## **Hello World RESTful Web Service**

## **1️ Objective**

To create a simple RESTful service using the Spring Web Framework that returns the text "Hello World!!" for a GET request, and verify the HTTP headers in both Chrome browser and Postman.

## **3️ Implementation Steps**

### **Step 1: Add Spring Web Dependency**

Make sure your pom.xml includes:

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
</dependency>

### **Step 2: Set Server Port**

Add src/main/resources/application.properties:

server.port=8083

### 

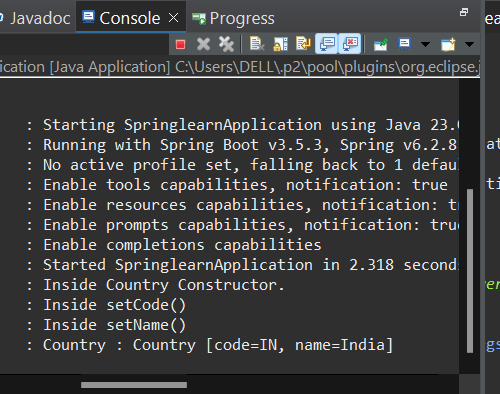
### **Step 3: Create HelloController**

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

package com.mycompany.myproject.controller;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);  
  
 @GetMapping("/hello")  
 public String sayHello() {  
 LOGGER.info("START sayHello()");  
 String response = "Hello World!!";  
 LOGGER.info("END sayHello()");  
 return response;  
 }  
}

**Explanation:**

* @RestController marks this as a RESTful controller.
* @GetMapping("/hello") maps GET requests to /hello.
* sayHello() returns hardcoded string “Hello World!!” and logs start and end.



### **Step 4: Run the Application**

Run SpringlearnApplication → Server starts at port 8083.

Access in:

* Chrome: <http://localhost:8083/hello>
* Postman: Send GET request to http://localhost:8083/hello

Sample Response:

Hello World!!

Logs:

START sayHello()  
END sayHello()

## 4️ **Verifying HTTP Headers**

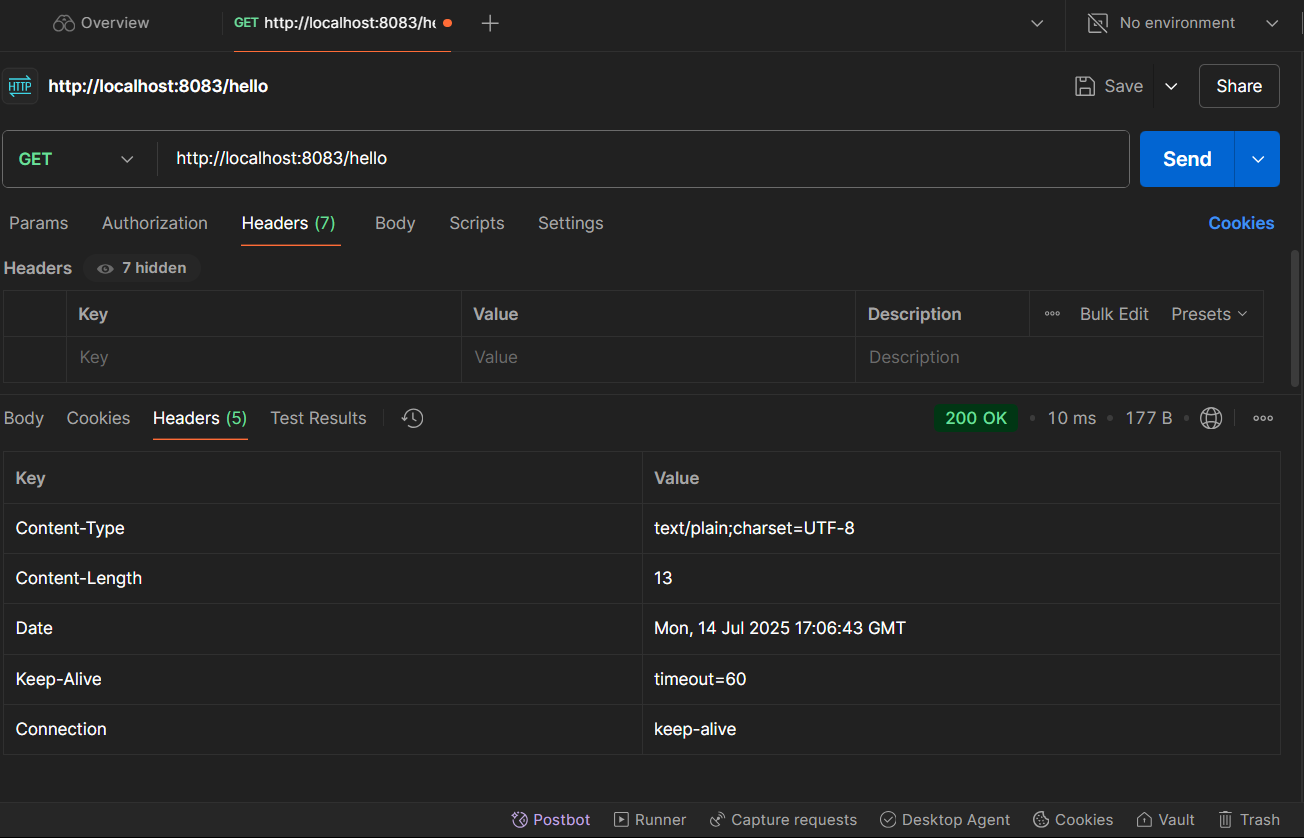
## Postman

1. Open Postman → New GET request → URL: http://localhost:8083/hello.
2. Click Send.
3. Click the Headers tab under response.

You’ll see:

* Content-Type: text/plain;charset=UTF-8
* Content-Length: 13
* Date: [Timestamp]

This confirms the server is returning correct headers.



## 