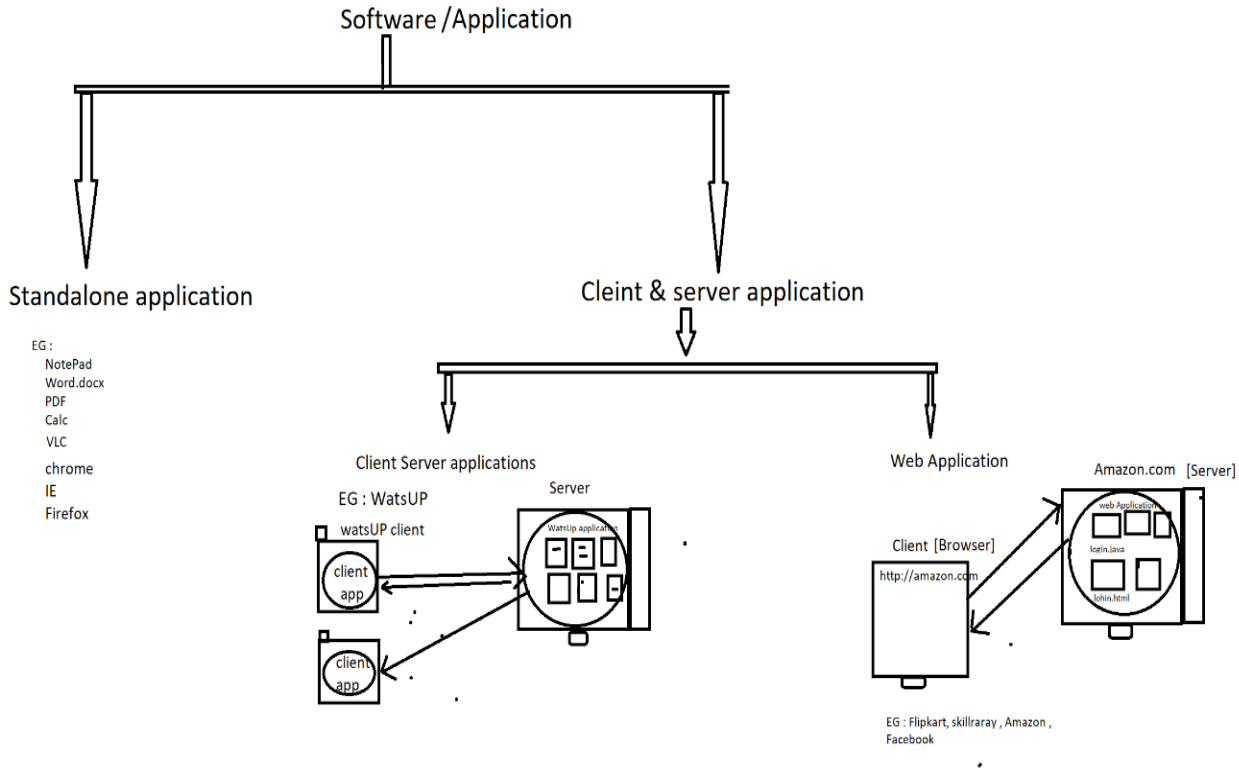
Syllabus

1. Introduction to types of Application
2. Introduction to Web Application
3. Web URL
4. XML
5. JSON
6. Introduction to Web Services Testing
7. Difference between SOAP and Rest Web Services
8. Postman(Installation)
9. Installation of RMGYantra & Database
10. Overview of Postman & RMGYantra
11. CRUD Operation in Postman
12. Environment Variable Setting
13. Test Case Execution
14. Verification in PostMan
15. Random Variable Implementation
16. Running the Collection
17. Request Chaining
18. Running Collection Multiple times with Multiple Data(CSV File)
19. Running Collection in Command Prompt(Newman & Node JS)
20. Authentication
21. Parameterisation
22. Postman Grooming & Transalation

Introduction to Types of Application

Software:

Software is a set of instructions, data or programs used to operate computers and execute specific tasks



Standalone application:

The Application which is installed in computer, where those application can be accessed by one user at a time is called Standalone software.

