Hands-on: Hello World RESTful Web Service using Spring Boot

# 1. Objective

The objective of this hands-on exercise is to create a simple RESTful web service using Spring Boot that returns the text 'Hello World!!' on a GET request. The service is accessible at the endpoint /hello and logs the start and end of execution.

# 2. Implementation Steps

• Create a controller class named HelloController under package com.cognizant.springlearn.controller.

• Annotate the class with @RestController to expose it as a RESTful controller.

• Use @GetMapping("/hello") to map the GET request to the method.

• Implement sayHello() method that logs start and end, and returns the string 'Hello World!!'.

# 3. HelloController.java

package com.cognizant.springlearn.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 message = "Hello World!!";  
 LOGGER.info("END - sayHello()");  
 return message;  
 }  
}

# 4. Optional Configuration - Port Change

To change the default port, add the following to application.properties:

server.port=8083

# 5. Execution and Testing

• Run the SpringLearnApplication.java class.

• Open browser and go to http://localhost:8083/hello

• Response: Hello World!!

• In Postman, perform a GET request to the same URL.

• Check the 'Headers' tab to see HTTP response headers.

# 6. HTTP Headers via Developer Tools

• Open Chrome Dev Tools → Network tab → Hit the /hello endpoint.

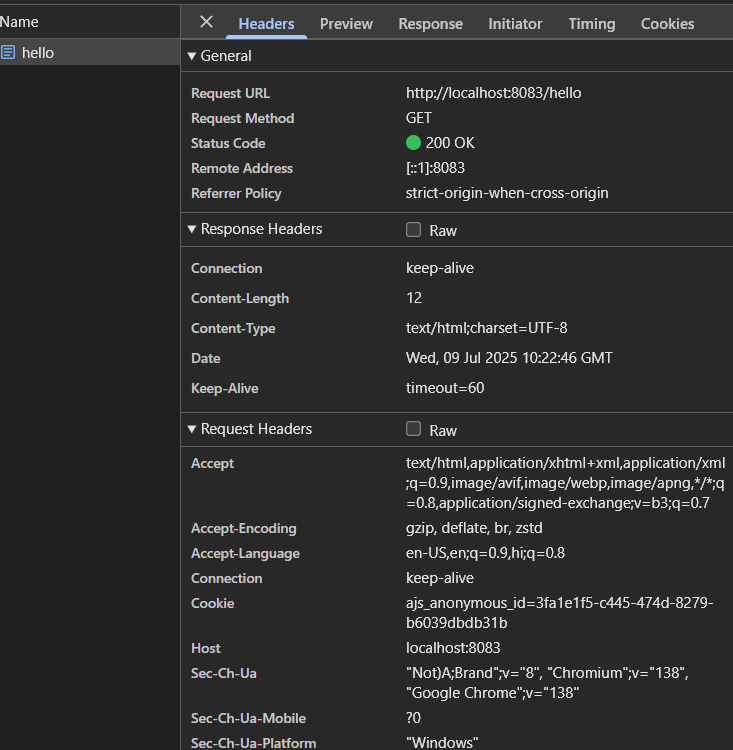
• Click the request to view:

- Request Headers: Method, Host, User-Agent, Accept

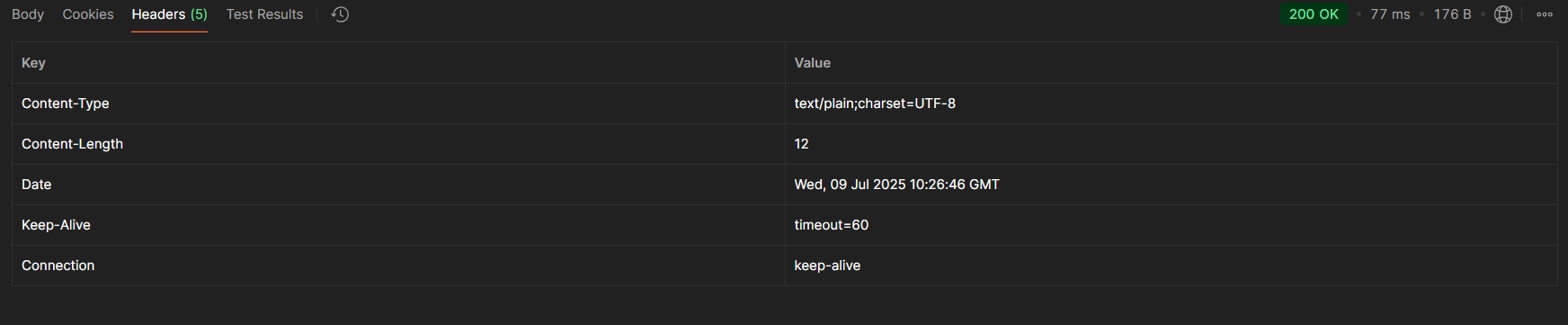
- Response Headers: Content-Type, Server, Date

**Output :**

Via Chrome browser -

****

Via Postman -

****

# 7. Conclusion

A basic REST endpoint was successfully implemented in Spring Boot to return a static message. Testing in browser and Postman confirms correct behavior. HTTP headers were also inspected using developer tools and Postman to understand the request/response cycle.