is returned as a response.

# 3. Bean Definition in XML

File: country.xml

Location: src/main/resources

<bean id="in" class="com.cognizant.spring\_learn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
</bean>

# 4. How the Bean Is Converted to JSON

- Spring Boot uses Jackson internally.  
- The @RestController annotation tells Spring that the returned object should be converted to JSON.  
- MappingJackson2HttpMessageConverter handles the object-to-JSON conversion automatically.  
- No need to write serialization logic.

# 5. HTTP Header Details in Developer Tools (Browser)

When the URL is accessed in a browser (F12 → Network → country → Headers):

Request Headers:

GET /country HTTP/1.1  
Host: localhost:8083  
Accept: application/json  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

Response Headers:

Content-Type: application/json  
Date: [current date]  
Content-Length: 34

# 6. HTTP Header Details in Postman

After sending a GET request to http://localhost:8083/country, click the Headers tab in Postman.

Typical Headers:

Content-Type: application/json  
Transfer-Encoding: chunked  
Date: [current date]