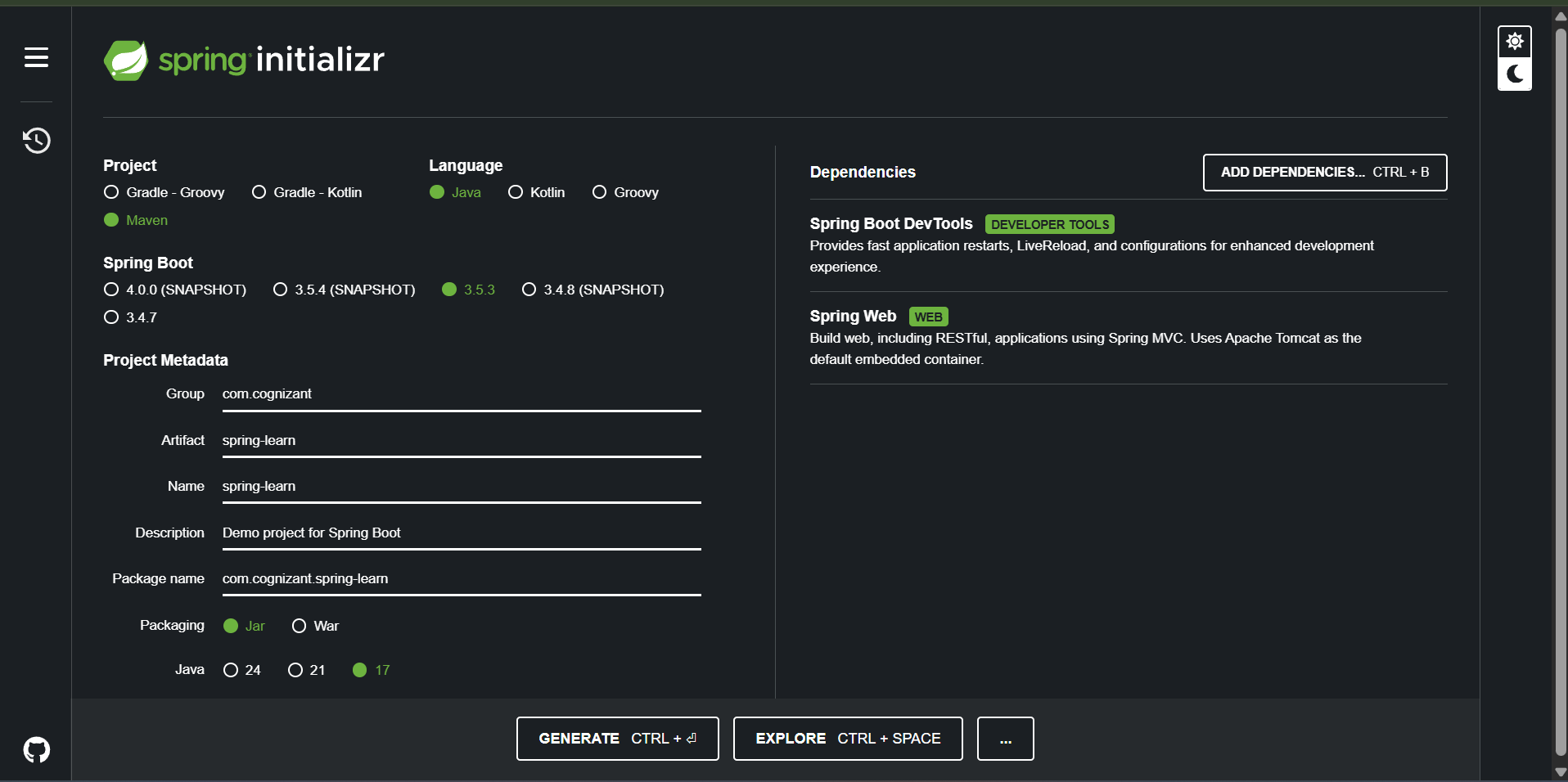
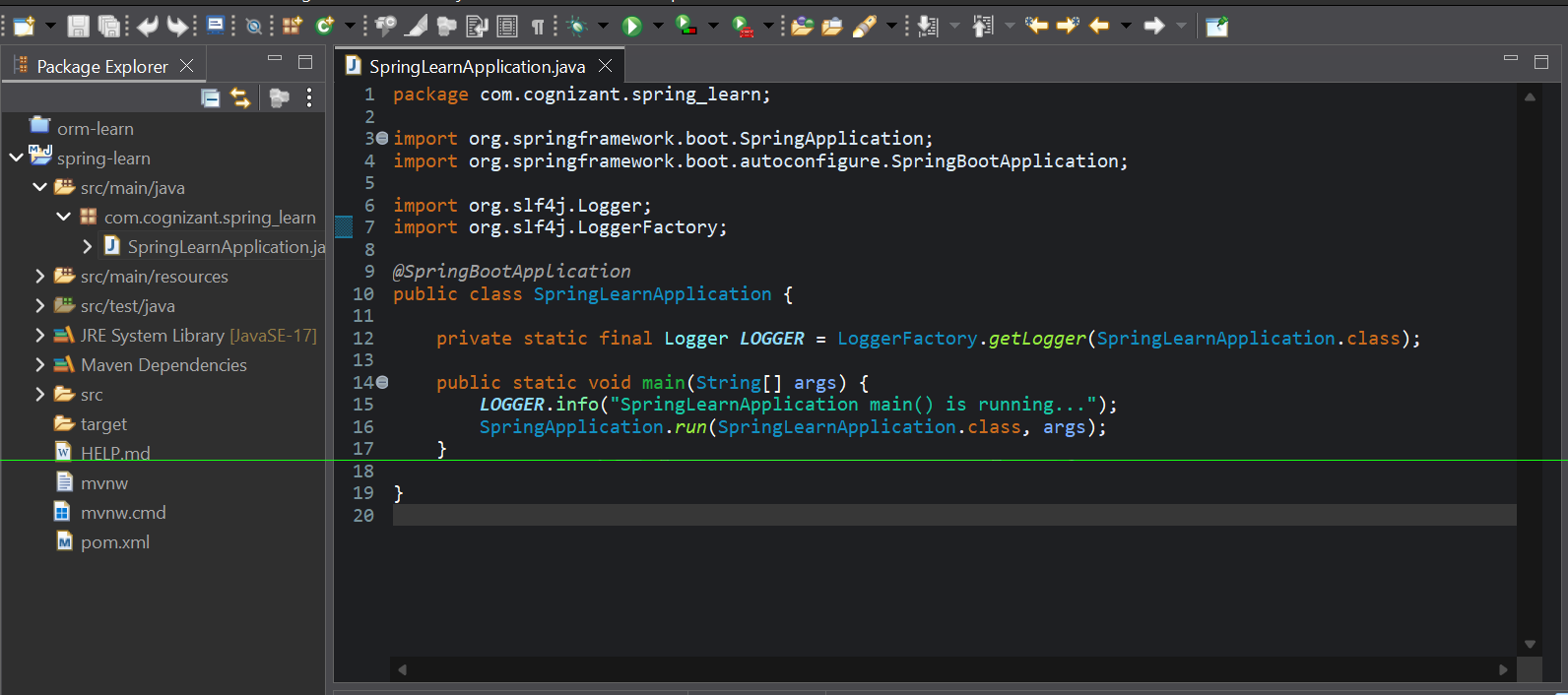
**Hands on 1**