EG :

Calc, Word, Excel, Notepad

Client server application:

client/server application consists of a client program that consumes services provided by a server program. The client requests services from the server by calling functions in the server application

➔Pure client server application is always accessed via client software

EG : WhatsApp, Skype, GoToMeeting

Introduction to Web Application

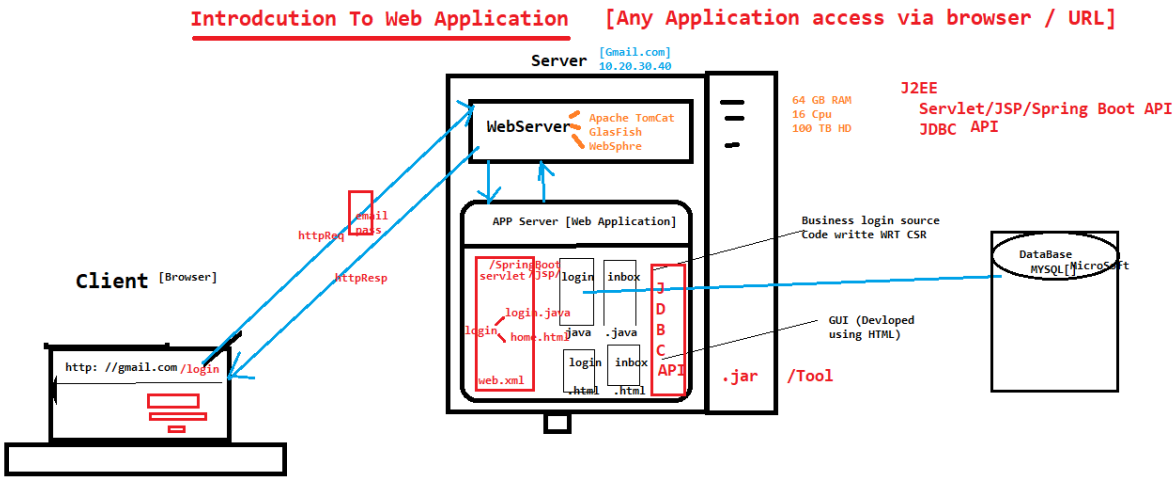
Web Application:

The application which is accessed via browser & URL is called web application &

Browser is one and only way to access any web application

➔Pure web application is always accessed via URL

EG : Facebook, Amazon, Flipkart, Irctc.com,MakeMyTrip



Web Browser

- **It is a standalone application which is used to access any web application over the network via URL**
- **Browser understands and interpret only html language so that it always creates http request and interpret httpResponse given by web server**
- **It's one & only way to access web application**
- **Browser always send / receive request via http Protocol**

EG :

chrome, IE , safari , firefox, Uc , opera etc

Http Protocol: it's a set of instruction or rules, & it's communication media between browser & webserver, http protocol by default understand html language

Web Server:

- **Like any other application (Adobe Reader, Media Player, VLC etc.,) , Webserver is also an application which runs on Operating System, & every AppServer(webApplication) is under control of Webserver**
- **Webserver as the name implies "Serves requests to a Browser, it helps both web browser & web Application to interact with each other**
- **It always receives the request from the browser & talk to web application & provide response back to browser**

Few Examples:

- **Apache Tomcat**
- **Apache JBOSS**
- **IBM WebSphere**
- **Oracle WebLogic**
- **Oracle GlassFish & many more…**

App Server / Web Application:

- **Collection of web Resources present inside webApplcation called AppServer**
- **AppServer contains business logic of the source code , & html code , Servlet Code , JDBC code**
- **Web Resource are always implemented using Servlet or JSP or SpringNBoot for JAVA , ASP.net for C#**

Web Resource: where business logic code is written

Two types of WebResource

1 . Static WebResource : response present inside the webApplication before the request

Wikki pedia

Documentation WebSite

Note: to develop static WebApplication only HTML is enough

When web plication is a collection of static web Resource is called static web Application

