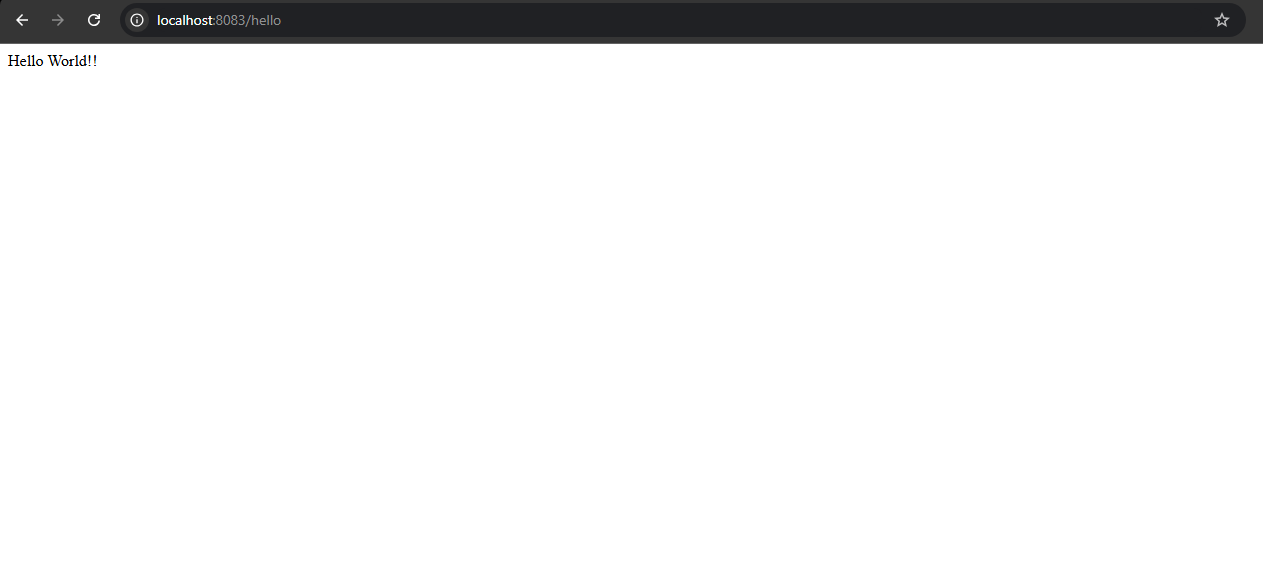
Screenshots



# 🌐 Hello World RESTful Web Service (Spring Learn)

## 📄 Description

This simple Spring-based RESTful web service demonstrates how to create a basic HTTP `GET` endpoint using the \*\*Spring Web\*\* module. The service is part of the `spring-learn` application and is designed to return the plain text message `"Hello World!!"` when accessed.

The endpoint acts as an introductory example of how Spring handles RESTful requests and responses using annotations such as `@RestController` and `@GetMapping`.

### ✅ Endpoint Details

| Method | URL | Response |

|--------|--------------------------|------------------|

| GET | `http://localhost:8083/hello` | `Hello World!!` |

### 🧩 Controller Implementation

- \*\*Package\*\*: `com.cognizant.spring\_learn.controller`

- \*\*Class\*\*: `HelloController`

- \*\*Method Signature\*\*: `public String sayHello()`

- \*\*Method Logic\*\*: Returns a hard-coded string `"Hello World!!"`

The method includes debug-level logging for both \*\*start\*\* and \*\*end\*\* of execution using a logger.

### 🎯 Features Covered

- Creating a REST controller using `@RestController`

- Mapping HTTP GET requests using `@GetMapping`

- Logging method entry and exit using SLF4J `Logger`

- Testing the endpoint via \*\*browser\*\* and \*\*Postman\*\*

### 🔍 HTTP Header Observations

- When tested in \*\*browser\*\*, open \*\*Developer Tools → Network tab → hello request\*\* to inspect HTTP headers (e.g., status, content-type, content-length).

- When tested in \*\*Postman\*\*, switch to the \*\*Headers\*\* tab in the response section to view the full list of response headers sent by the Spring server (e.g., server, date, connection, etc.).