

Date : 11/03/2021
Spring Boot 9AM
Mr. RAGHU

Webservices#Theory
<https://www.youtube.com/watch?v=fvG7FEU1Rt8>

Webservices (ReST)

Http Protocol :

a. Request Method (9)

GET : Fetch Resource(Data) From Provider App

ex: `getEmployees()`, `getEmployeeById()`

POST : Create new Resource(Data) From Provider App

ex: `saveEmployee()`, `createProduct()`

PUT : Modify existed (HighLevel/Full Data) Resource(Data) From Provider App

ex: `updateEmployee()`, `modifyEmpData()`

DELETE : Remove existed Resource(Data) From Provider App

ex: `deleteEmployeeById()`

PATCH : Modify Existed (LowLevel/Partial Data) Resource(Data) From Provider App

ex: `updateEmployeeMailById()`

HEAD
TRACE
OPTIONS
CONNECT

*) All above methods are recommended standards given to all developers in world. They are not MUST method. They are Good Named methods.

Even we can create resource using GET, delete using POST, partial update using PUT..etc They are not the correct usage of Code.

-> This is a contract between two developers, so naming standard are recommended to follow.

-> If we use these Guidelines in wrong way and got output 'It doesn't mean Method Type are wrong'.

--
b. Response Status: (at last what happens?)

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

Code	Messages
1xx	Information
2xx	Success
3xx	Redirect
4xx	Client Side Error
5xx	Server Side Error

*) To define our RestController code (class#method)
must have
 one unique path
 one Http Method
must return
 Data In Global Format(XML/JSON)
 Http Status

*) ResponseEntity<T> contains Body(Data) and Http Status
which behaves like Http Response given back to client.

POSTMAN:

Tool used for testing web/webservice applications.

In simple it is HttpClient that makes request and gets response.

Goto : <https://www.postman.com/downloads/>

Click on Download Link > Next > Next > Finish

> Open POST MAN > In bottom Skip button exist for first time
(DO NOT REGISTER)

-----code-----

```
#
Name: SpringBoot2RestFirstApp
Dep : Spring web

@RestController
package in.nareshit.raghu.controller;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PatchMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/employee")
public class EmployeeRestController {

    @GetMapping("/show")
```

```

        public ResponseEntity<String> showData() {
            ResponseEntity<String> resp =
                new ResponseEntity<String>("Hello
GET",HttpStatus.OK);
            return resp;
        }

        @PostMapping("/save")
        public ResponseEntity<String> createData() {
            ResponseEntity<String> resp =
                new ResponseEntity<String>("Hello
POST",HttpStatus.OK);
            return resp;
        }

        @PutMapping("/modify")
        public ResponseEntity<String> updateData() {
            ResponseEntity<String> resp =
                new ResponseEntity<String>("Hello
PUT",HttpStatus.OK);
            return resp;
        }

        @DeleteMapping("/remove")
        public ResponseEntity<String> deleteData() {
            ResponseEntity<String> resp =
                new ResponseEntity<String>("Hello
DELETE",HttpStatus.OK);
            return resp;
        }

        @PatchMapping("/modifymail")
        public ResponseEntity<String> updateMailData() {
            ResponseEntity<String> resp =
                new ResponseEntity<String>("Hello
PATCH",HttpStatus.OK);
            return resp;
        }
    }
}

```

*) Run App and Test using POSTMAN.

```

---Screen#1-----
GET | http://localhost:8080/employee/show | SEND
-----

```

*) @RestController = @Controller + @ResponseBody + (HTTP
additional)

```

-----
https://www.youtube.com/c/NareshIT/search?query=reactjs%20raghu

```