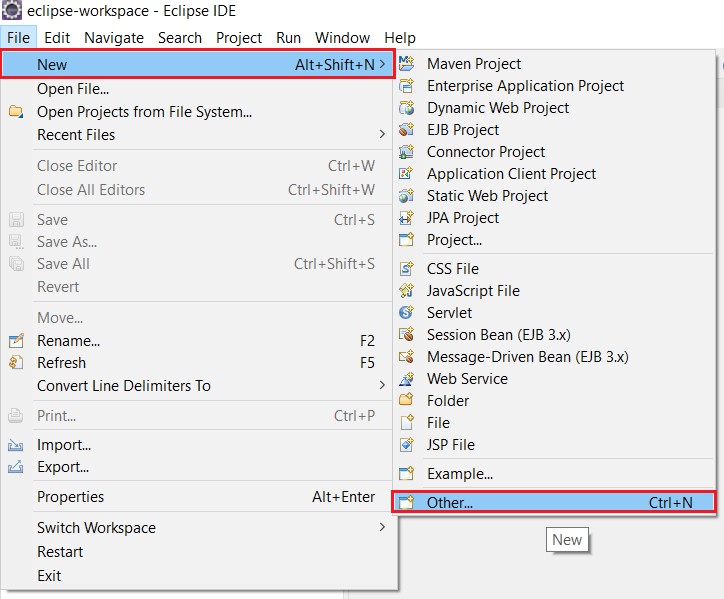
Assignment No 10

and RESTful Web Services

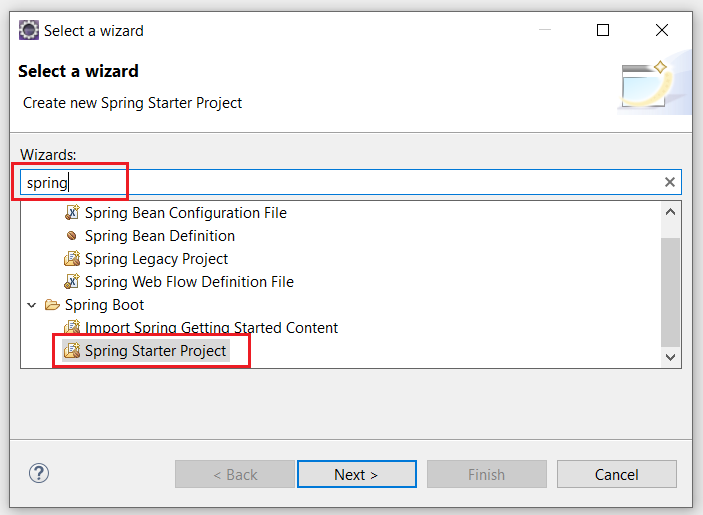
1. Write a program to create a simple Spring Boot application that prints a message.
2. Write a program to demonstrate RESTful Web Services with spring boot

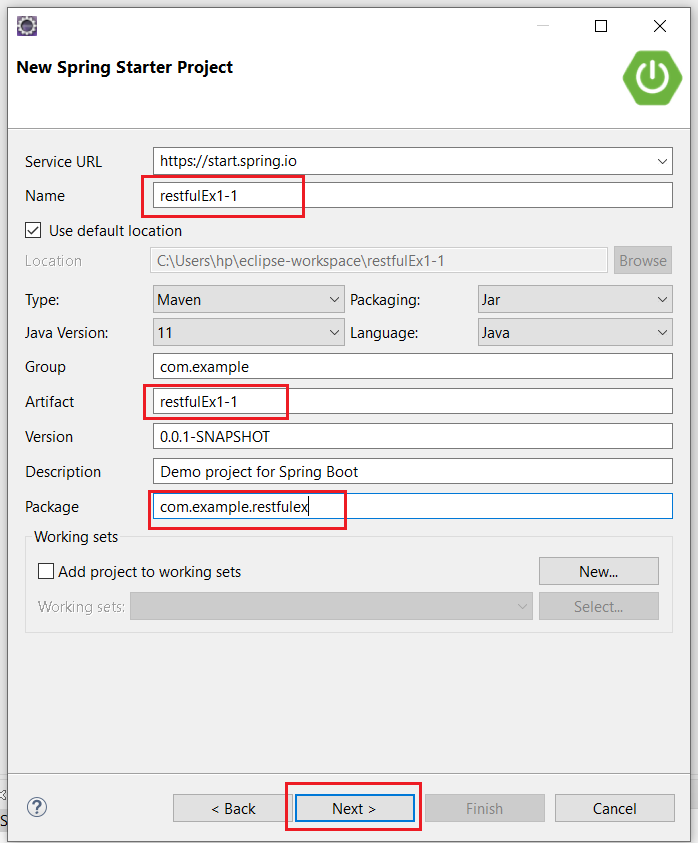
**Problem Statement 10.1. Write a program to create a simple Spring Boot application that prints a message.**

