RESTful WEB SERVICES ----

Monday, July 4, 2022 10:21 AM

Web Service ---- It is a functionality or a feature available on the WEB(remotely) (not on my machine(locally))

Usually the functions that we call are LOCAL Function/Method CALLS --- function is **defined** and **executed** and **called** in the same JVM (LOCALLY)

In Web Services the method is called REMOTELY

Method **definition** and **execution** on server side

Method **call** is on client side

The webservices are platform , language independent

Server could be Python, Java, dot net

Client could be Java , c++ , Python ,React , javascripts

Person1 (Russian) ----------->**Convertor**/translator(Russian to Marathi)---------Person2(Marathi)

<----------**Convertor** (Marathi to Russian) ----------

Web Services

1. SOAP based web services

   XML SOAP format is used for communication --- the neutral language is XML

Client **m1()**------->CALL is converted to XML SOAP -------> XML is converted to server side CALL -- Server **m1(){ …}**

<--Convert return value from XML SOAP to client ----------<convert return value to xml soap

2. **RESTful web service -----** the HTTP is used for communication client call and server method -- the neutral language is HTTP

REST ---------- Representational **S**tate **T**ransfer

Client ------- only knows about **HTTP methods** ----- GET ,POST,PUT,DELETE

Uniform methods !!! Common methods are used by all REST services

All parameters are passed VIA HTTP to the **Mapped methods (remote methods)**

mapped methods RUN on server side

Return value is the DATA (represented in different formats = xml , JSON, plain text, RSS feed , html……)

Data ---- product name = laptop product cost = 50000
Representations of the data :

Plain text	laptop , 50000
Xml	<product> <productName>laptop</productName> <cost>50000</cost> </product>
JSON	{ 'productName' : "laptop" , 'cost':50000' }
HTML	<html> <body> Name = ${productName} Cost = <span>${cost } </body> </html>

REST CLIENT -

1. Another Java program for all methods (simple java program with main)
2. Browser (only for GET or POST simple form / AJAX technology for all method calls)
3. POSTMAN (for all methods)

REST SERVER ---

Tomcat using Spring REST Controller (JAVA)

DOWNLOAD POSTMAN ----

Step 1

1. create a class MyRestController ---- study.controllers
   Annotate this class with @RestController
   --- add a GetMapping("hello")
   Returns "hello world"

Start the tomcat server
, add componentScan

Go to the browser
http://localhost:8080/hello

Add, POST , PUT and DELETE mappings to the REST Controller -- TEST from POSTMAN

---

405 ---------- Method not found ------ the method is not present in the controller

---

General Convention

GET	Should have select queries	R
POST	Inserting a new row	C
PUT	Modifying the data	U
Delete	Removing the data	D

***We can have controller level subpaths and method level paths in the URL --- use @RequestMapping at Controller level

***** Pass Data to REST API**

1. **Using query parameters**
2. **Using path parameters**
3. **Request body**

Using Query parameters ---- http://localhost:8080/postit ? Queryparam1 = kkkkk& queryp=flklsjfsld
Access using **@RequestParam** !!!

In the PUT mapping write a method that accepts a number and returns its square !!! Use Query parameter to pass the num !!!

Using Path parameters ---- http://localhost:8080/first/22/23
Access using @PathVariable

In the POST mapping write a method that accepts 3 numbers and returns their sum !!!Use Path Parameter to pass the data.

Using Request Body -----

___we have to pass the JSON data in request body
Access it using @RequestBody

In the GetMapping write a method that accepts uname in request body and show welcome user name !!!

Mapped methods are returning any type of data---
String
int
We can return an object ----- translated to JSON (Representational format)