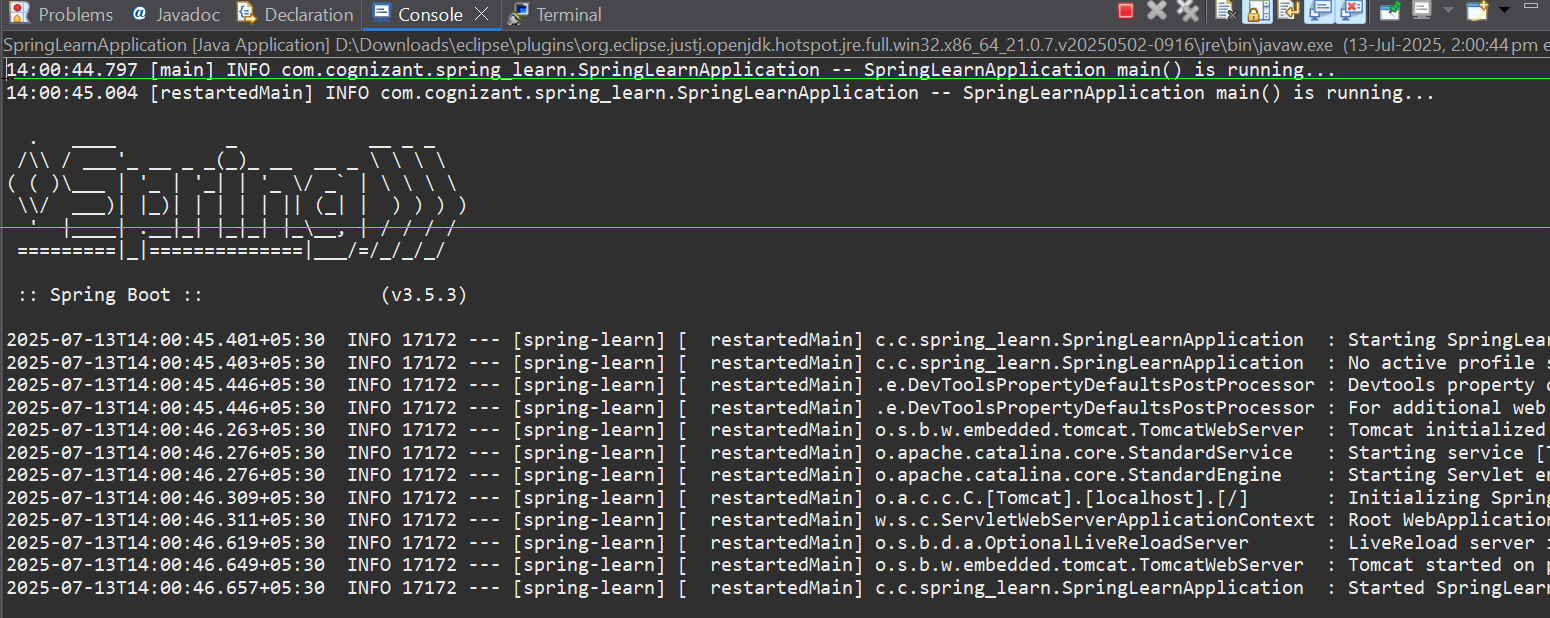
**Create a Spring Web Project using Maven**Configuring the Project,



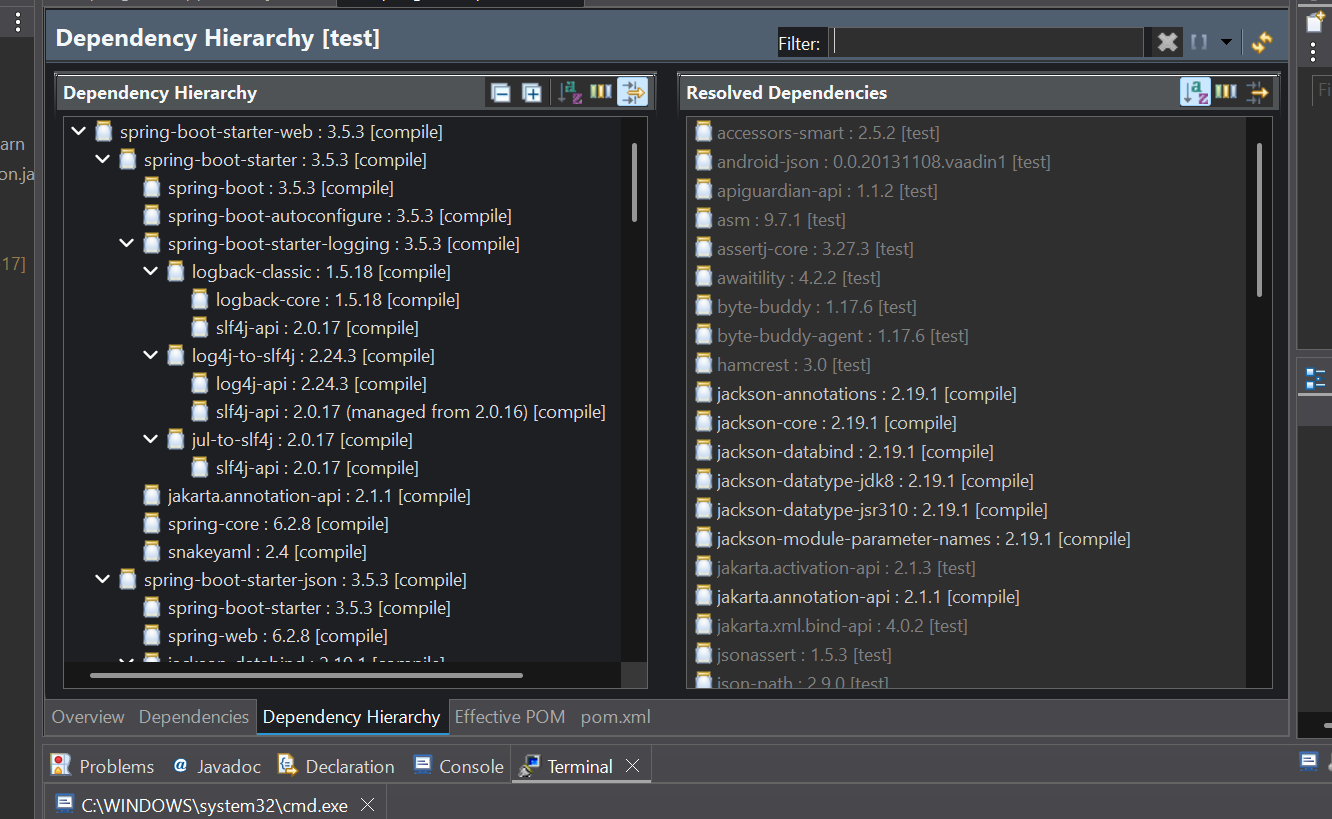
Importing to workspace and adding logger to the main