**Note : Make sure you have installed the Spring Plugin in Eclipse Itself. Step 1 :**

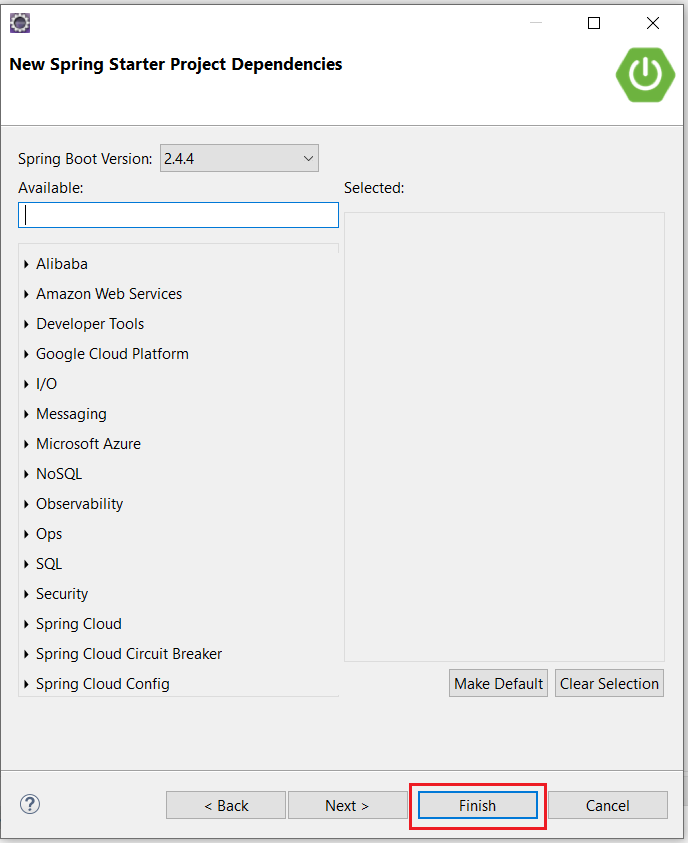
* 1. **:** Open Eclipse. Go To File > New > Other.
  2. **:** Search for ‘Spring Boot ’ and Select ‘Spring Starter Project’. Then Click on Next.

On Next Wizard, Choose your Project Name, and other parameters such as Group ID, Artifact ID. Then Choose Next.





