

Rest API-

Why?

Suppose I have one online website called as fresherworlds.com in which I want to show the tour packages of different countries then what options I have?

There is only one option call the API of tour packages into my site.

That's why Rest API's comes into picture.

Or

If your application supports third party application in that case we can use REST API.

What is REST?

The term REST stands for **RE**presentational **S**tate **T**ransfer. It is an architectural style that defines a set of rules in order to create Web Services.

In a client-server communication, REST suggests to create an object of the data requested by the client and send the values of the object in response to the user.

There are four types of method as

- Get-It is used to read resource
- Post-It is used to create new resource.
- Put-It is generally used to update resource
- Delete-it is used to delete source

Pre-requisites-

1. Create the spring boot project.
2. Pom.xml
3. Postman tool

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```

<modelVersion>4.0.0</modelVersion>
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.4.2</version>
    <relativePath /> <!-- lookup parent from repository
-->
</parent>
<groupId>com.example</groupId>
<artifactId>Example</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>demo</name>
<description>Demo project for Spring Boot</description>
<properties>
    <java.version>1.8</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-
test</artifactId>
        <scope>test</scope>
    </dependency>

    <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>

<build>
    <plugins>
        <plugin>

            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-
plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>

```

Example-

Student.java

```

package com.spring.model;

public class Student {

    private Long id;
    private String name;
    private String city;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }
}

```

```

    }

    public void setName(String name) {
        this.name = name;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }

    @Override
    public String toString() {
        return "Employee [id=" + id + ", name=" + name + ",
city=" + city + "]";
    }
}

```

SpringBootDemoApplication.java

```
package com.spring;
```

```
import org.springframework.boot.SpringApplication;
```

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

```
@SpringBootApplication
```

```
public class SpringBootDemoApplication {
```

```
    public static void main(String[] arg) {
```

```
        SpringApplication.run(SpringBootDemoApplication.class, arg);
```

```
    }
```

```
}
```

StudentController.java

```
package com.spring.controller;

import java.util.ArrayList;
import java.util.List;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RestController;

import com.spring.model.Student;

@RestController
@RequestMapping("/student")
public class StudentController {

    @GetMapping("/getdata")
    public String getTest() {

        return "student data";
    }

    @GetMapping("/getlist")
    public List<String> getStudentData() {
        List<String> list = new ArrayList<String>();
        list.add("ram");
        list.add("10");
        list.add("pune");
        return list;
    }

    @GetMapping("/getStudent")
    public Student getStudentDetails() {
```

```
        Student student= new Student();  
        return student;  
    }  
}
```

Open the postman tool and enter the URL as

<http://localhost:8080/student/getdata>

and press enter button

will see response as

student data