Documentation for the Hello World Program in Spring Boot

- **Prerequisites :**

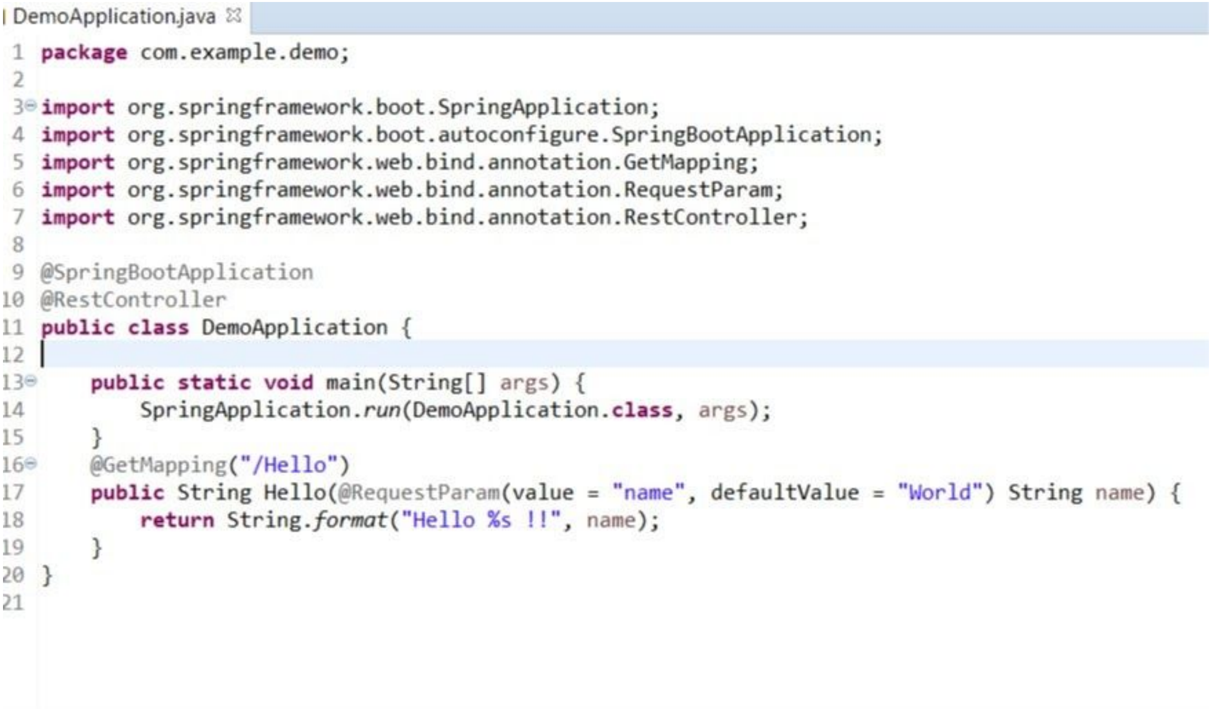
  - Spring Tool Suite
  - JDK

- **Steps for creating a web project :**

  1. Go to the **start.spring.io** website.
  2. In the project field choose **Maven**, java version (that you have on your computer) , select **Web** dependency, spring boot version, set the name for your project and generate the zip file.
  3. Extract the zip file on a folder on your computer.
  4. It contains several folders like java, resources, test java, maven dependencies, src, pom, etc.
  5. Go to the DemoApplication.java (or the main file of your package).
  6. It has some basic code. Now, it's time to edit the code as per your use.

- **Description of the project :**

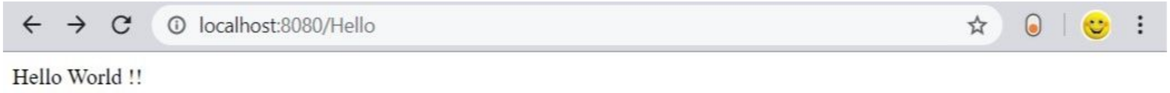
  - I have used Maven instead of Gradle as the build system.
  - In the DemoApplication.java, I have used the most common features of spring boot.
  - For Example : auto configuration, GetMapping, RequestParam, RestController.
  - I have created a separate function called **Hello** that takes an input parameter name and returns a string that is to be printed on the localhost. The function combines the string "Hello" with the request and prints eg: "Hello John" if name = john is sent as request. Else if no request is there, it is by default "Hello World".
  - The **@RestController** tells spring that this code describes an endpoint that should be made available over the web.
  - The **@GetMapping("/Hello")** tells the spring to use our Hello method to answer requests that get sent to the https://localhost:8080/Hello address.
  - The **@RequestParam** is telling spring to expect a name value in the request else it will have some default value.



```java
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
    @GetMapping("/Hello")
    public String Hello(@RequestParam(value = "name", defaultValue = "World") String name) {
        return String.format("Hello %s !!", name);
    }
}

```

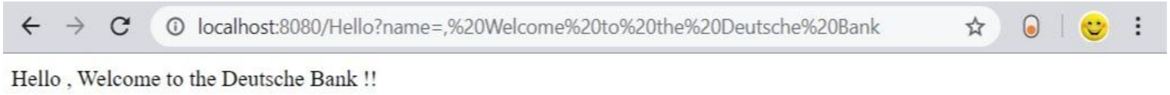
- **It's time to RUN :**
  - To run the program run the command in the command prompt by navigating to the folder where we have our project files : **mvnw spring-boot:run**
  - When I ran it for the first time it was showing some error regarding the localhost:8080 to be busy. To tackle the given problem, I run the command **netstat -ao |find /i "listening"** to get the details for the running processes. It shows that the port was to the process with Process ID 12368. So to kill that process I executed the command **Taskkill /F /IM 12368.**
  - After running the code again, it successfully gets executed on port 8080.

- When we open https://localhost:8080/Hello the output is as shown in the image:



```
Hello World !!
```

- To make a request using the parameter we just need to add: **?name=(anything you want to display)** after the URL.
- For example, **https://localhost:8080/Hello?name=,Welcome to Deutsche Bank**, the output is like :



localhost:8080/Hello?name=,%20Welcome%20to%20the%20Deutsche%20Bank

Hello , Welcome to the Deutsche Bank !!

- References are taken from the official website of spring.

THANK YOU

By:
Deshna Wankhede
deshna.wankhede@db.com