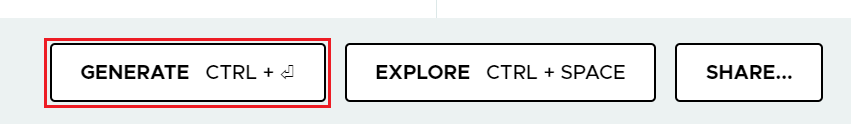
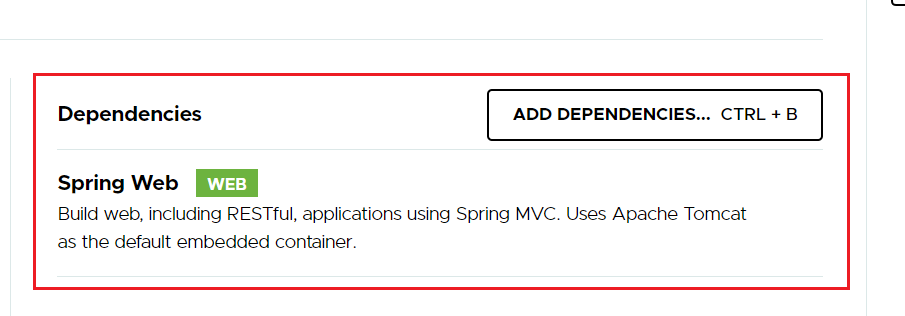
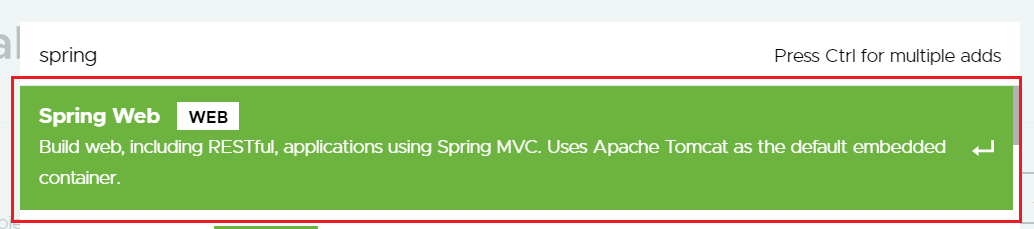
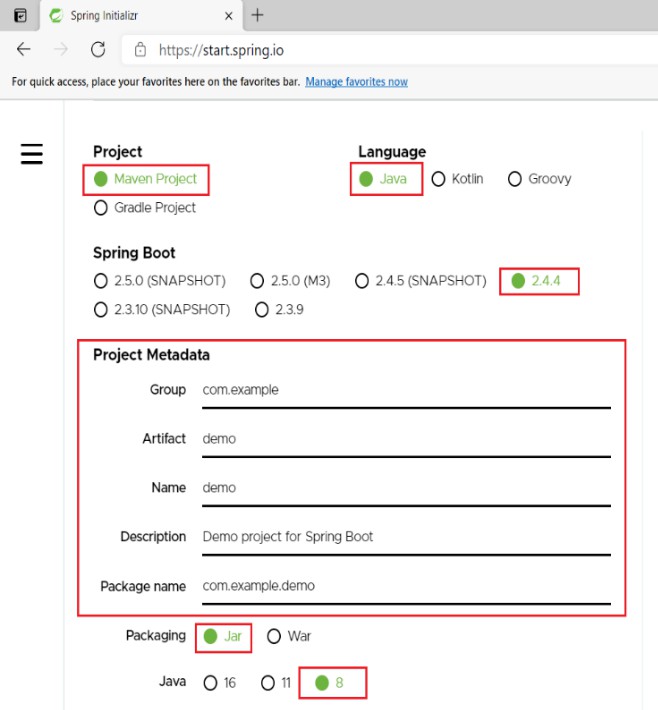
* 1. On next wizards, just click on “Finish”, once it is available.



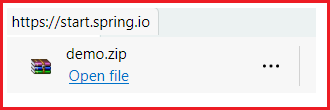
**Step 2 : Go to** <https://start.spring.io/>

**Select All the Options specific to your Machine and Java Version.**

**After click on Add Dependencies -> search Spring Web -> Add Spring Web -> Click on Generate.**

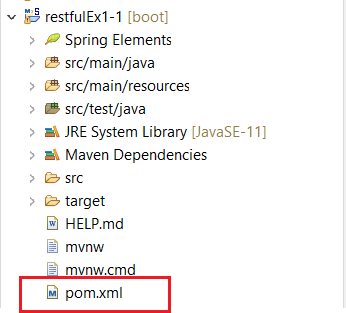


**Click On Generate, a zip file will be downloaded.**

****

**Unzip the downloaded zip file and open the pom.xml file inside the demo folder.**

**Copy the Contents of the pom.xml file & paste it in the pom.xml file of our created project from step 1.**



**Save the file, an automatic download process will start, wait till its completed. Now you are good to go and develop Spring Boot Applications.**

**Problem Statement 1 :** Write a program to create a simple Spring Boot application that prints a message.

Right click on src -> new -> class -> class name-

**Problem Statement 1 :** Write a program to create a simple Spring Boot application that prints a message.

**Solution : HelloWorldController.java** package com.example.demo;

import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloWorldController { @RequestMapping("/") public String hello()

{

return " Hello World...!";