Output of the application

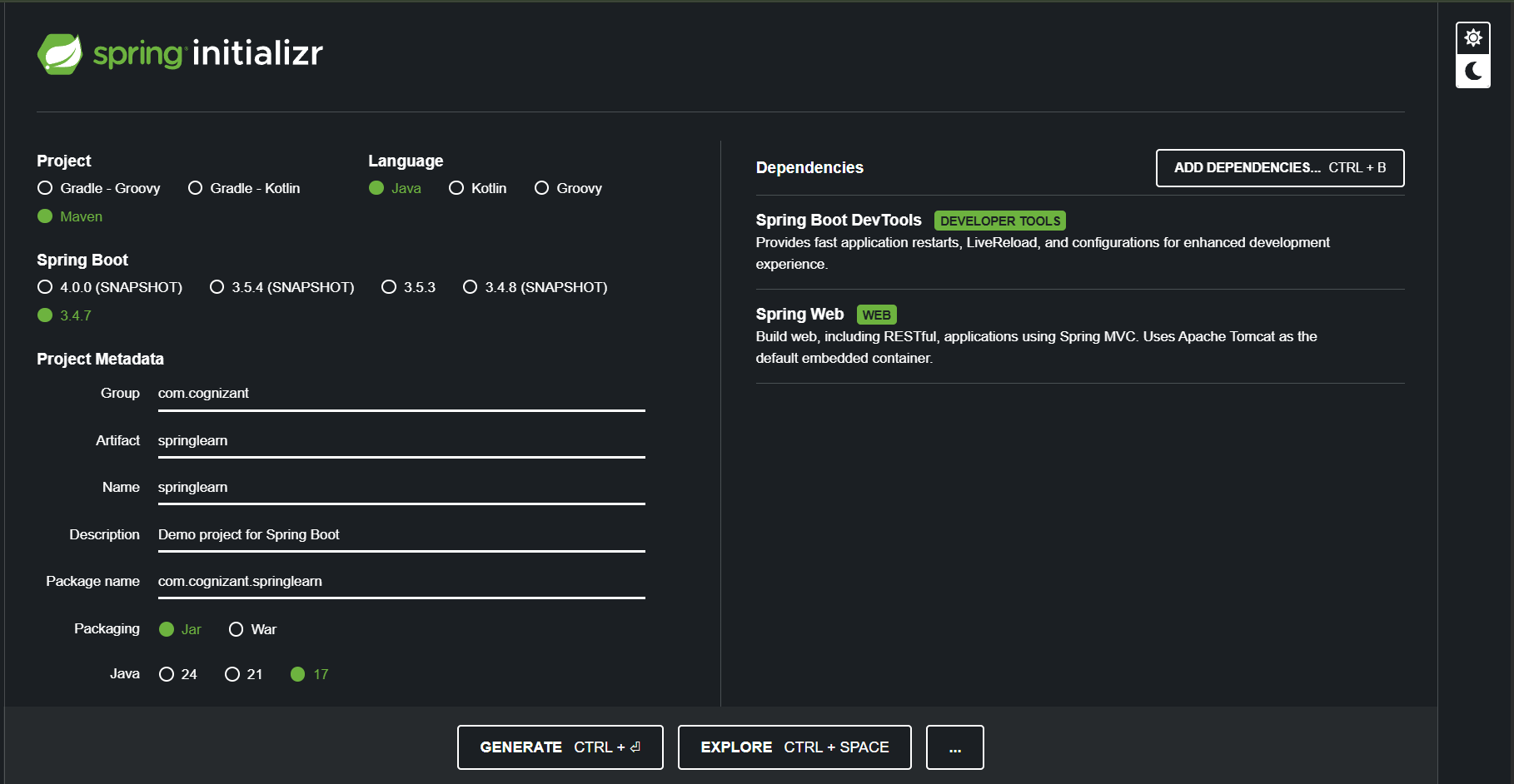


Showing Dependency Hierarchy of pom.xml

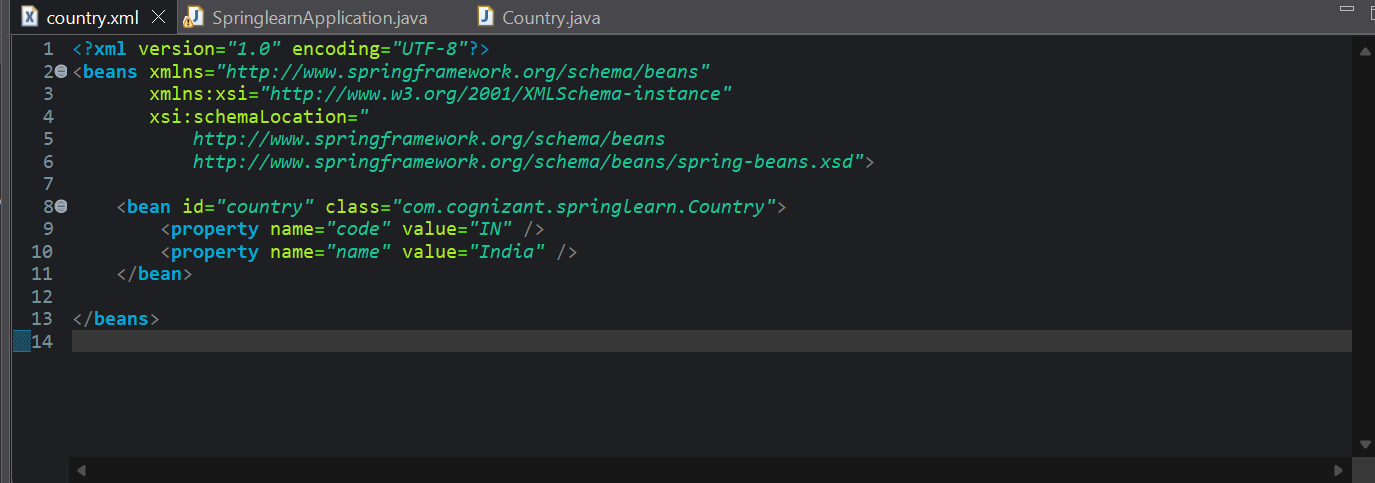


**Hands on 4**

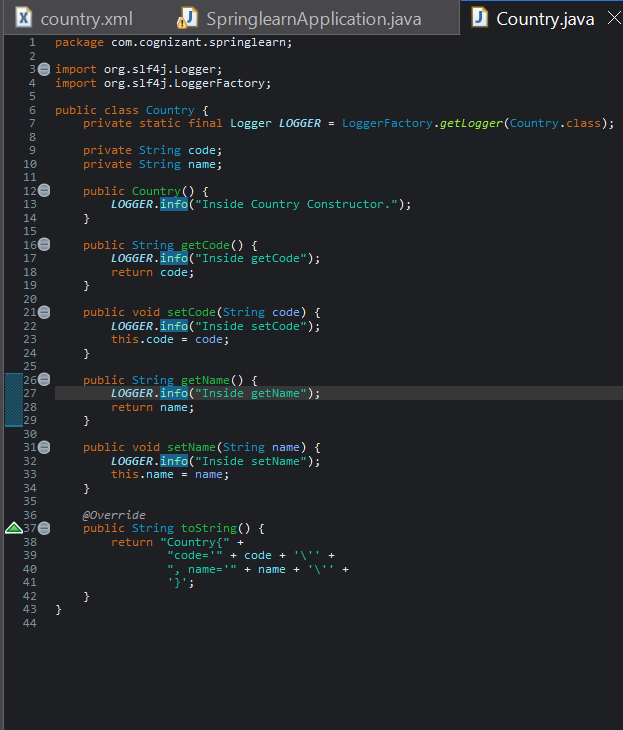
**Spring Core – Load Country from Spring Configuration XML**   
  
  
Created a new project using Spring Initializr



Configured country.xml file,



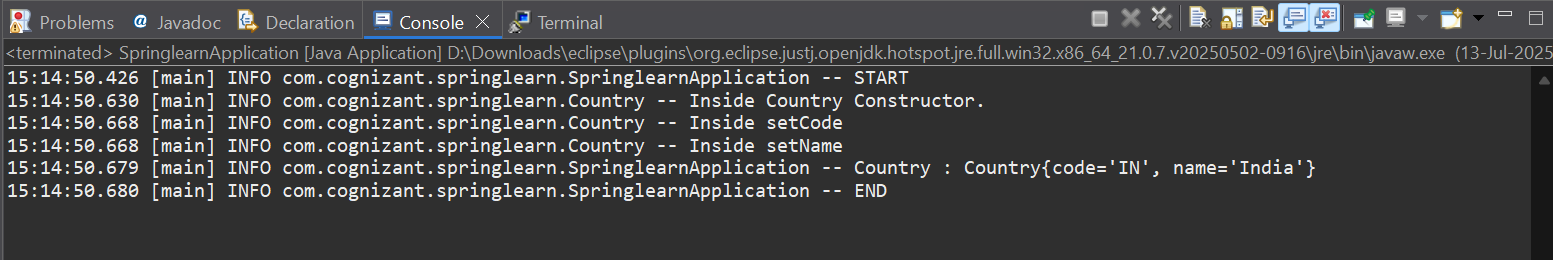
Configured Country Class, where the bean will be created



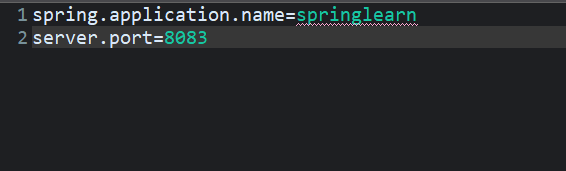
Configured main method to call display method from country



