REST - Country Web Service Assignment

# 1. Introduction

This assignment demonstrates a RESTful web service created using Spring Boot. It returns details of a country (India) configured as a Spring bean in an XML file.

# 2. REST Service Details

• URL: http://localhost:8083/country

• Controller: com.cognizant.spring-learn.controller.CountryController

• Method Annotation: @RequestMapping

• Method Name: getCountryIndia()

• Implementation: Load 'India' bean from Spring XML configuration and return

# 3. Code Snippets

## 3.1 Country.java

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; }  
}

## 3.2 country.xml

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

## 3.3 CountryController.java

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

# 4. Explanation of Key Aspects

## 4.1 What happens in the controller method?

When a request is made to /country, the controller method loads the Spring context, retrieves the bean with ID 'in', and returns the Country object.

## 4.2 How is the bean converted into JSON?

Spring Boot uses Jackson to automatically convert the returned Country object to JSON. This happens because of the @RestController annotation which implies @ResponseBody.

## 4.3 HTTP Headers in Browser Developer Tools

In the browser, open Developer Tools (F12), go to Network tab, select the /country request. Headers like 'Content-Type: application/json' and 'Status: 200 OK' will be shown.

## 4.4 HTTP Headers in Postman

In Postman, send a GET request to /country. Under the 'Headers' tab in the response, headers such as 'Content-Type: application/json' and 'Transfer-Encoding: chunked' can be seen.

# 5. Sample Output

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