## **REST - Country Web Service**

## **1️ Objective**

Create a RESTful API in the existing Spring Learn Application that:

* Loads the India country bean from the country.xml Spring configuration.
* Returns the country details as JSON.

## 2 **Implementation Steps**

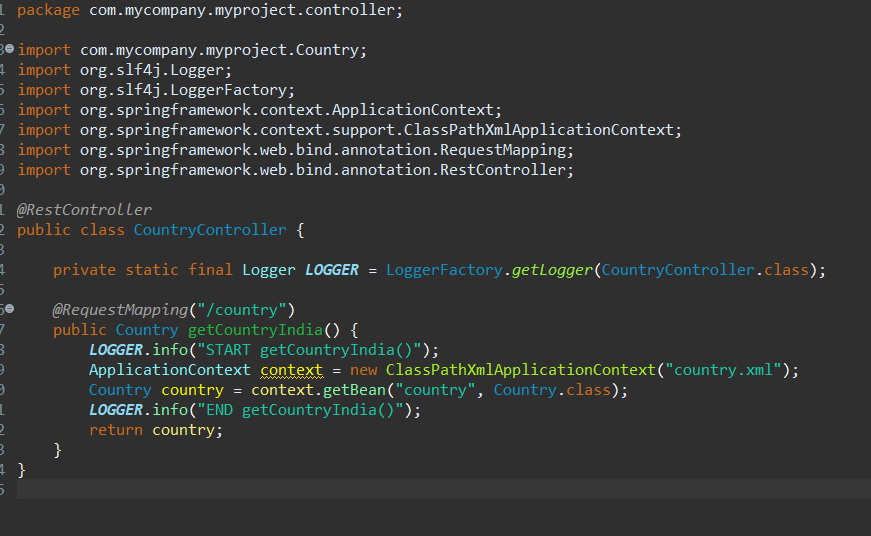
### **Step 1: Create CountryController**

Path: src/main/java/com/mycompany/myproject/controller/CountryController.java

package com.mycompany.myproject.controller;  
  
import com.mycompany.myproject.Country;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class CountryController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 LOGGER.info("START getCountryIndia()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
 LOGGER.info("END getCountryIndia()");  
 return country;  
 }  
}

What Happens:

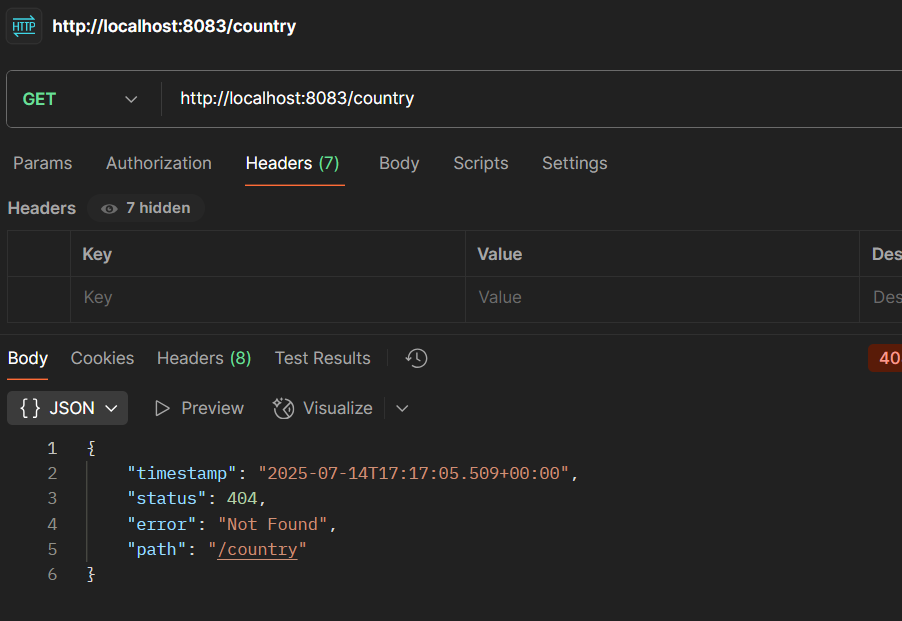
* @RestController → Marks it as REST API.
* @RequestMapping("/country") → Maps GET request to /country.
* getCountryIndia() → Loads the Country bean from XML and returns it.



### **Step 2: Sample Request & Response**

* Request: GET http://localhost:8083/country
* Sample JSON Response:
* {  
   "code": "IN",  
   "name": "India"  
  }

Spring Boot automatically converts the Country object to JSON using Jackson.



### **Step 3: SME Explanations**

|  | Explanation |
| --- | --- |
| What happens in the controller? | Loads Country bean from country.xml using Spring IoC and returns the object. |
| How converted to JSON? | spring-boot-starter-web includes Jackson. Spring automatically serializes the Java object to JSON for HTTP response. |
| Check DevTools → Network: | Shows Request Headers (GET, Host, User-Agent, Accept) and Response Headers (Content-Type: application/json). |
| Check Postman → Headers: | Confirms Content-Type: application/json, Content-Length, Date. |

### **Step 4: Verify in Browser & Postman**

Postman:

1. Send GET http://localhost:8083/country
2. Click Headers tab in response
   * Confirms Content-Type: application/json
   * Content-Length
   * Date