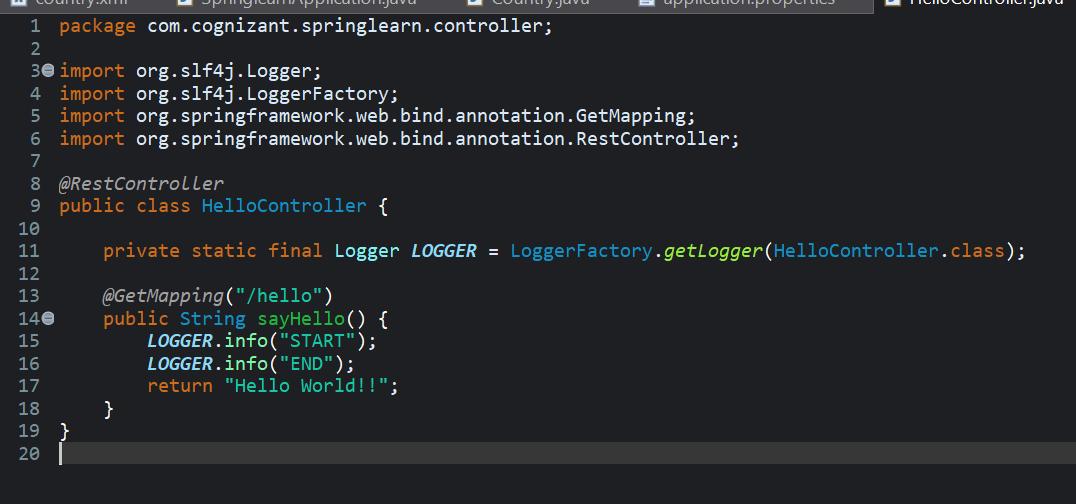
Output to show the flow of execution

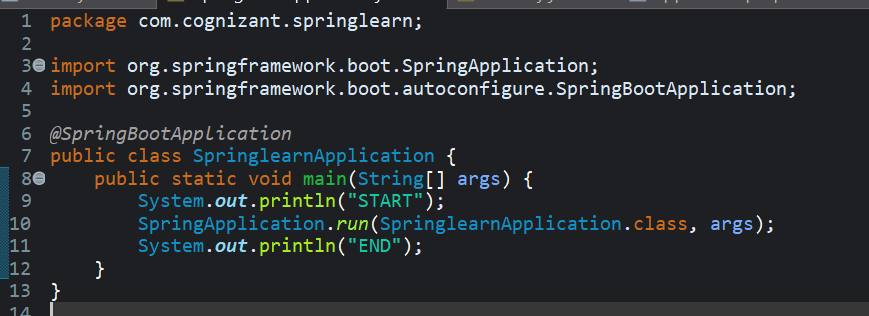


**Hello World RESTful Web Service**   
  
configured applicationcontext.xml file, to use port 8083

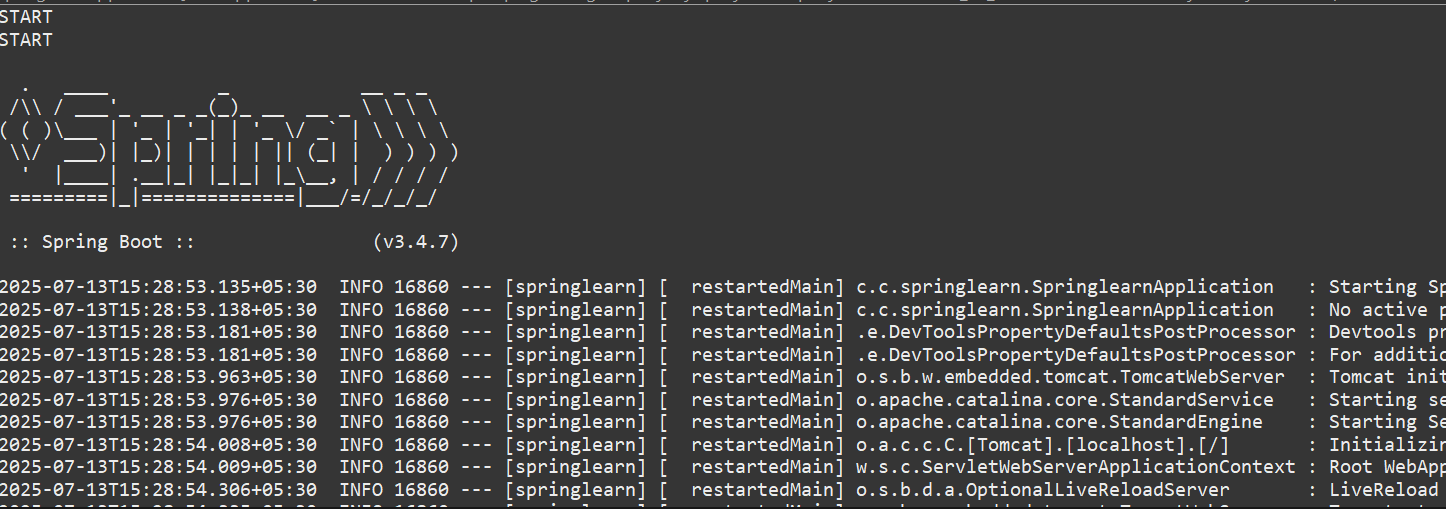


Defining Spring Controller to show hello world in the web page!

