**2.1 Hello World RESTful Web Service**

**Objective:**

Create a Restful web service in the spring-learn application that responds with the text "Hello World!!" when accessed through a browser or API testing tool i.e., postman. Add logging to the controller method to track execution flow.

Steps to create and Test Hello World REST Service

**Step1**: Create a Maven project with configurations consisting as below:

* GroupId
* ArtifactId
* Name
* Package etc.…

**Step2**: Create a Controller Class

The folder structure of created project is as follows:

1. **src/main/java** – Contains application code: controllers, services, repositories.
2. **src/main/resources** – Configuration files like application.properties.
3. **src/test/java** – Contains unit and integration tests.

**Step3**: Configure the created pom.xml file with following dependencies.

|  |
| --- |
| <groupId>com.cognizant</groupId>  <artifactId>spring-learn</artifactId>  <version>0.0.1-SNAPSHOT</version>  <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-devtools</artifactId>  <scope>runtime</scope>  </dependency>  </dependencies> |

**Step4**: Create a Controller class.

In STS, create a new class i.e., HelloController.java

Following code is added to the controller class to define a REST endpoint.

|  |
| --- |
| 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 response = "Hello World!!";  LOGGER.info("END - sayHello()");  return response;  }  } |

Here,

@RestController purpose is, it marks the class as REST controller that returns the data directly.

@GetMapping – Maps HTTP Get request to /hello URL to sayHello() method.

Logger – Adds logging to show start and end of method execution in console.

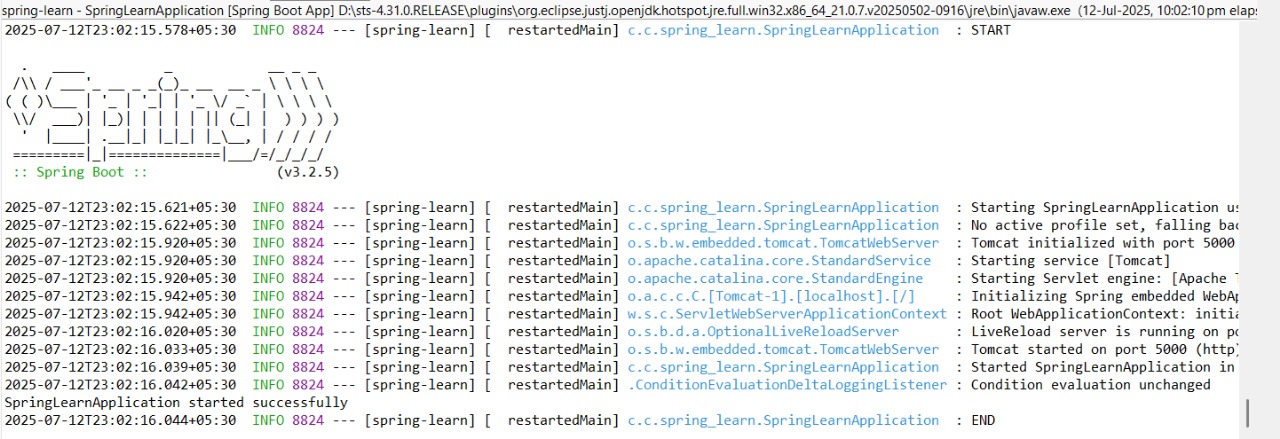
**Step5:** Create SpringLearnApplication.java

|  |
| --- |
| package com.cognizant.spring\_learn;  import org.slf4j.Logger;  import org.slf4j.LoggerFactory;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  public class SpringLearnApplication {  private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  public static void main(String[] args) {  LOGGER.info("START");  SpringApplication.run(SpringLearnApplication.class, args);    LOGGER.info("END");  }  } |

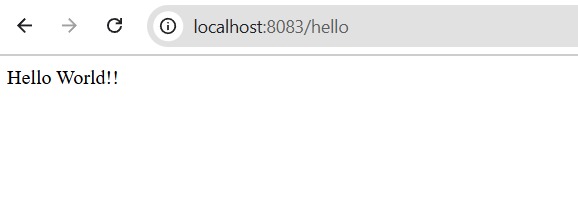
**Step6:** Run the project i.e.,

The SpringLearnAppication.java is running as spring boot app.

**Expected outcome:**



When we enter the server port in the web browser the outcome is,



***Now,***

**Step7**: Test the REST Service using Postman

1. Open Postman.
2. Create a new GET request:

* URL: <http://localhost:8083/hello>

1. Click **send**.
2. The outcome will be appeared in the Body

i.e.,

