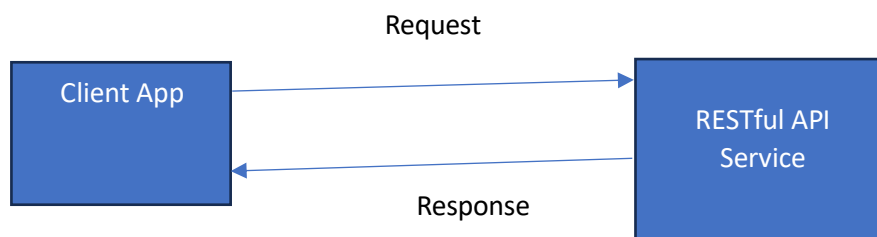


What are RESTful APIs?

REST stands for Representational State Transfer.



In RESTful API, a client send a request to RESTful API Server over HTTP Protocol and RESTful API Service send back response in a plain text format like XML, JSON, etc.

Following are the HTTP Request that are used with RESTful API

- GET
- POST
- PUT
- DELETE

The advantage of using RESTful API is that Client App can be written in any language like C#, Java, PHP, Python, Swift (for Ios Apps).

Similarly the REST API Service can be written in any language like C#, Java, PHP, Python, Swift (for Ios Apps).

Common Clients you can use to test your RESTful API are Webbrowser like Chrome, Firefox, Edge, Safari or a Client like POSTMAN.

Get a Explanation of POSTMAN here.

[postman.pdf \(cppcourses.com\)](https://www.cppcourses.com/postman.pdf)

REST API, RESTful API

or

REST Web Services, RESTful Web Services

or

REST Services, REST Web Services

means the same thing

Now we will create a spring rest application to give output as Hello World

To take output from the application we will use POSTMAN

Go to <https://start.spring.io> and create following demo project

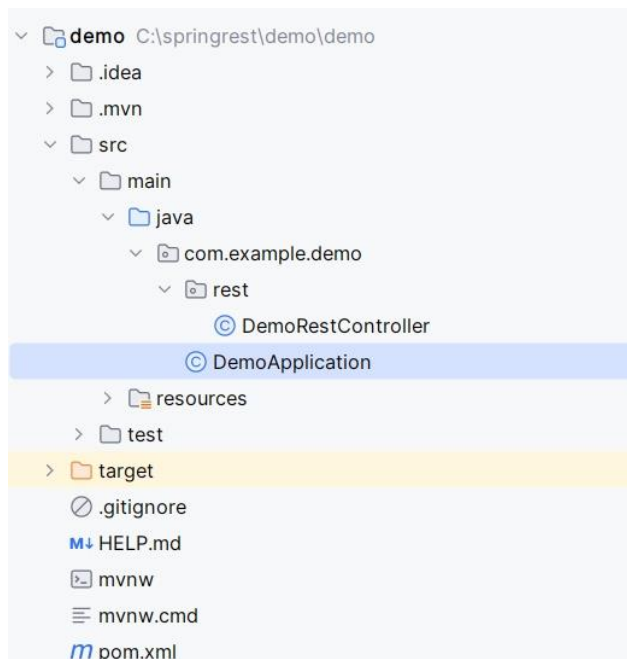


Project <input type="radio"/> Gradle - Groovy <input type="radio"/> Gradle - Kotlin <input checked="" type="radio"/> Maven	Language <input checked="" type="radio"/> Java <input type="radio"/> Kotlin <input type="radio"/> Groovy	Dependencies ADD DEPENDENCIES... CTRL + B
Spring Boot <input type="radio"/> 3.2.0 (SNAPSHOT) <input checked="" type="radio"/> 3.2.0 (M3) <input type="radio"/> 3.1.5 (SNAPSHOT) <input type="radio"/> 3.1.4 <input type="radio"/> 3.0.12 (SNAPSHOT) <input type="radio"/> 3.0.11 <input type="radio"/> 2.7.17 (SNAPSHOT) <input type="radio"/> 2.7.16		Spring Web WEB Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
Project Metadata		
Group	com.example	
Artifact	demo	
Name	demo	
Description	Demo project for Spring Boot	
Package name	com.example.demo	

[GENERATE](#) CTRL + G [EXPLORE](#) CTRL + SPACE [SHARE...](#)

Click Generate and open the project in IntelliJ IDEA Community Edition

Following is the structure for this project



Following is the code for class DemoRestController

```
package com.example.demo.rest;

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

@RestController
@RequestMapping("/test")
```

```
public class DemoRestController {  
  
    // add code for the "/hello" endpoint  
  
    @GetMapping("/hello")  
    public String sayHello() {  
        return "Raman Deep Singh";  
    }  
}
```

Following is the code for class DemoApplication

```
package com.example.demo;  
  
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);  
    }  
}
```

Now run the project on right click on DemoApplication and select Run 'DemoApplication.main()'

Now start POSTMAN

and give following input



http://localhost:8080/test/hello

GET



http://localhost:8080/test/hello

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

	Key	Value
	Key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

Text



1 Raman Deep Singh

Now the text Raman Deep Singh is the output