**INITIALIZING SPRING PROJECT**

A screenshot of a computer

AI-generated content may be incorrect.

**FILES STRUCTURE**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Country.java**

package com.cognizant.spring\_learns3.country\_rest.model;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country

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

private String code;

private String name

public Country() {

*LOGGER*.debug("Inside Country Constructor.");

}

public String getCode() {

*LOGGER*.debug("Inside getCode()");

return code;

}

public void setCode(String code) {

*LOGGER*.debug("Inside setCode()");

this.code = code;

}

public String getName() {

*LOGGER*.debug("Inside getName()");

return name;

}

public void setName(String name) {

*LOGGER*.debug("Inside setName()");

this.name = name;

}

*@Override*

public String toString() {

return "Country{code='" + code + "', name='" + name + "'}";

}

}

**CountryController.java**

package com.cognizant.spring\_learns3.country\_rest.controller;

import com.cognizant.spring\_learns3.country\_rest.model.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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***.debug("START");

try (ClassPathXmlApplicationContext context =

new ClassPathXmlApplicationContext("country.xml")) {

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

***LOGGER***.debug("END");

return country;

}

}

}

**CountryRestApplication.java**

package com.cognizant.spring\_learns3.country\_rest;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryRestApplication {

public static void main(String[] args) {

SpringApplication.run(CountryRestApplication.class, args);

