**Spring REST using Spring Boot3**

REST - Country Web Service

**Implementation:**

CountryController.java

package com.cognizant.spring\_learn.controller;

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.RequestMethod;

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

import com.cognizant.spring\_learn.Country;

@RestController

public class CountryController {

private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

@RequestMapping(value = "/country", method = RequestMethod.GET)

public Country getCountryIndia() {

LOGGER.info("START - getCountryIndia()");

// Load the Spring XML and pull the "country" bean

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

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

LOGGER.info("END - getCountryIndia()");

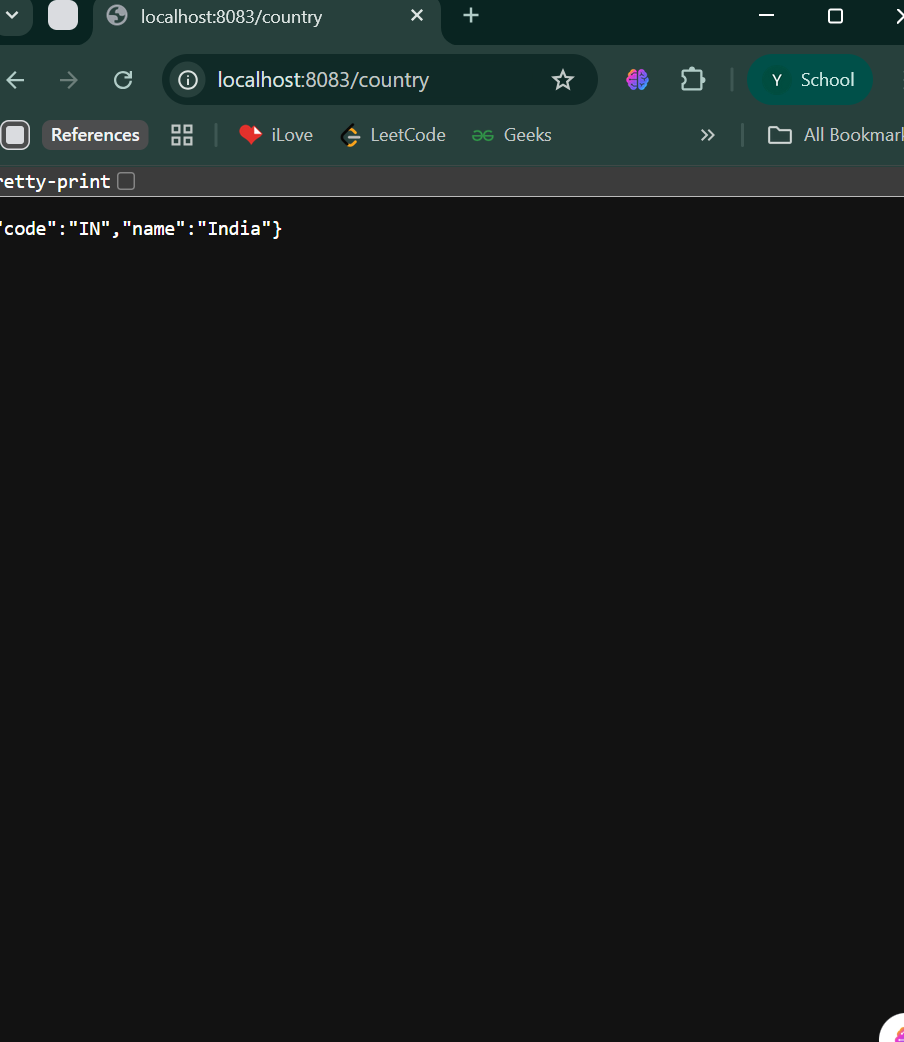
return country;