}

}

**Main file will create automatically**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Restful11Application {

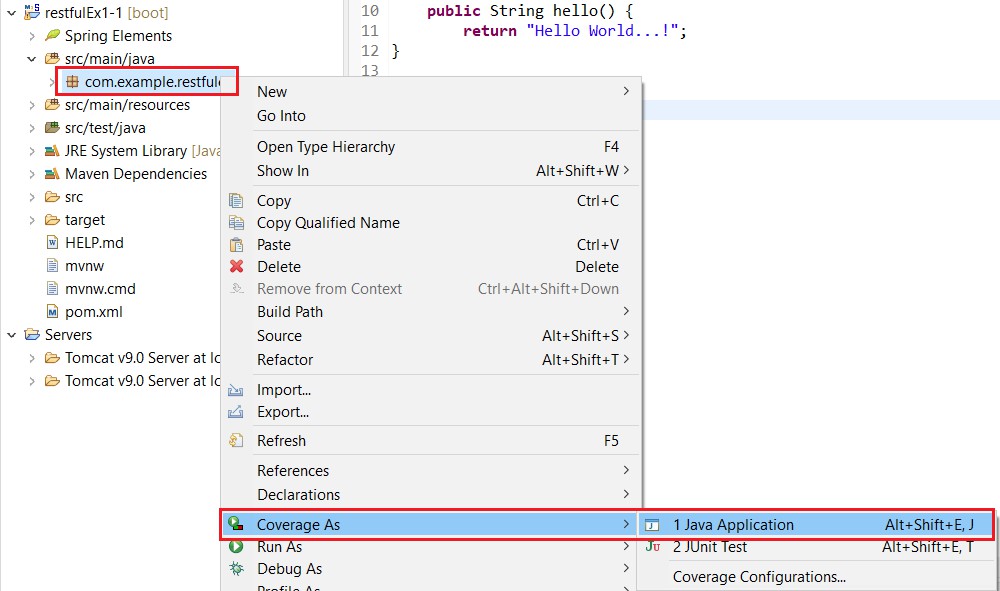
public static void main(String[] args) { SpringApplication.run(Restful11Application.class, args);

}

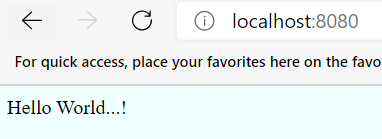
}

**Output :**

Go to Project-> src/main/java -> com.example.restfulex -> Right click on it then Run as -> Java Application



**OUTPUT-**

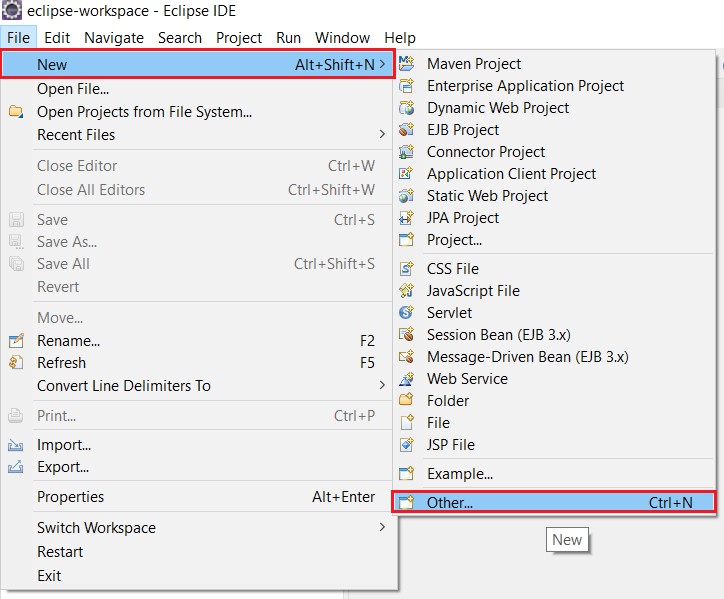


**Problem Statement 10.2 :** Write a program to demonstrate RESTful Web Services with spring boot

Solution :