2. Dynamic WebResource: response getting generated at the time of request

Ex : internet Banking, Vodafone bill, Gmail, Facebook

When webAplication contains at least one dynamic webResource is called Dynamic web Application

Note: servlet or JSP or Spring boot is mandate to develop dynamic Web application

Data Base

- **It also an application which helps us to store the data in the form of table.**

Example :

- **MySql**
- **MsSql**
- **Oracle 10 g**
- **DB2**
- **Mongo DB**
- **NoSql**
- **Sybase**

JDBC API :

- **It is collections of J2EE API , which is used to connect to data base from the java Program**

- **Using JDBC, we can store & fetch the data from the database**

  EG:

  mysql-jdbc-connector.jar

  Ojdbc.jar

  Db2-jbdv.jar

Servlet:

- **Servlet is a collection J2EE API, used Its used to develop dynamic web Application**
- **Servlet also contains web.xml file, which is used for mapping between java & html file**

What technology your Application Built on ?

Front end: html-5 or Anguler-JS or React JS

Backend : java + (SpringBoot or Servlet or JSP)

Database : Mysql or Oracle-11g or DB2 or MANGO db

What is URL?

Unified resource location: used to identify specific webresourse inside the web application, URL is one & only way to accesses any web application via browser

Web URL :

- Web Uniformed Resource Location (URL): uniquely identify the specific web resource inside the web application
- Every web application should have its unique address in the form of URL
- URL is one & only way to access web application via browser
- Max number of charterer in the URL is 200

Syntax

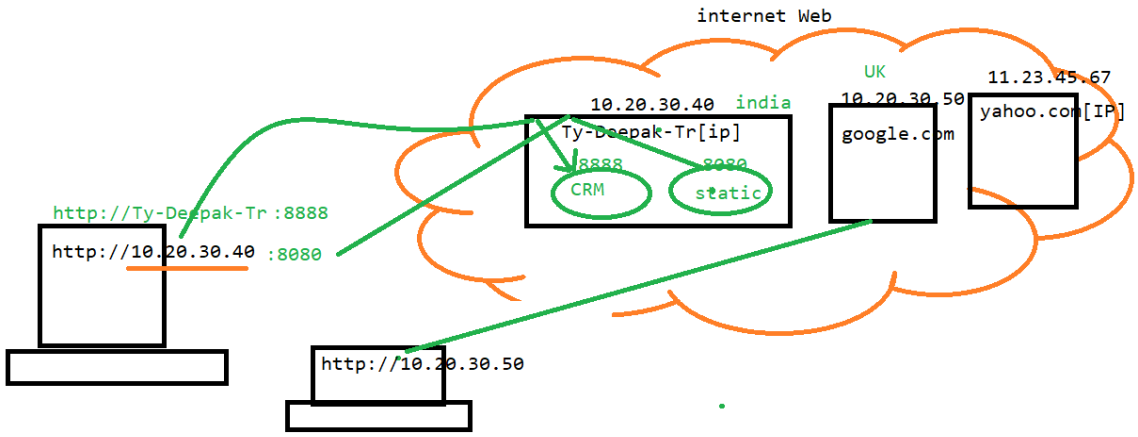
protocal://<domainName>:port/resourcePath?queryString#fragementId

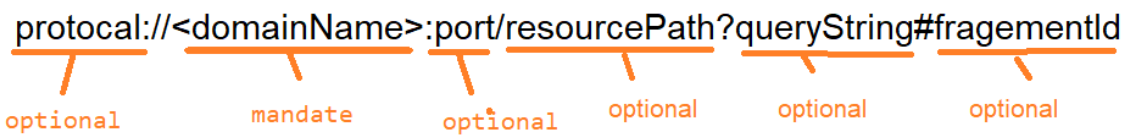
EG :

http://localhost:8888/index.php

http://google.com/search?q=iphone11

http://172.217.160.142/search?q=iphone11





Protocol

- It's a common language where two application exchange information each other
- When one application wants to communicate with each other (in our case browser & server), these need to be a common language which both applications understand each other
- This language is known as protocol, where protocol as set of rules & instruction
- Browser always send a request & receive response via http protocol hence it is called http request / http response.
- It's an optional information & case-insensitive
- Types of Protocol

	Tomcat	JBoss	WebLogic
http	**8080**	**80**	**123**
https	**8443**	**443**	**456**

- ✓ http
- ✓ https
- ✓ ftp
- ✓ smtp etc..