}

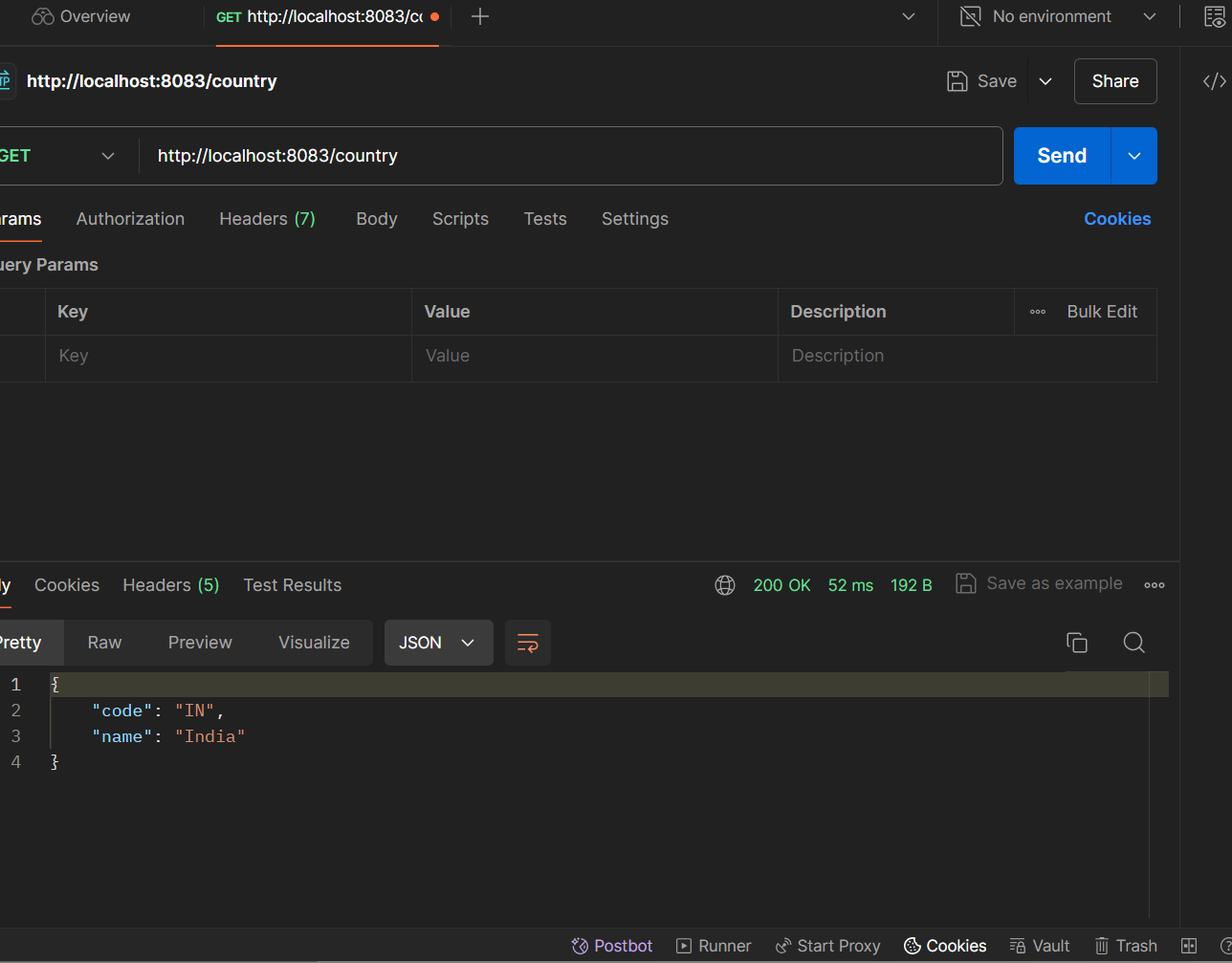
}

**OUTPUT:**

In Chrome

****

In PostMan

****

**SME Walkthrough :**

1. **What happens in the controller method?**

When a GET request is made to /country, the Spring DispatcherServlet matches it to getCountryIndia() in CountryController.

Inside the method:

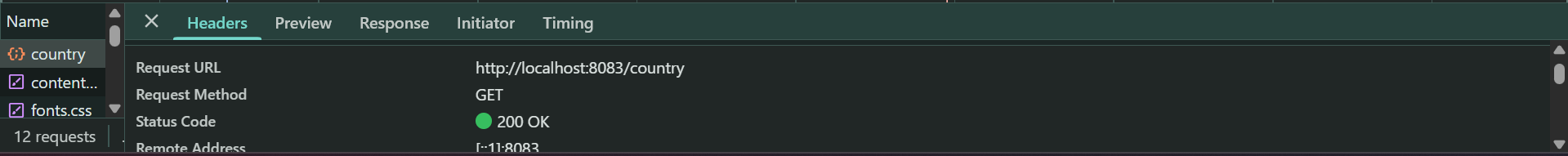
* The Spring XML file country.xml is loaded.
* The Country bean is retrieved using its ID: "country".
* The bean is returned directly by the method.

2. **How is the bean converted into a JSON response?**

The method returns a Country Java object. Spring Boot automatically converts it into JSON.

**3. HTTP Header Inspection**

Chrome -> Inspect-> Network tab->Headers

****

PostMan-> Headers tab