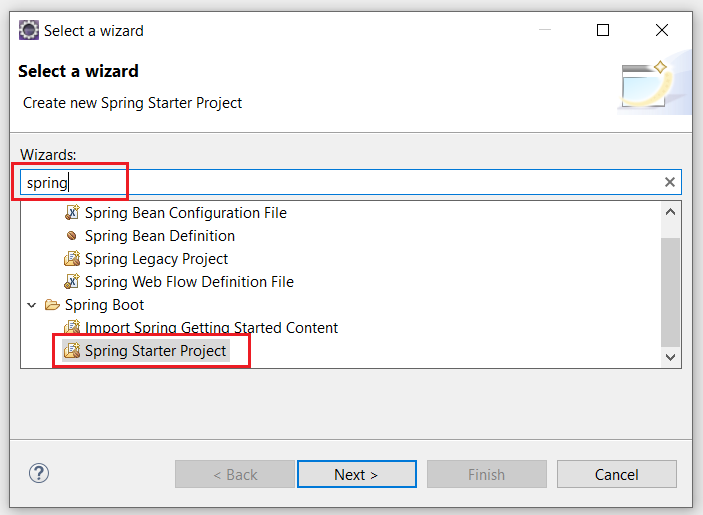
**Step 1 :**

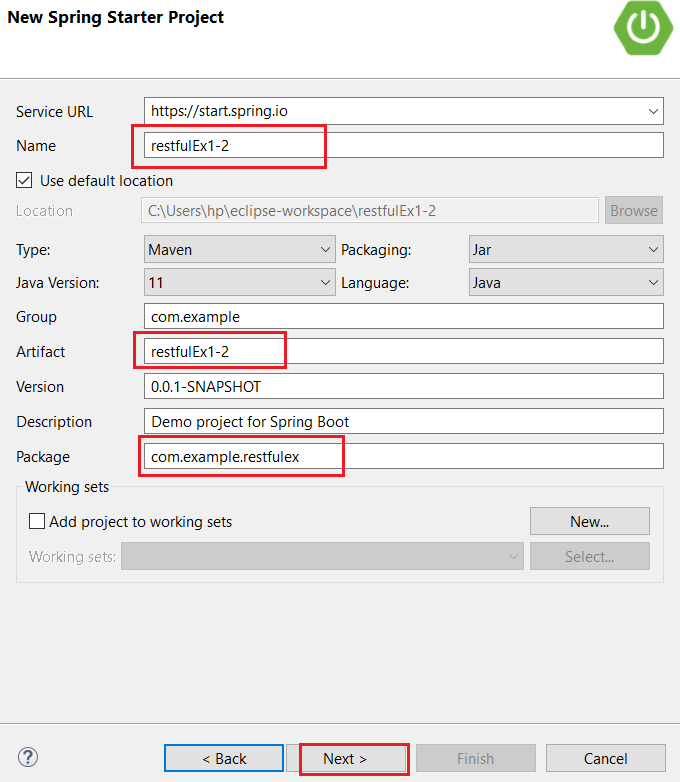
* 1. **:** Open Eclipse. Go To File > New > Other.



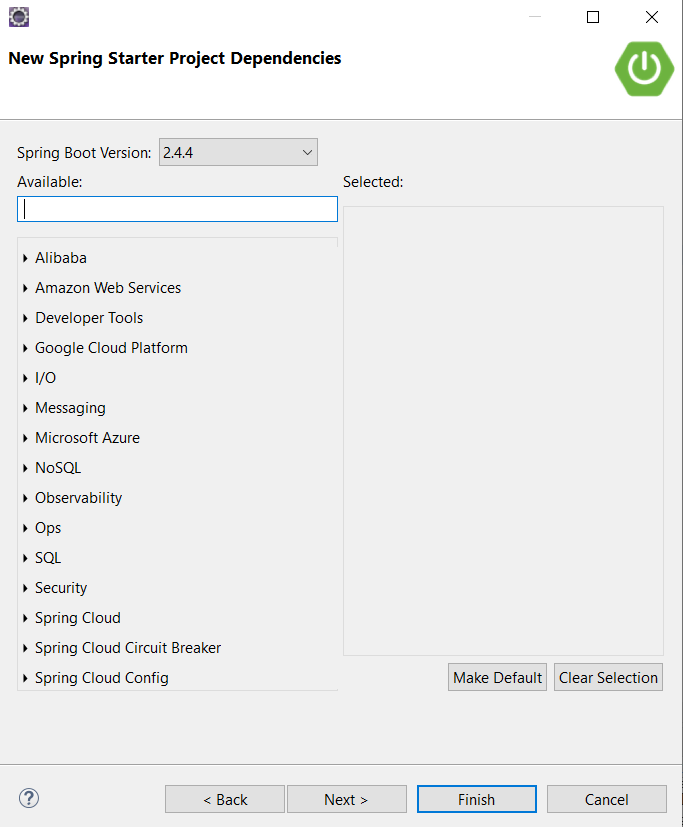
* 1. **:** Search for ‘Spring’ and Select ‘Spring Starter Project’. Then Click on Next.