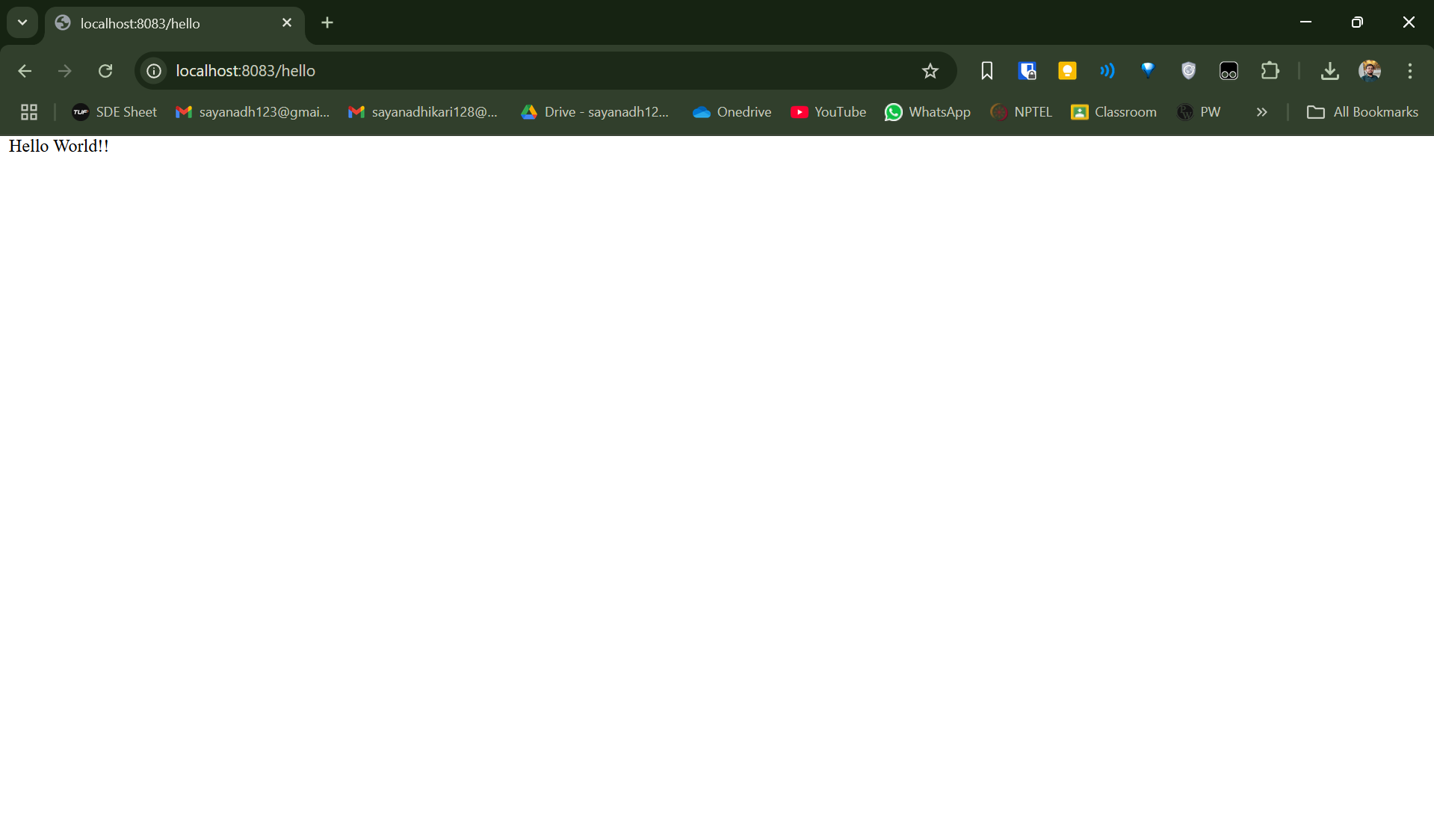


Main Method,

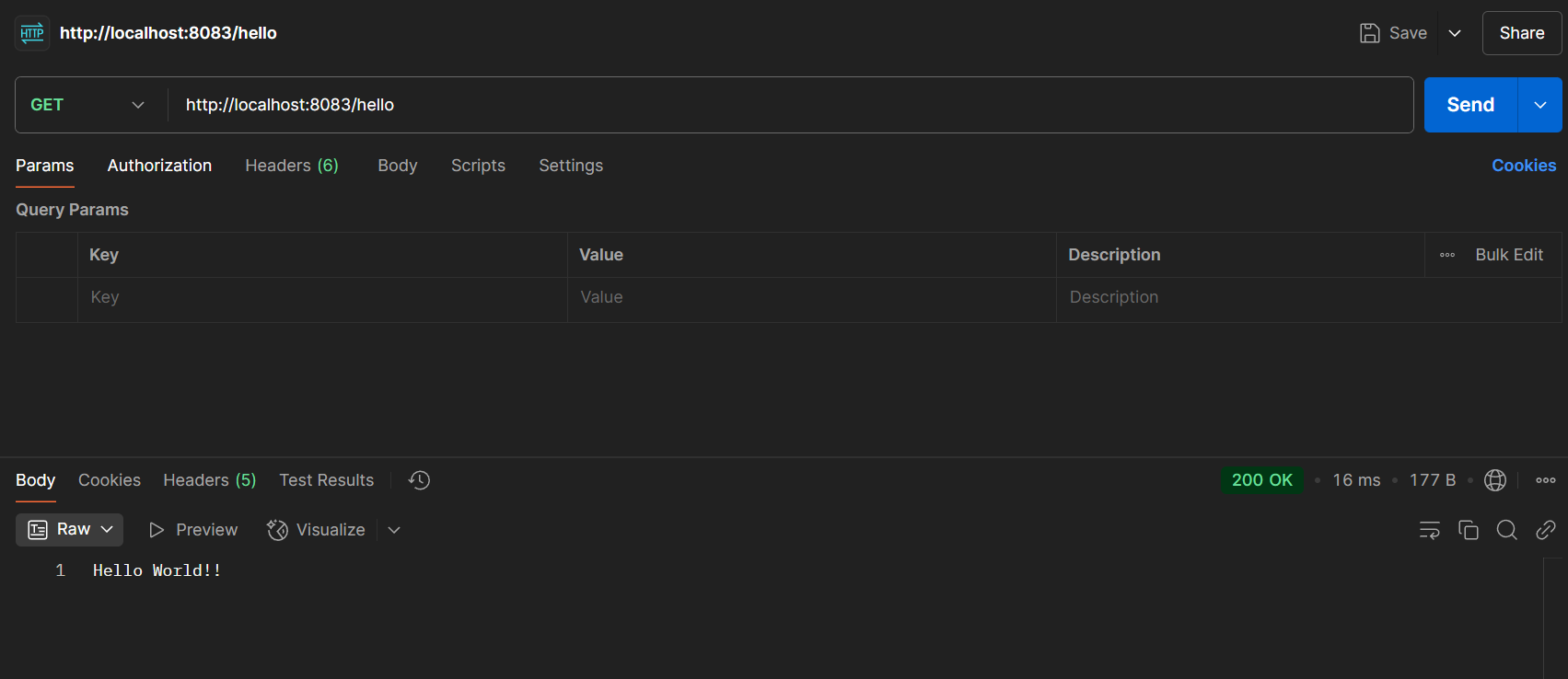
Console Output,



Web Browser Output,



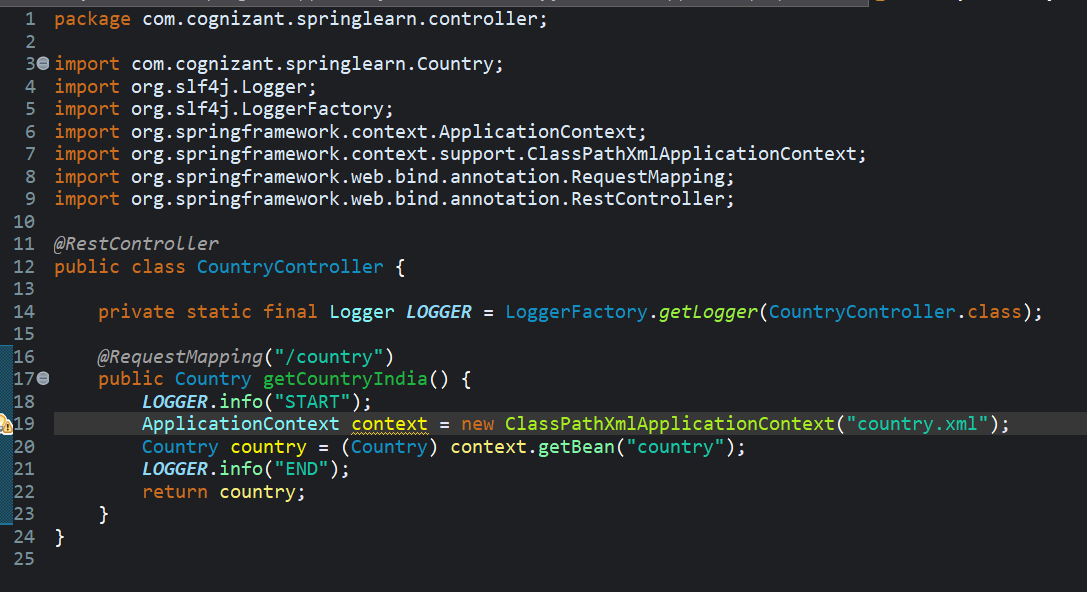
Postman Output,



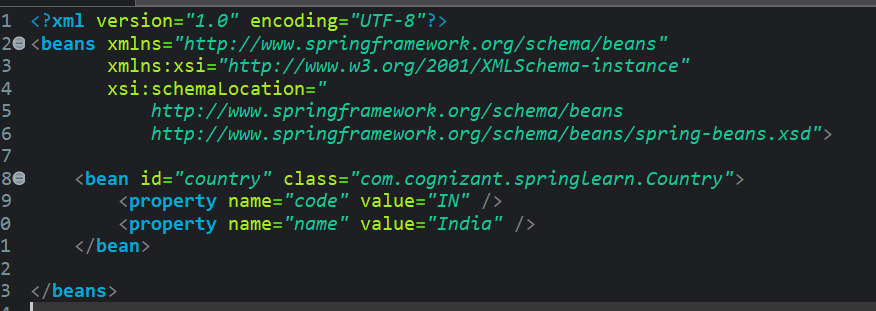
**REST - Country Web Service** 

Defining the controller for returning the given json response,

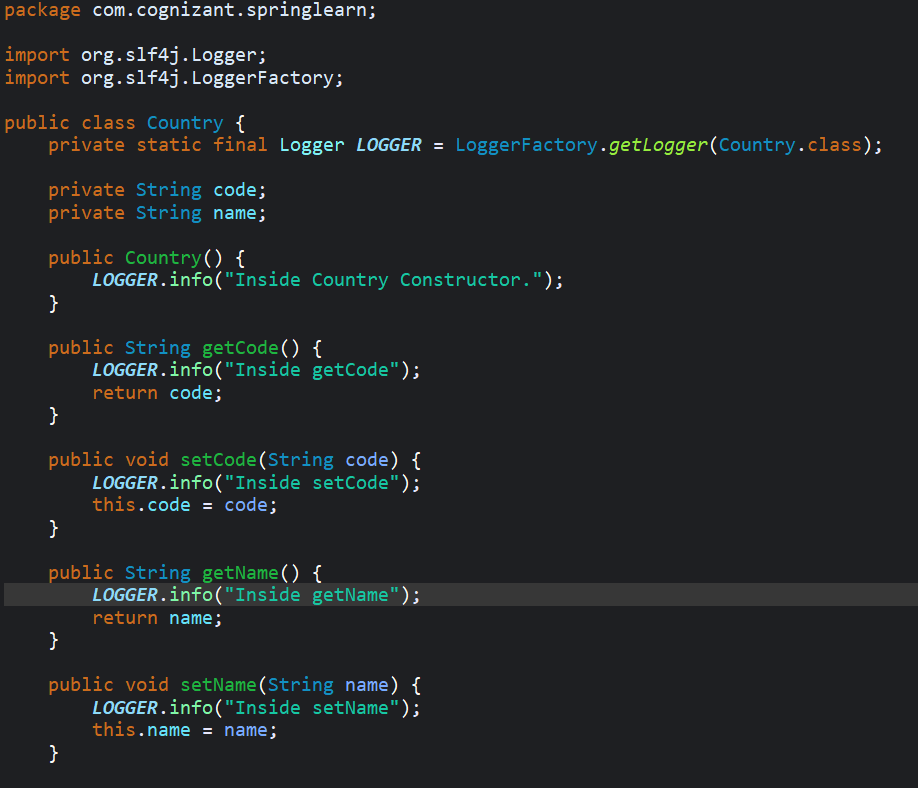
using @RequestMapping, and at URL - /country