}

}

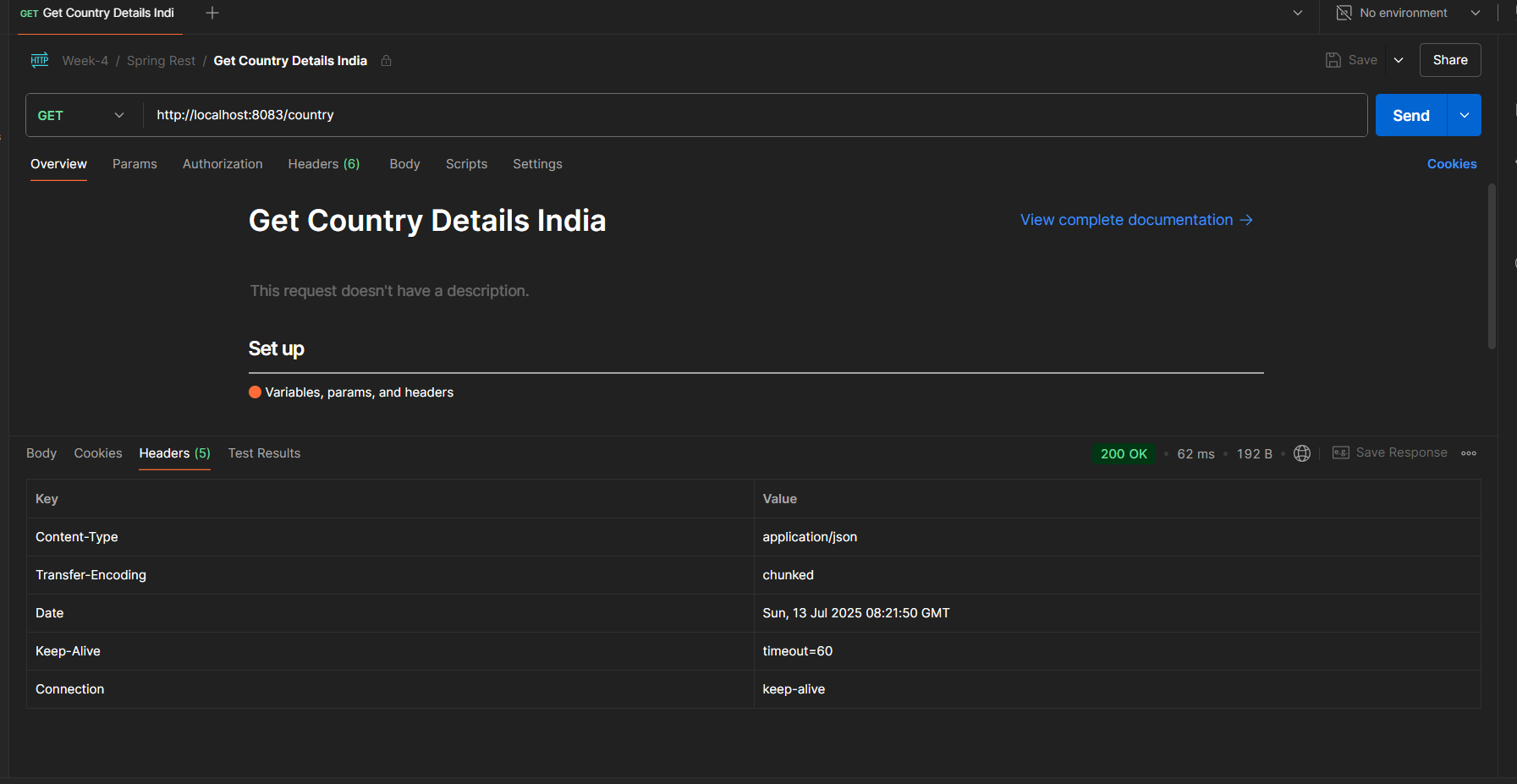
A screen shot of a computer

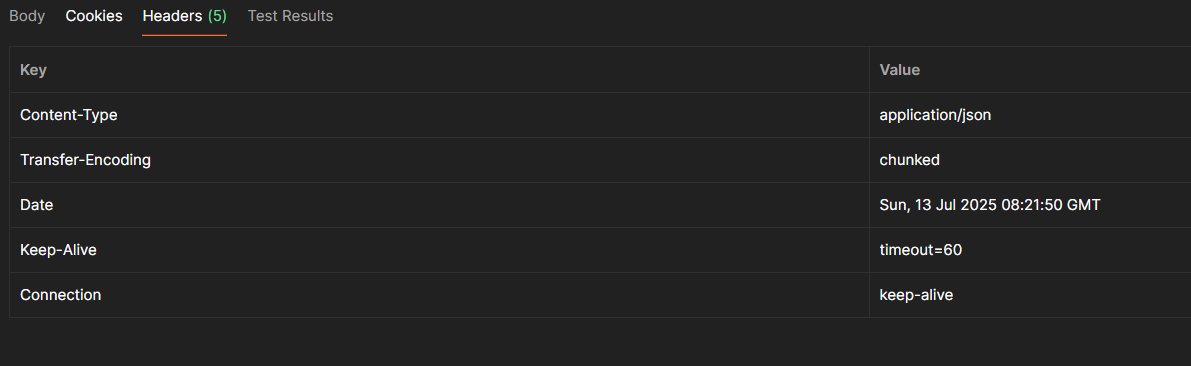
AI-generated content may be incorrect.

**Running in the Localhost google chrome**

A screenshot of a computer

AI-generated content may be incorrect.

Using POSTMAN to get the outputA screenshot of a computer

AI-generated content may be incorrect.

When a client (like a browser or Postman) makes a GET request to the /country endpoint, Spring identifies the matching controller method using the @RequestMapping("/country") annotation. The method getCountryIndia() is executed. Inside the method, the country.xml file is loaded using ClassPathXmlApplicationContext, and the bean with ID "in" is retrieved. This bean is a Java object (Country), which is then returned by the method. Spring handles the response conversion automatically and sends it back to the client.

**How the bean is converted into JSON response?**

Spring Boot uses a library called **Jackson** under the hood. When a Java object (like Country) is returned from a @RestController, Spring checks the Accept header in the request (typically application/json) and uses Jackson to serialize the Java object into a JSON format. No manual conversion is needed. The @RestController and Jackson integration take care of it seamlessly.