On Next Wizard, Choose your Project Name, and other parameters such as Group ID, Artifact ID. Then Choose Next.

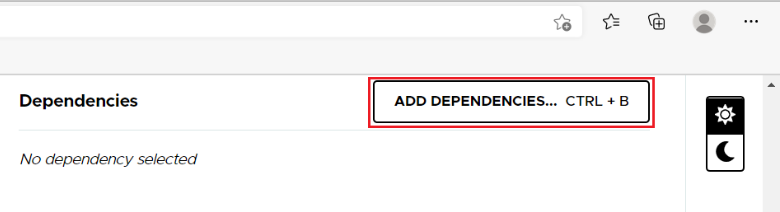
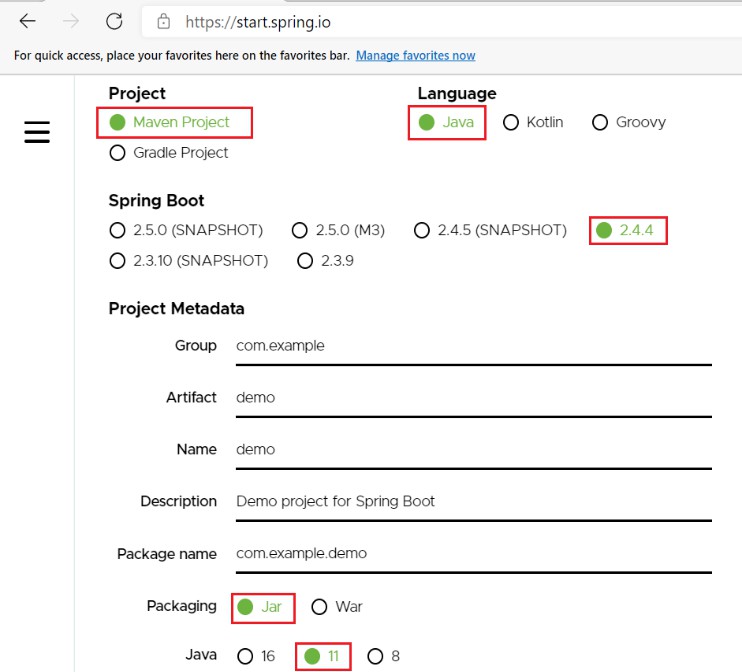




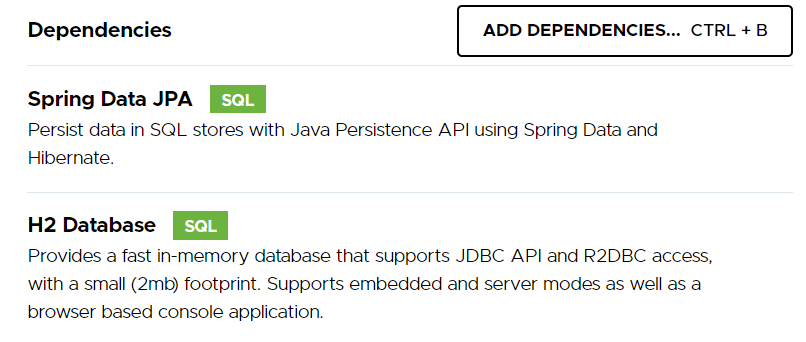
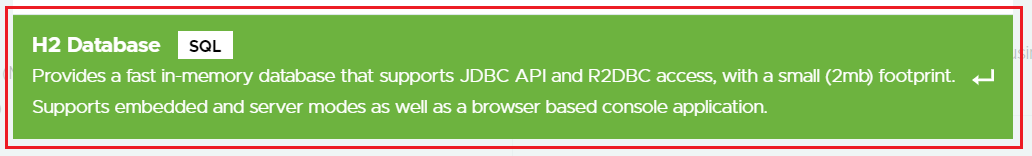
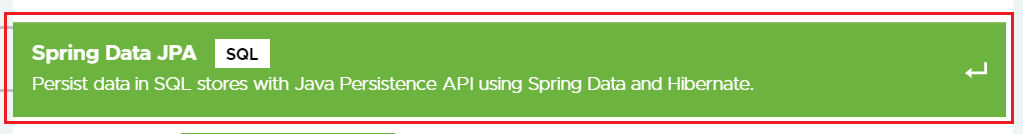
* 1. On next wizards, just click on “Finish”, once it is available.