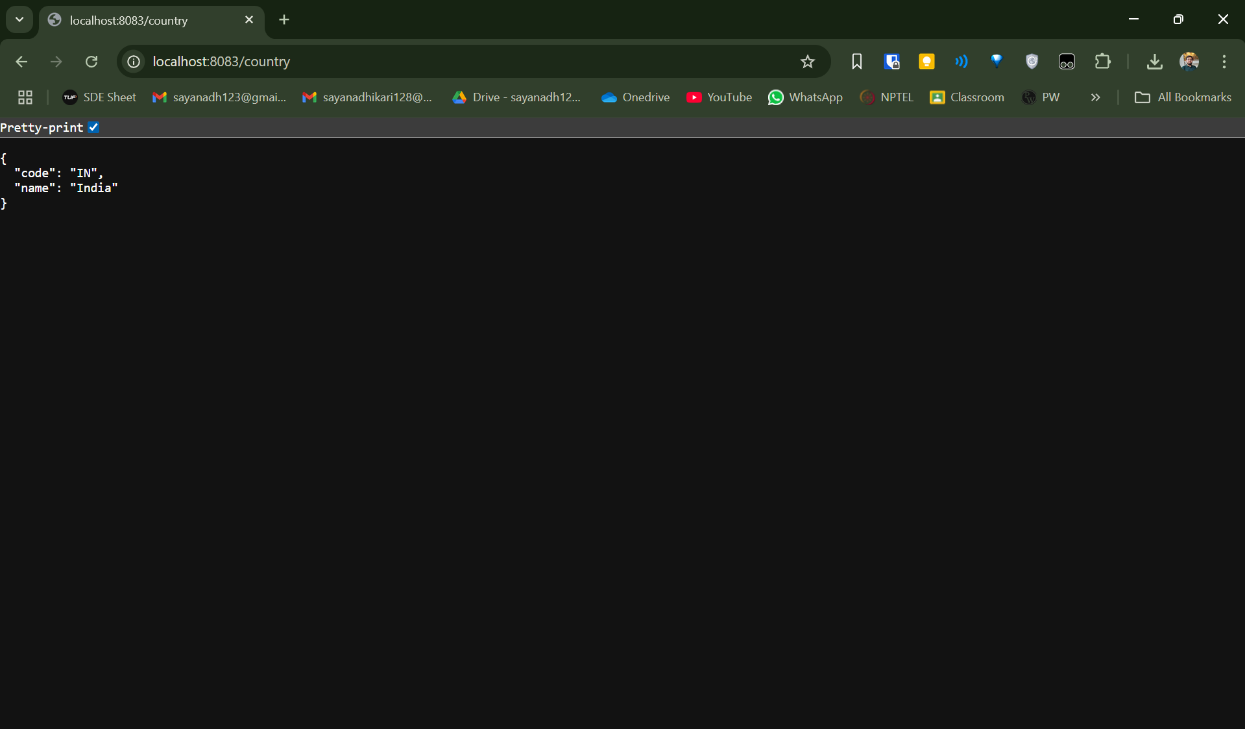
Defining the bean to load,



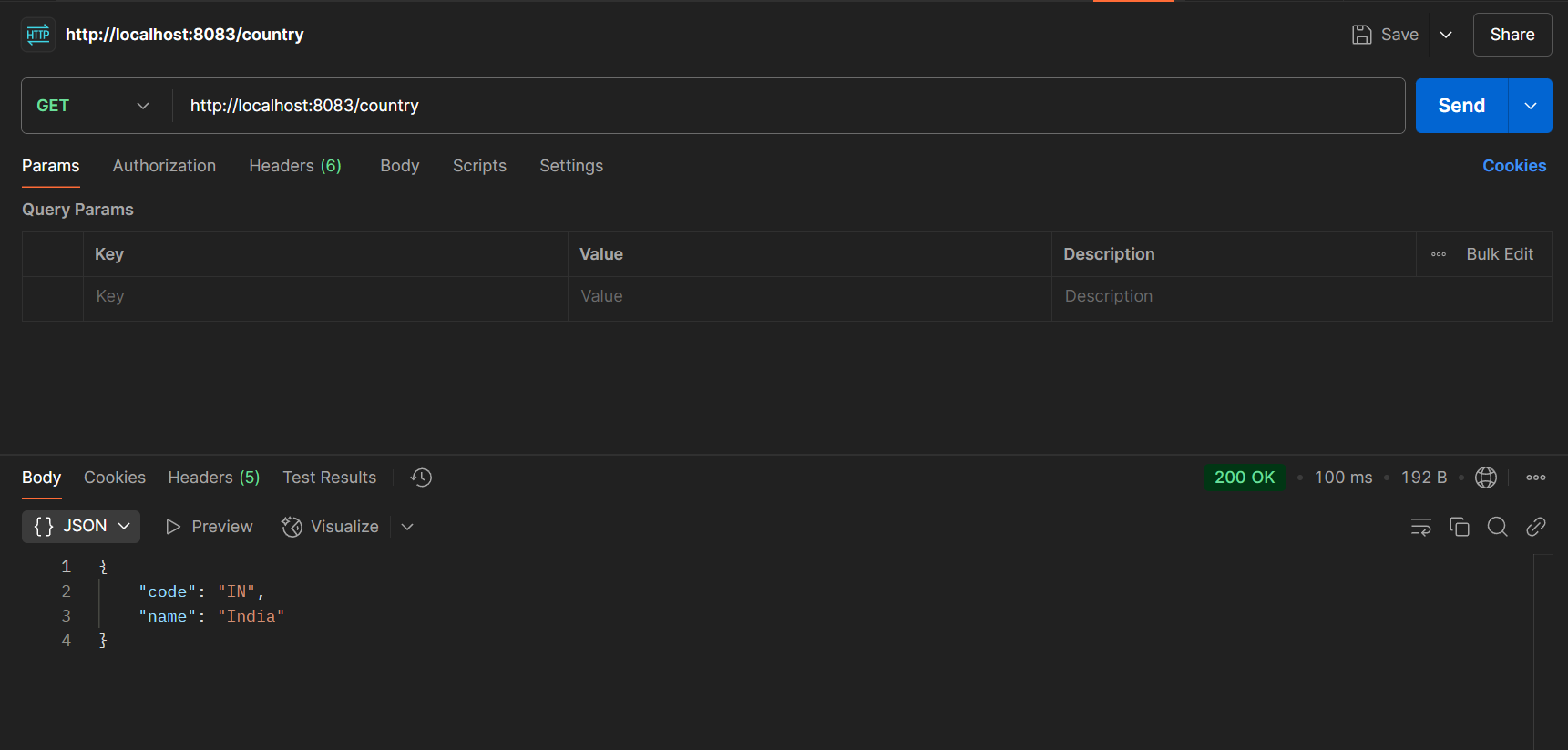
Defining the Country Class (Reusing from previous exercise)