Domain Name

- Uniquely identify the specific computer in the network in which web application in present
- Domain name might be computer name or iP addresses
- It's a mandatory information & case sensitive

Example:

www.amazon.com (commercial)

.org (organization)

.edu (education)

.gov.in (government**)**

Port:

- Uniquely identify the specific software/application inside the computer
- In the URL this is an optional information
- Below table specify the default port number

Resource Path:

- Uniquely identify the specific web resource inside the application Server (web application)
- We know web application is a collection of web resources, so it's a full path of web resource at web application side

Example :

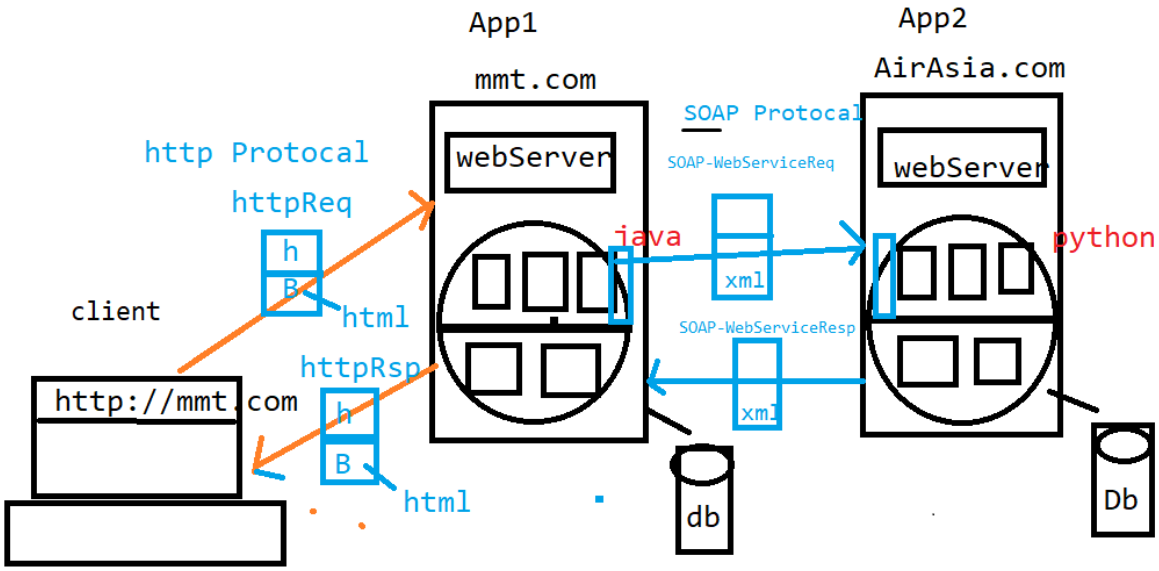
http://localhost:8080/index.html

http://localhost:8080/login

Fragment ID :

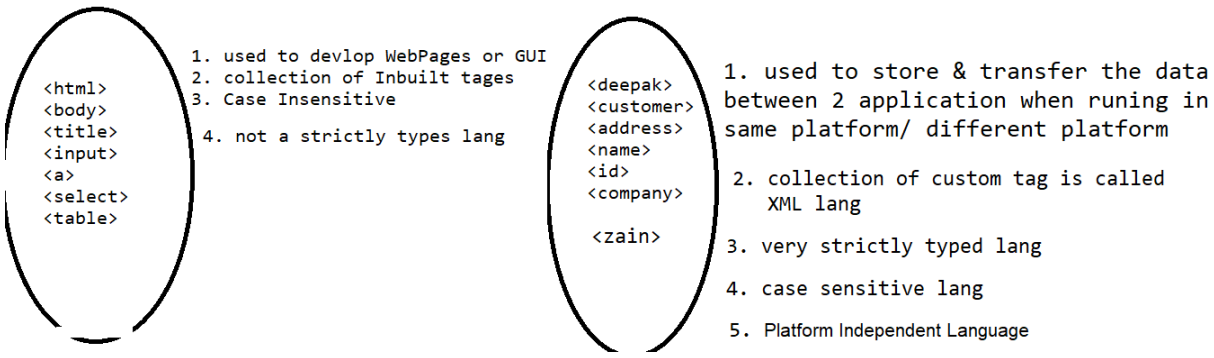
- Identify the specific fragment / section in the webpage
- It's an optional information in the url