**Step 2 : Go to** <https://start.spring.io/>

**Select All the Options specific to your Machine and Java Version. After click on Add Dependencies -> search**

**Spring Data JPA and h2 Database -> Add -> Click on Generate.**



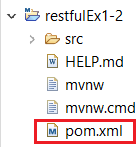


**Click On Generate, a zip file will be downloaded.**



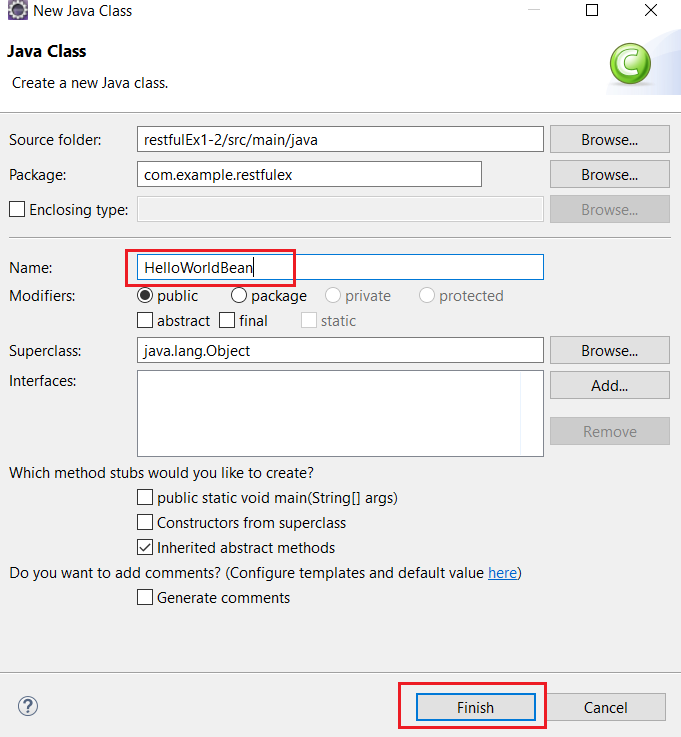
**Unzip the downloaded zip file and open the pom.xml file inside the demo folder.**

**Copy the Contents of the pom.xml file & paste it in the pom.xml file of our created project from step 1.**

****

**Save the file, an automatic download process will start, wait till its completed. Now you are good to go and develop Spring Boot Applications.**

**Problem Statement 1 :** Write a program to create a simple Spring Boot application that prints a message.

Right click on src -> new -> class -> class name-HelloWorldBean

Solution :

**HelloWorldBean.java**

package com.example.restfulex;

public class HelloWorldBean { public String message;

public HelloWorldBean(String message)

{

this.message=message;

}

public String getMessage()

{

return message;

}

public void setMessage(String message)

{

this.message = message;

}

@Override

//generate toString public String toString()

{

return String.*format* ("HelloWorldBean [message=%s]", message);

}

}

**Right click on src -> new -> class -> class name-HelloWorldController HelloWorldController.java**

package com.example.restfulex;

import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloWorldController { @GetMapping(path="/hello-world") public String helloWorld()

{

return "Hello World";

}

@GetMapping(path="/hello-world-bean") public HelloWorldBean helloWorldBean()

{

return new HelloWorldBean("Hello World"); //constructor of HelloWorldBean

}

}

RestfulEx12Application**.java** package com.example.restfulex;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