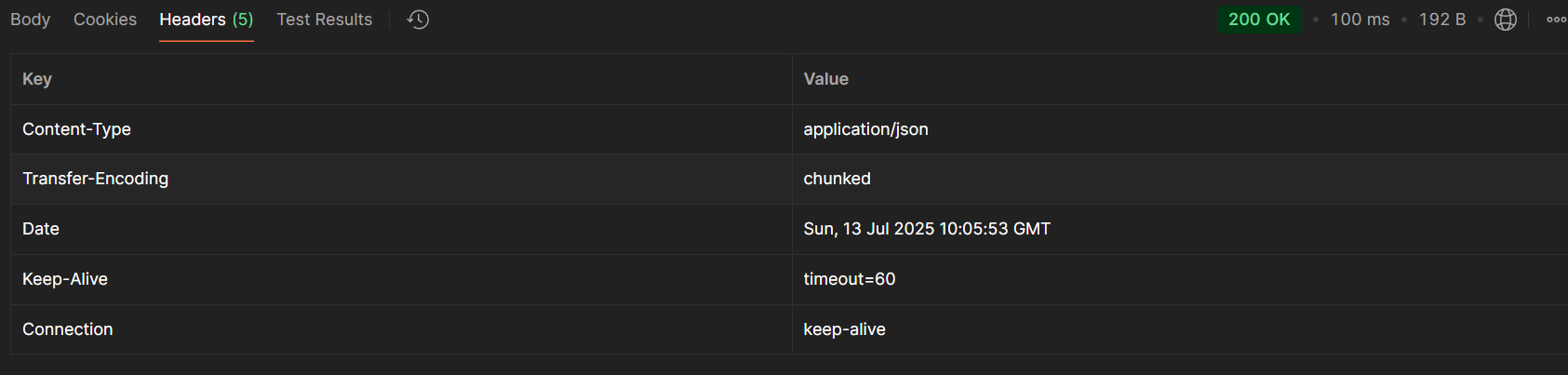
Browser Output



Postman Output

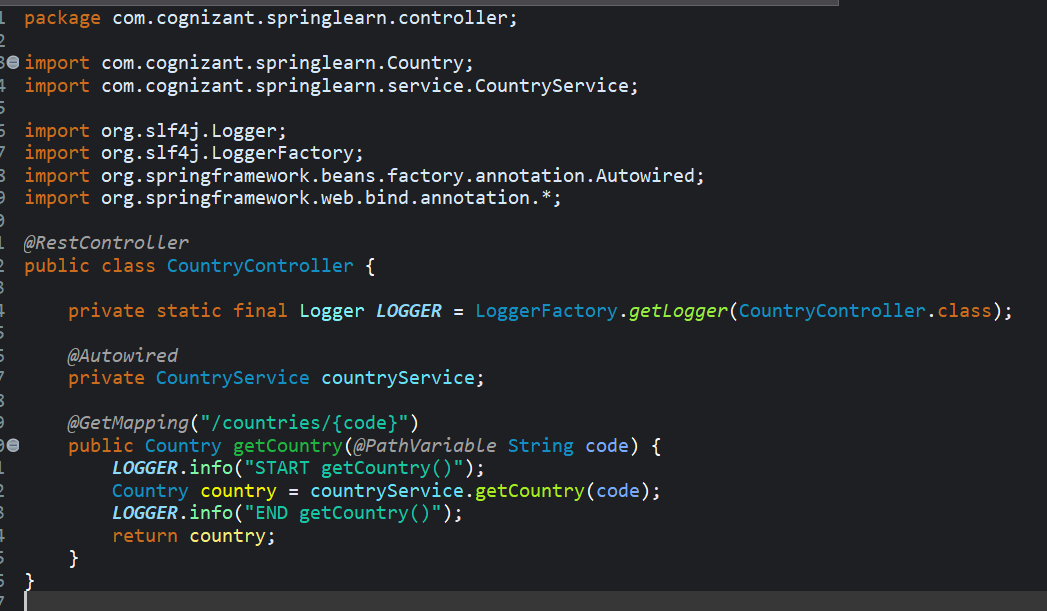


Headers tab in Postman



**REST - Get country based on country code**

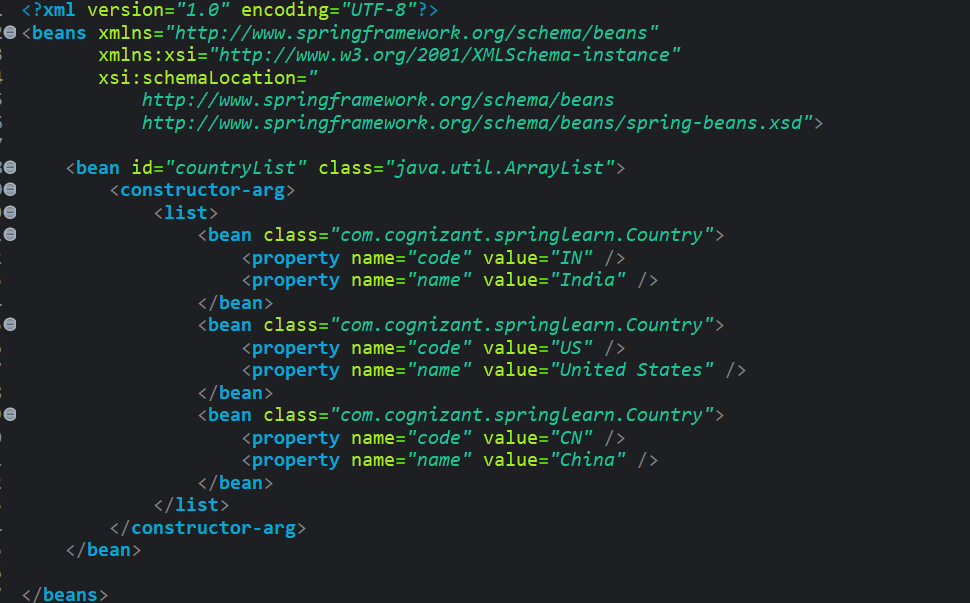
Defining the controller



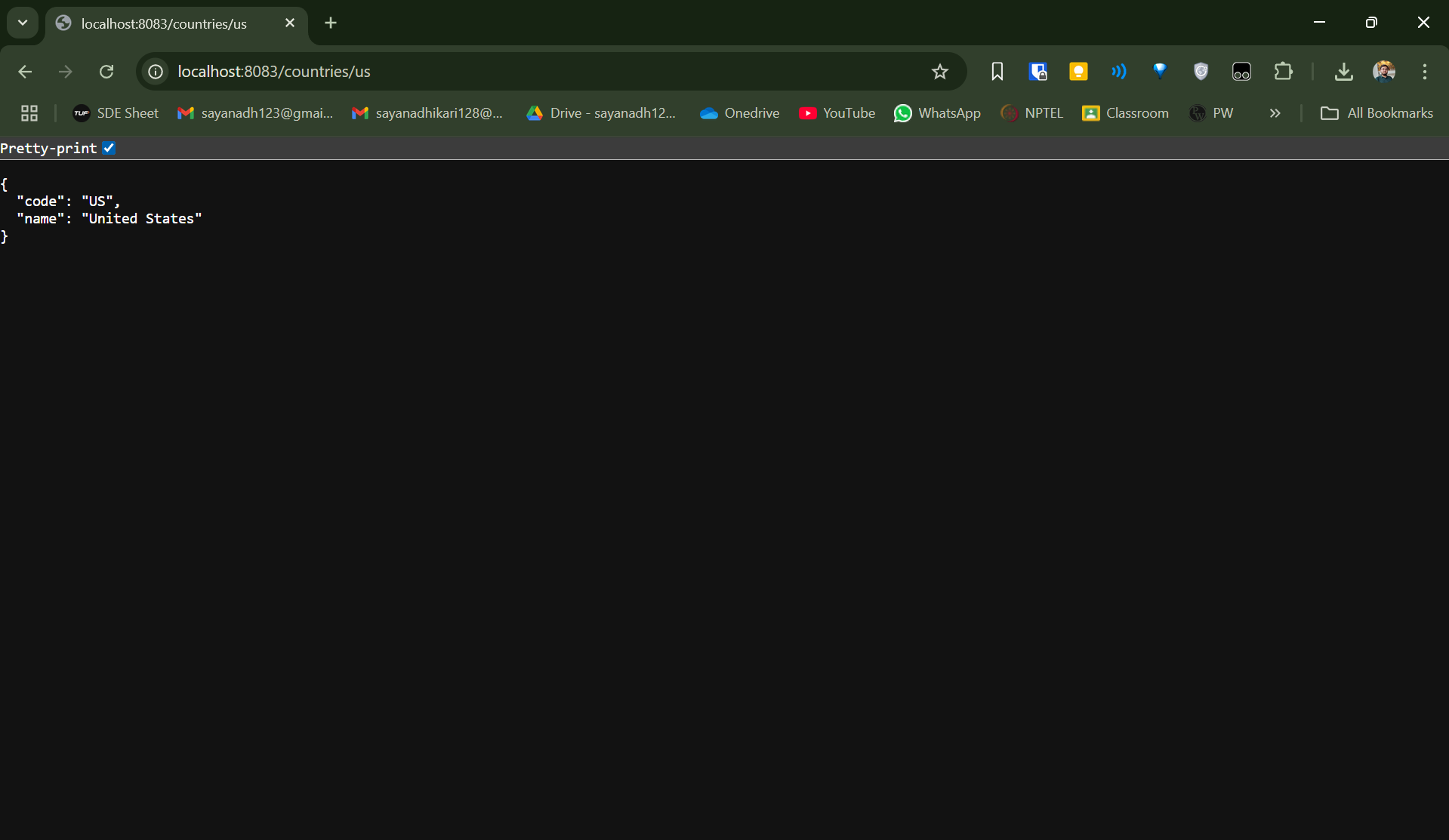
Country Service class, where the logic is defined