Extensible Mark-up Language (XML):



- XML is "Markup Language & Platform Independent Language" which helps to store and transport data.
- Different Applications which are developed using different technologies or same technologies can Transfer the Data among themselves with the help of XML.
- As the name implies it's an extension of HTML & hence XML looks similar to HTML but it's not a HTML
- XML has User-defend Tags.
- XML tags are also called as "elements".

HTML vs XML



XML Syntax

- XML is "Strictly Typed" Language hence
- Case-sensitive
- They cannot contain spaces
- For every element data, “data-type” should be defined,
- Every opening element should have corresponding closing element and also XML elements must be properly nested/closed
- They must start with a letter or underscore
- They are cannot start with the letters like xml or XML or Xml etc.
- MIME type (Content Type) of XML is "application/xml“
- File extension of XML is ".xml"

Rule 1: XML Prolog :

Below line is called as "XML prolog", which is optional. If it exists, it must be the First Line of XML

EG :

<?xml encoding=’UTF-8’ verion=1.1 scema=http://testing.xom…>

Rule 2: Xml Comments

- The syntax of XML comment is similar to that of HTML

  Ex:

  <!--This is a comment -->

Rule 3 : XML Structure

- Like HTML, XML follows a Tree Structure
- An XML tree starts at a "root element" and branches from

"root element" will have "child elements“

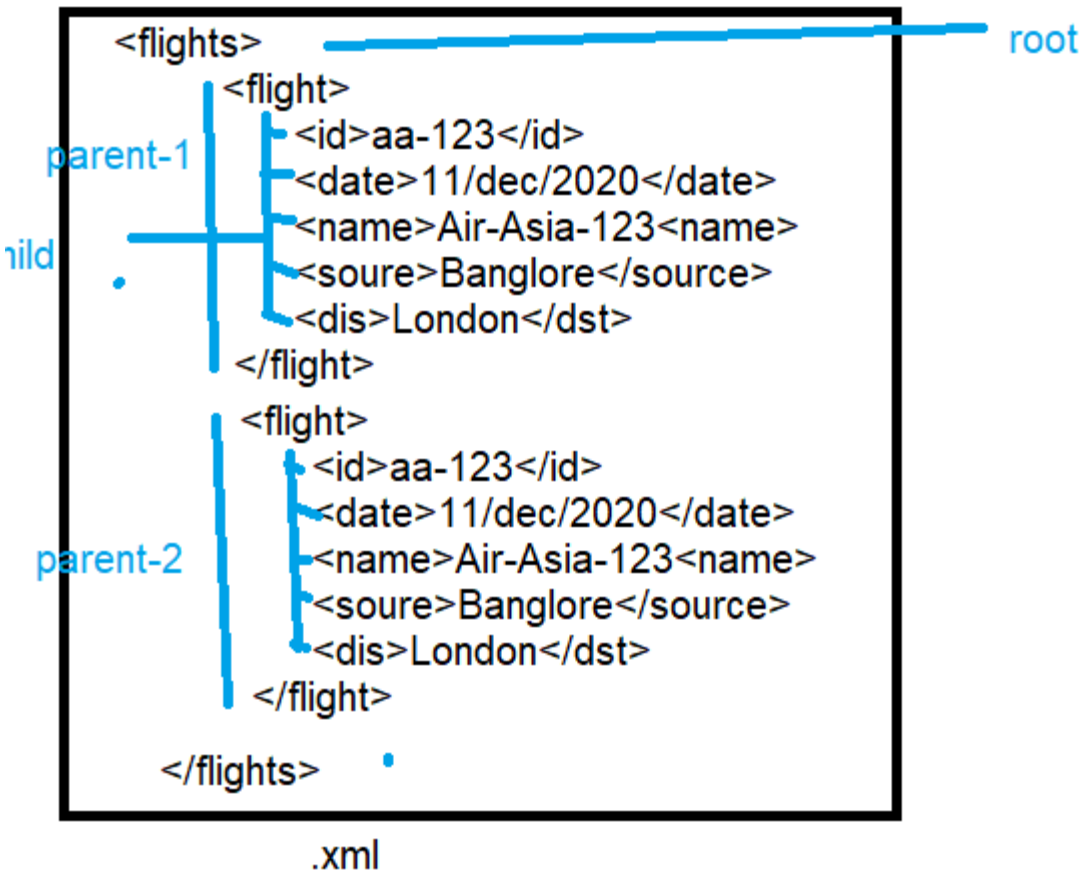
- XML Consists of "Only One" root element