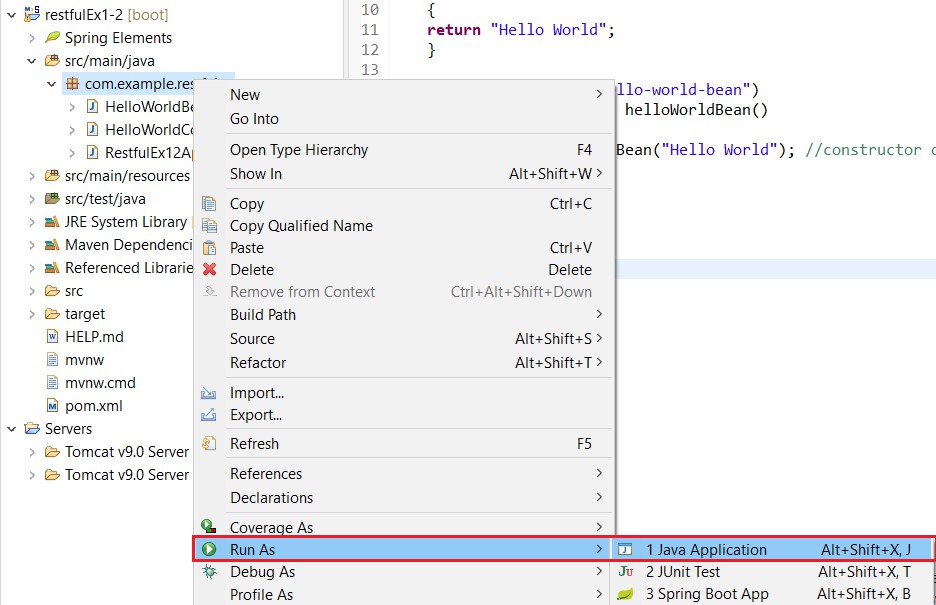
public class RestfulEx12Application {

public static void main(String[] args) { SpringApplication.run(RestfulEx12Application.class, args);

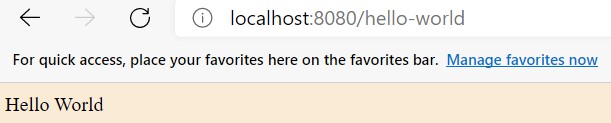
}

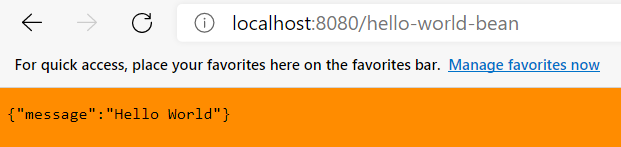
}

Go to Project-> src/main/java -> com.example.restfulex -> Right click on it then Run as -> Java Application



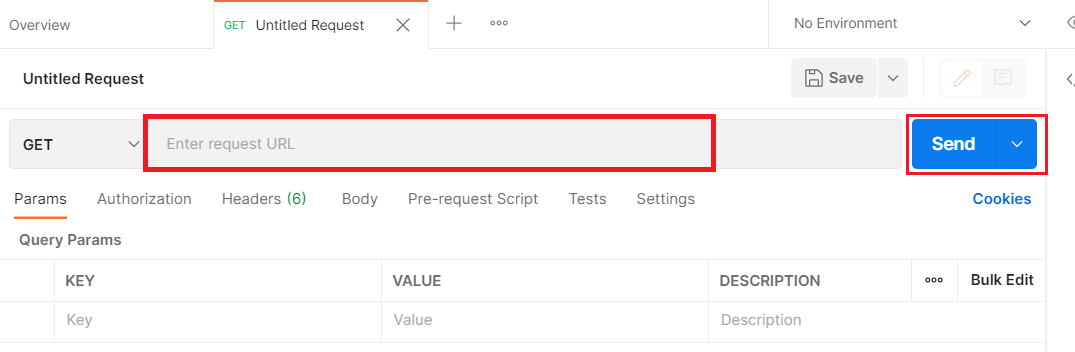
**OUTPUT-**





Testing API with PostMan.

EndPoint : <http://localhost:8080/hello-world-bean>

**Open Poastman -> Enter url**

After pest url-> click on send button.

