

Hello World RESTful Web Service

Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework. Refer details below:

Method: GET

URL: /hello

Controller: com.cognizant.spring-learn.controller.HelloController

Method Signature: public String sayHello()

Method Implementation: return hard coded string "Hello World!!"

Sample Request: http://localhost:8083/hello

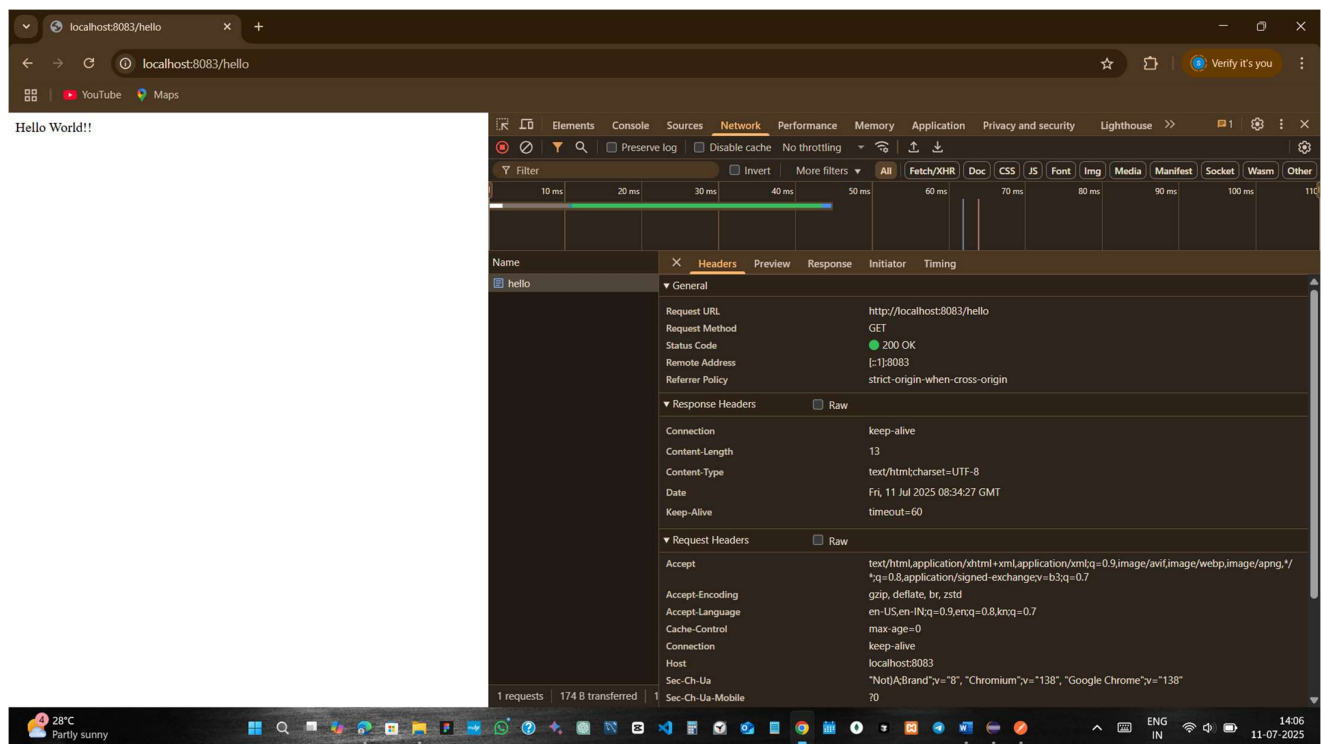
Sample Response: Hello World!!

IMPORTANT NOTE: Don't forget to include start and end log in the sayHello() method.