which is parent of all other elements child elements

child can have "sub elements / child elements

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

```
<employee>
        <name>Deepak</name>
</employee>
```



Rule 4 : XML-Entity References

- Some characters have a special meaning in XML.
- If you place a character like "<" inside an XML element,
- it will generate an error because it represents the start of a new element

Ex:<message>salary<1000</message>

- To avoid this error, we can replace the "<" character withan "entity reference" as shown below

<message>salary <1000</message>

There are 5 pre-defined entity references in XML:

< < less than

> > greater than

& & ampersand

' ‘ apostrophe

" " quotation mar

Rule 5 : XML PCDATA: Parsed Character Data

- Text between start-element and end-element is called as PCDATA which will be examined by the parser

Example:-

<employee>Ram</employee>

The string "Ram" is considered as PCDATA

PCData : will be always consider as String

Rule 6 : XML-CDATA: Character Data

- If XML data contain many special characters, it is good practice to replace all of them. Instead we can use "CDATA (character data) section“

Example: -

<employee>emplyeSal >1000 & sal < 10 </employee> : Wrong

<![CDATA[<emplyeSal >1000 & sal < 10 </employee>]]>: Correct

Rule 7 : XML-Elements & Attributes

- XML element is everything from (including) the element's start tag to (including) the element's end tag
- An element can contain:

1. data
2. Attributes
3. other elements
4. All of the above

➔**XML-Attributes**

- Like HTML, XML elements can also have attributes, but **attributes can't easily expandable like elements**
- XML Attributes Must be Quoted either single or double quotes can be use
-

```
EG :1                              EG:2
     <person gender = 'male'>          <person>
         <name>deepak</name>              <gender>male</gender>
     </person>                            <name>deepak</name>
                                      </person>
```

Example 1 gender is an attribute

Example 2 gender is an element

➔XML Elements

Will go Elements when data extendable

```
<?xml version=1.0 , Encoding="UTF-8 scema='http://resource.com.dtd'>
<person>
        <id>sk-01</id>
        <name>deepak</name>
        <gender>male</gender>
        <mobileNum>9886662262</mobileNum>
<person>
```

```
<person id="sk-01"  gender="male">
    <name>deepak</name>
    <mobilenum>9886662262</mobilenum>
</person>
```

or

```
<person id="sk-01"  gender="male">
    <name>
          <fname>deepak</fname>
          <lname>gowda</lname>
     </name>
    <mobilenum>9886662262</mobilenum>
</person>
```

