**Hands-On 5: REST - Country Web Service**

**Requirements :**

* **URL**: http://localhost:8083/country
* **Controller**: com.cognizant.spring\_learn.controller.CountryController
* **Method**: getCountryIndia()
* **Returns**: A Country object from country.xml
* **Response**:

**json**

{

"code": "IN",

"name": "India"

}

**Check pom.xml**

Ensure you **have this exact spring-boot-starter-web**:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

**Application Properties:**

spring.application.name=spring-learn

logging.level.root=DEBUG

logging.level.com.cognizant=DEBUG

server.port=8083

**Sample Controller Code:**

package com.cognizant.spring\_learn.controller;

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

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

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.cognizant.spring\_learn.Country;

@RestController

public class CountryController {

@RequestMapping("/country")

public Country getCountryIndia() {

try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml")) {

return context.getBean("country", Country.class);

}

} }

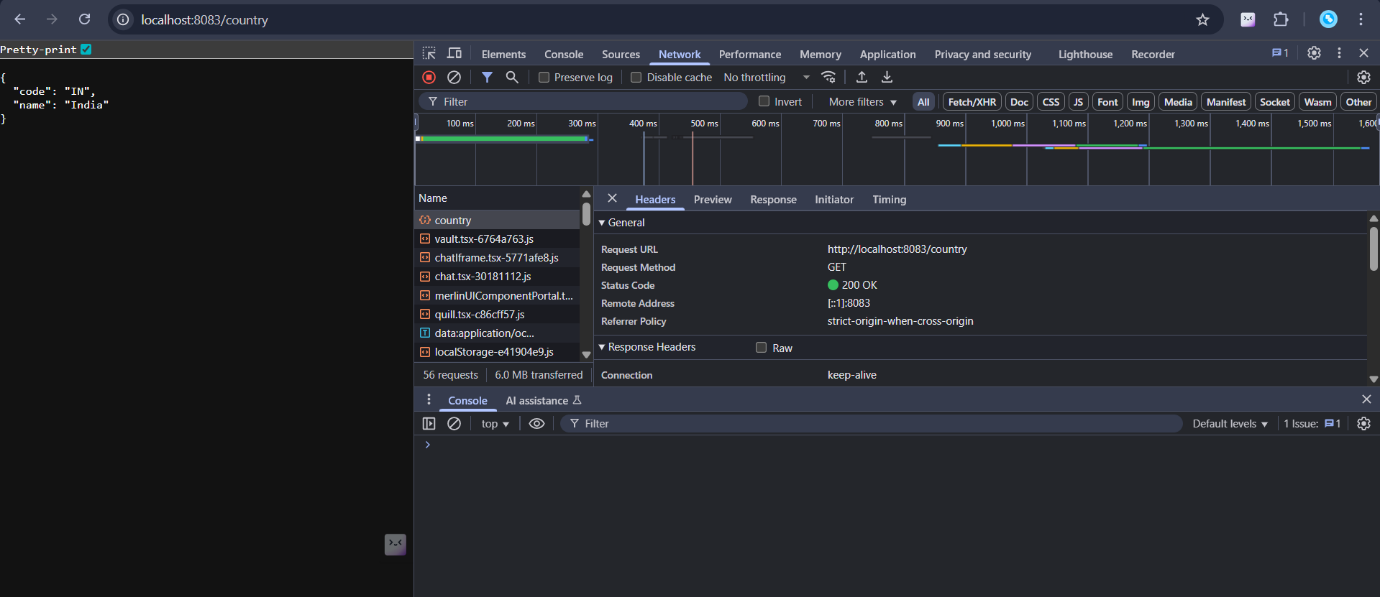
**What Happens in the Controller?**

* The method loads country.xml (defined under src/main/resources) using ClassPathXmlApplicationContext.
* It retrieves the bean with ID country, which contains India's details.
* The Country object is returned directly — Spring Boot automatically serializes it to **JSON** (using Jackson under the hood).

**How is the Bean Converted to JSON?**

* Spring Boot uses **Jackson**, a JSON serialization library.
* Your Country class must be a proper Java bean (POJO) with getters and setters.
* When you return the object, Jackson serializes it into JSON for the HTTP response.

**In Browser Developer Tools (F12 > Network Tab):**



**In Postman:**