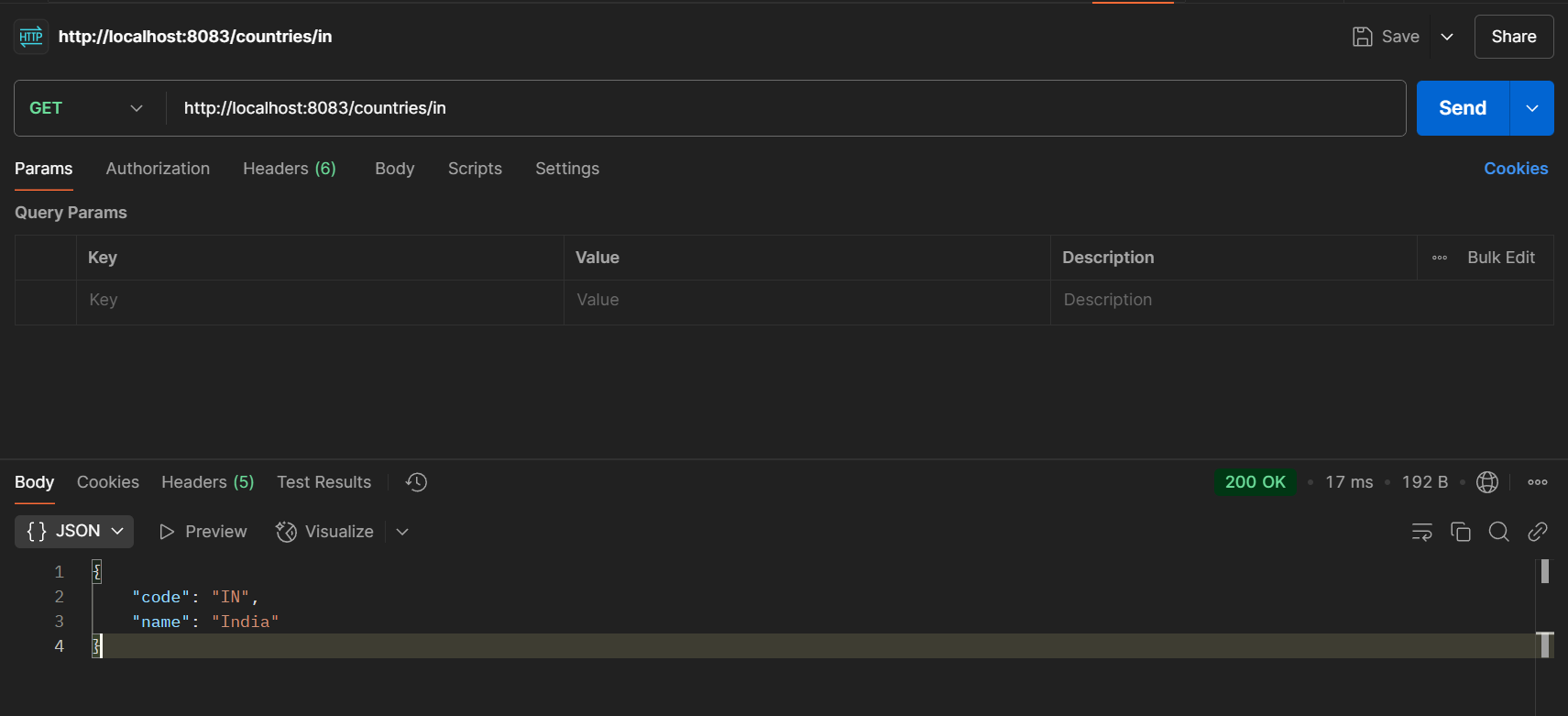
country.xml file



Browser Output,



Postman Output,



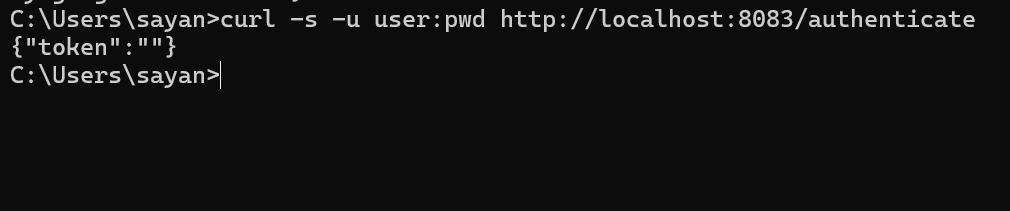
**Create authentication service that returns JWT**

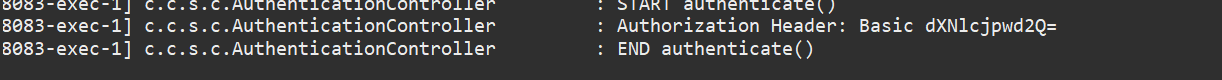
1. **Create authentication controller and configure it in SecurityConfig**

Created an auth controller to return a json response having token as a key and value as an empty string.



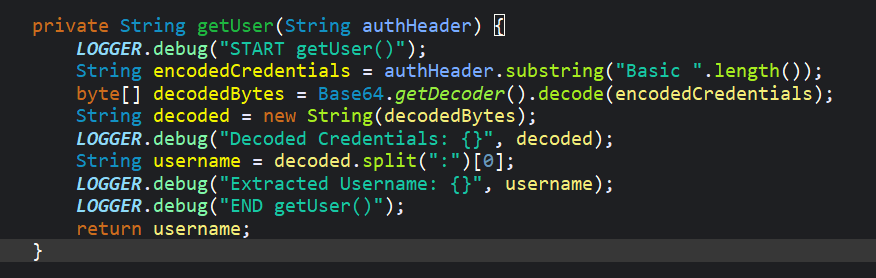
 Configuring SecurityConfig.java as mentioned,

 Testing our api as this point (Returns an empty string)

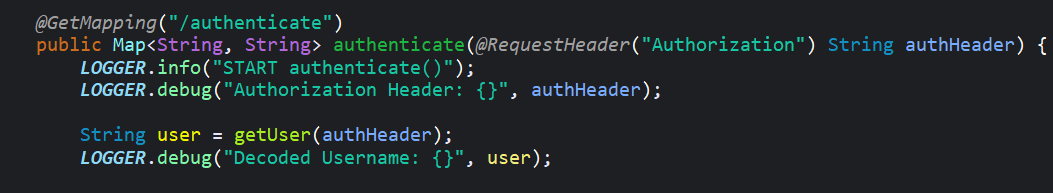
 Checking if Auth Header is logged correctly

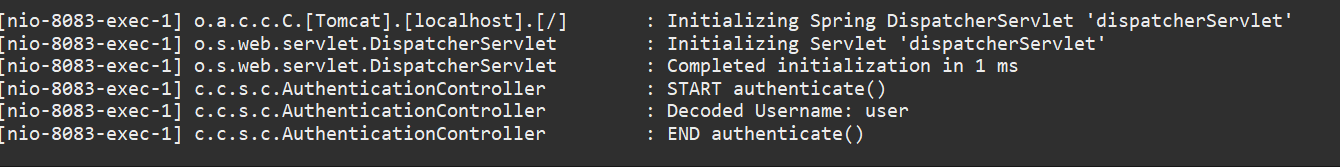
1. **Read Authorization header and decode the username and password**

Decoding the username from auth header, and returning it to be used by our previous method



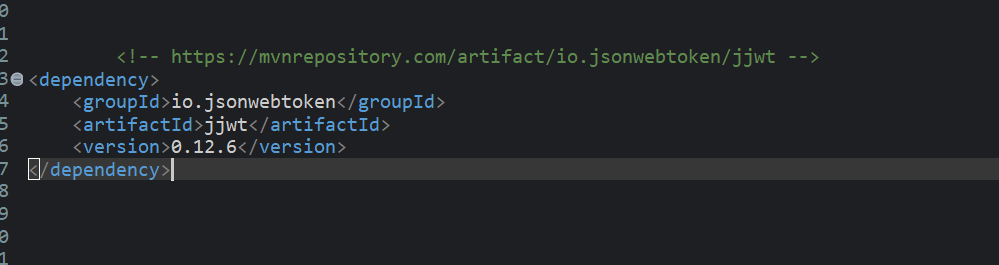
Making our authenticate method call getUser function and log the content



 Noticing it decodes correctly

1. **Generate token based on the user**

Include jwt libraray in pom.xml



 Creating generatejwt method,

Our final authenticate method which no longer sends empty string, rather the actual jwt token

Checking if jwt is printed,