Rule 8 : XML Schema's

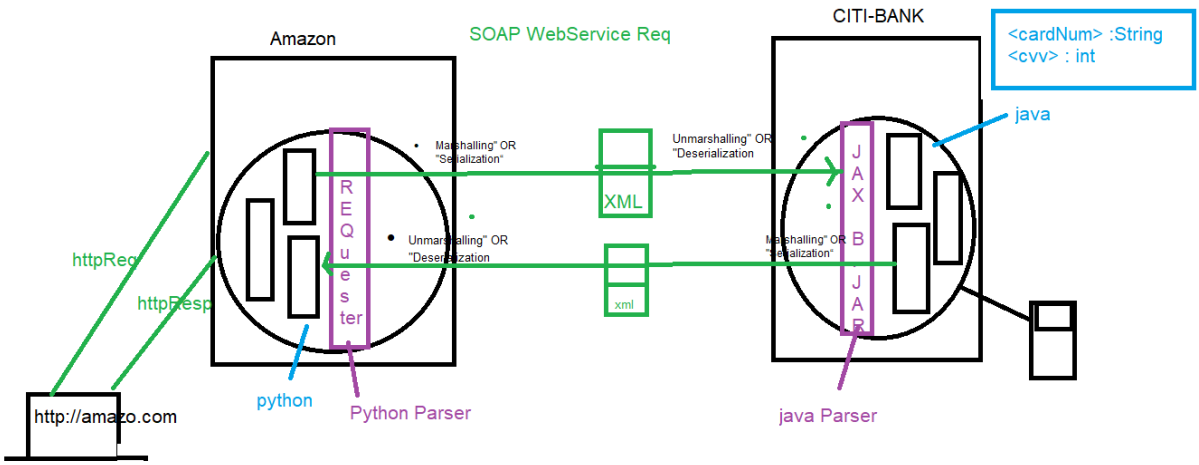
- W.K.T XML helps us to store & transfer the data
- When sending data from one application to an another, it is essential that both applications have the same "expectations / agreement" about the content/data
- for example, A date like "03-11-2004" -in some countries, be interpreted as 3rd November and -in other countries as 11th March

There are two ways to define a Schema for XML

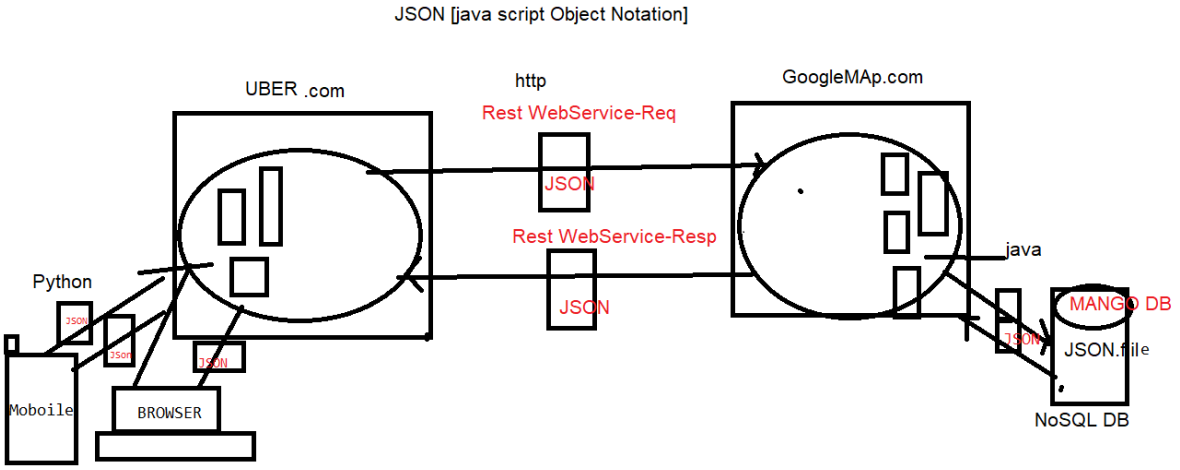
1. Document Type Definition (DTD)
2. XML Schema Definition (XSD)

XML-PARSAR(JAXB)

- JAXB is a Java API helps us to convert Java Object to XML & vice-versa
- The Process of converting Java Object to XML is called as "Marshalling" OR "Serialization“
- The Process of converting XML to Java Object is called as "Unmarshalling" OR "Deserialization



JavaScript Object Notation [JSON]



