Building RESTful Web Services

using Spring Boot

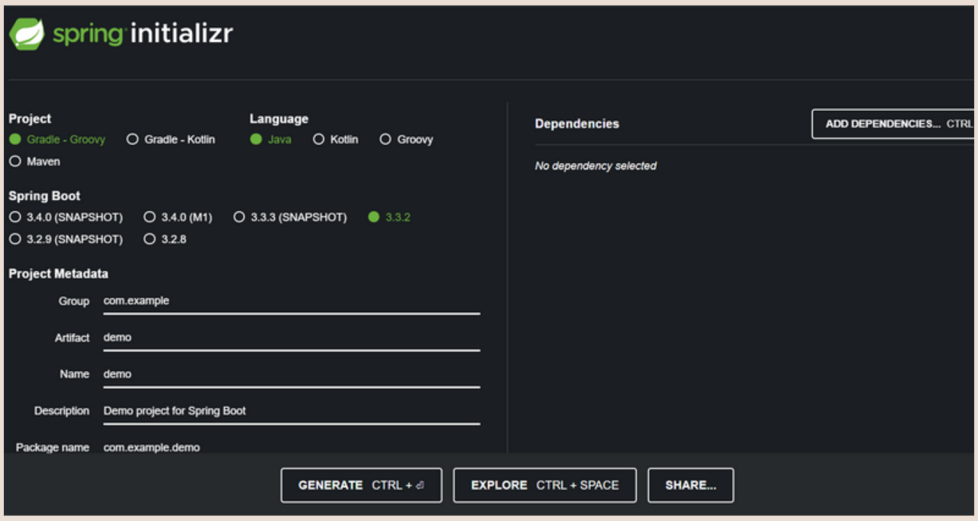
Part - 1

@MILAPMAGAR

Step 1: Set Up Spring Boot

using Spring Initializr.

- Website: https://start.spring.io/



Step 2: Create a REST Controller

DemoController.java

```java
import org.springframework.web.bind.annotation.RestController;
public class DemoController {  no usages
    @RestController  no usages
    @RequestMapping("/api")
    public class DemoControllers {

        // GET method to fetch a message
        @GetMapping("/hello")  no usages
        public String getHello() {
            return "Hello, World!";
        }

        // GET method to fetch a message with a path variable
        @GetMapping("/hello/{name}")  no usages
        public String getHelloName(@PathVariable String name) {
            return "Hello, " + name + "!";
        }

        // POST method to create a new message
        @PostMapping("/message")  no usages
        public String postMessage(@RequestBody String message) {
            return "Message received: " + message;
        }
    }
}
```

Step 3: Application Entry point

DemoController.java DemoApplication.java

```java
package com.milap.myfirstProject;


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

    @SpringBootApplication
    public class DemoApplication {

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

Step 4: Run the Application

Run your Spring Boot application, and you should be able to access the endpoints:

- GET->http://localhost:8080/api/hello-will return "Hello, World!"
- GET http://localhost:8080/api/hello/{name} will return "Hello, {name}!" (replace {name} with any string)
- POST->http://localhost:8080/api/message with a JSON body will return "Message received: {your_message}"

Summary

- **REST Controllers:** Define controllers to handle HTTP requests and responses. Request Mapping: Map HTTP requests to specific handler methods using annotations.
- **GET Method:** Retrieve data from the server.
- **POST Method:** Send data to the server for creation or processing. Path Variables: Capture dynamic values from the URI. Request Body: Bind the HTTP request body to a method parameter.
- **Spring Boot Application:** Entry point to run the Spring Boot application.
- **Testing Endpoints:** Use tools like Postman or curl to test your REST APIs.

Follow for more updates

@milapeeeyyyy

@Milap-Magar