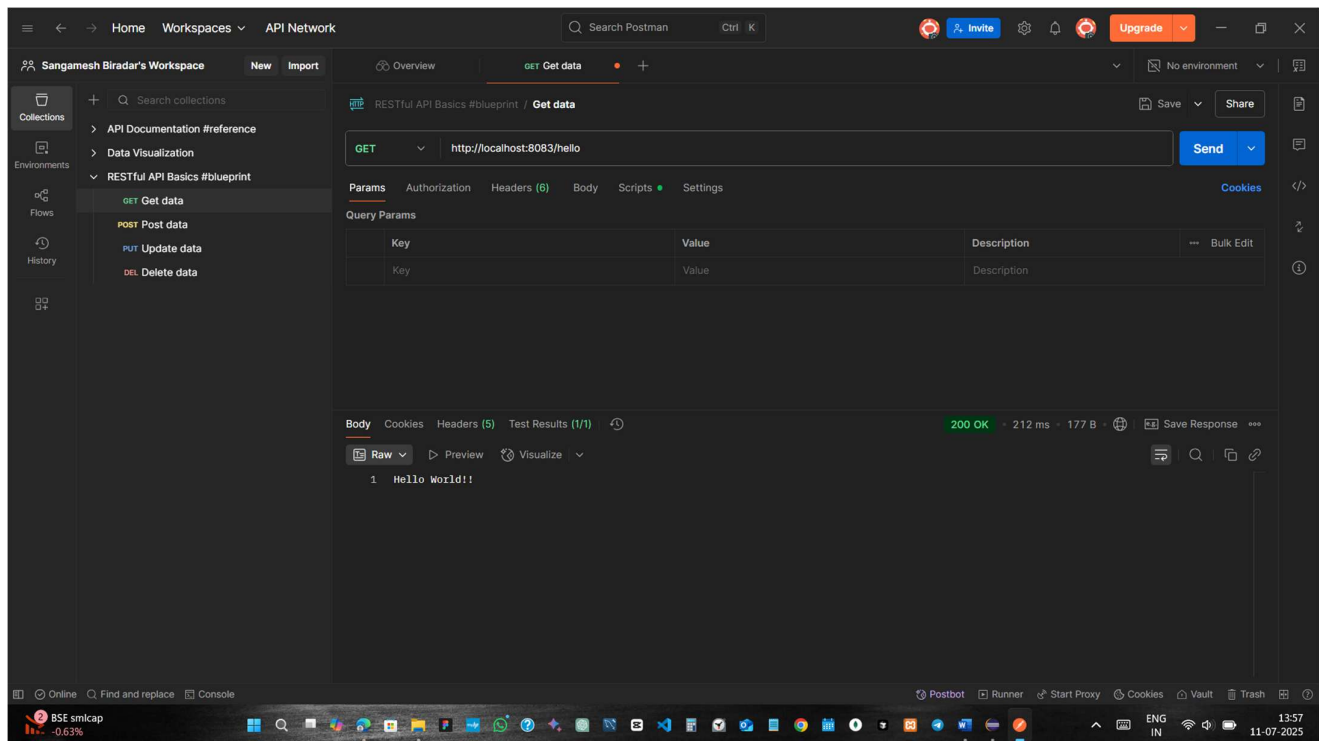
Try the URL <http://localhost:8083/hello> in both chrome browser and postman.

SME to explain the following aspects:

- In network tab of developer tools show the HTTP header details received



- In postman click on "Headers" tab to view the HTTP header details received



HelloController.java

```
package com.cognizant.spring_learn.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;
}
}

```

SpringLearnApplication.java

```

package com.cognizant.spring_learn.controller;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

    public static void main(String[] args) {

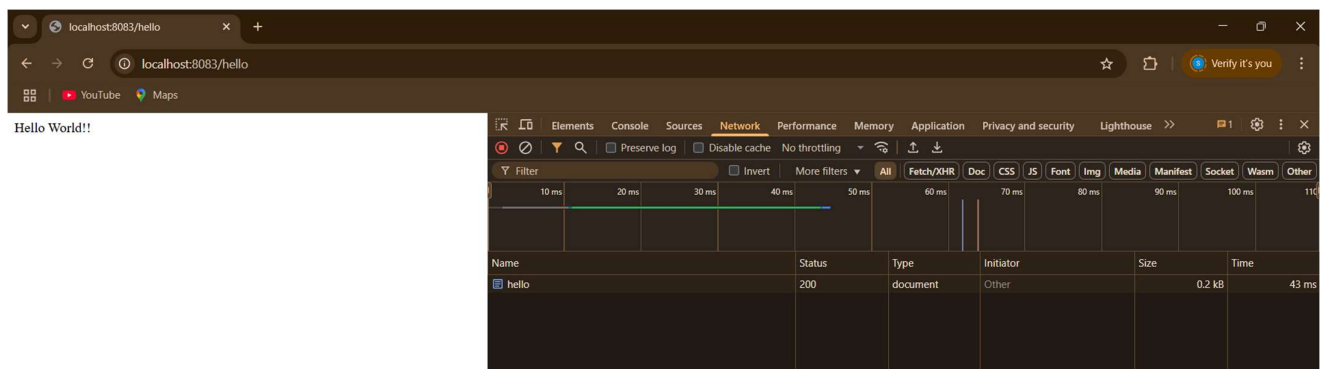
        SpringApplication.run(SpringLearnApplication.class, args);

    }

}

```

Result: [Browser]



Postman:

The screenshot displays the Postman application interface. The top bar shows the workspace name "Sangamesh Biradar's Workspace" and the current collection "RESTful API Basics #blueprint". The main panel is set to "GET Get data" with the URL "http://localhost:8083/hello". The "Send" button is visible. Below the URL bar, the "Params" tab is active, showing a table with headers "Key", "Value", and "Description". The "Body" tab is also visible, showing a "200 OK" status, a response time of "212 ms", and a body size of "177 B". The response body is displayed in "Raw" format as "1 Hello World!!". The bottom status bar shows the system tray with various icons and the time "13:57" on "11-07-2025".

RESTful API Basics #blueprint / Get data

GET http://localhost:8083/hello

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results (1/1)

200 OK · 212 ms · 177 B

Raw Preview Visualize

1 Hello World!!

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

BSE smicap -0.63%

13:57 11-07-2025