- Like XML, JSON also is a "Language" & Platform Independent Language" which helps to store and transport data
- However, compared to XML, it's a lightweight, easy for applications to parse and generate by avoiding complicated parsing and translations
- JSON, as the name implies, which consists of data similar to "Object Notation of JavaScript".It's an extension of JavaScript
- Hence if we receive data from a server in JSON format, we can directly use it like any other JavaScript object
- The filename extension of JSON is ".json"
- MIME type (Content Type) of JSON is "application/json"

- JSON syntax is derived from JavaScript object notation
- Examples of Data Formats:

JAVA

```
String str1 = "
00.12";
String str2 = "EmpID=123 |  EmpNM=Deepak |  EmpSal=200.12";
```

XML

```
<employee>
    <emp-id>123</emp-id>
    <emp-name>Deepak</emp-name>
    <emp-salary>200.12</emp-salary>
</employee>
```

JSON

```
{
   "EmpID":123,
 "EmpNM":“Deepak",
  "EmpSal":200.12
 }
```

JSON Syntax

Data is in "name : value" pairs

Data is separated by "commas“

“ Curly braces" hold objects

"Square brackets" hold arrays

JSON Data

- JSON data is written as name/value pairs.
- A name/value pair consists of a field name (Should be in double quotes) followed by a colon-followed by a value

Ex:

"employee-name“ : “Deepak”

JSON value

- In JSON, values must be one of the following data types

1.String

2.Number

3.Boolean

4.NULL

5.an Object (JSON object)

6.an Array

7.an Object Array

In JSON,

-String values must be written with double quotes

-Numbers must be an integer/decimal values

-Boolean values must be true/false

-JSON NULL values must be null

EG:

{ "name":“Deepak", "age":35, "isEmployed":true, "girlFriend":null }

JSON Object

- Values in JSON can be objects
- JSON Objects are -surrounded by curly braces { }

-JSON object data is written in "key:value" pairs

-Each "key:value" pair is separated by a comma

-Keys must be String and Values must be a valid

-JSON data type (String, Number, Object, Array, Boolean or null)

- EG :

```
{
   "employee":{
                  "name":"Praveen D",
                  "age":33,
                  "isEmployed":true,
                  "girlFriend":null
                 }
}
```

JSON Array

- Values in JSON can be arrays
- JSON Arrays are -surrounded by "Square Brackets [ ]“

-JSON Arrays values is separated by a comma

-Array values must be a valid JSON data type

-(String, Number, Object, Array, Boolean or null

Example 1:-

```
{
  "employees":[ “deepak", “ram", "Malleshwar" ]
}
```

Example 2:-

```
{
   "name":“Deepak",
   "age":33,“
    cars":[ "GM", "BMW", "Audi" ]
}
```

XML

```
<person>
      <id>sk-01</id>
      <name>deepak</name>
      <gender>male</gender>
      <mobileNum>9886662262</mobileNum>
</person>
```

JSON

```
{
   "id" : "sk-01",
   "name" : "deepak",
   "gender" : "male"
   "mobileNumber" : 9886662262
}
```

```
<person>
    <name>Jhon</name>
     <mobileNum>
              <primaryNum>988000000</primaryNum>
              <secondNum>988000011</secondNum>
     </mpbileNum>
</person>
```

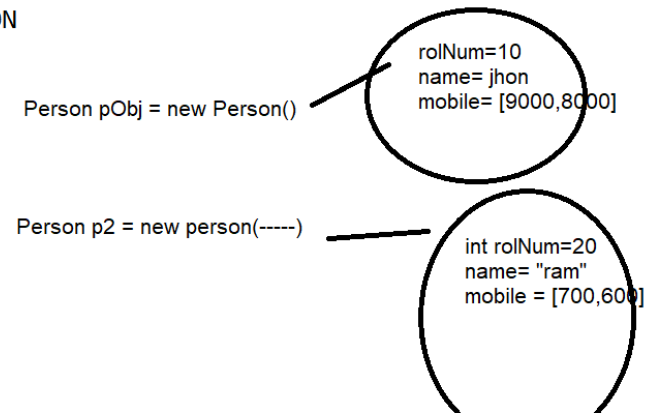
```
{
   "name" : "  jhon   ",
   "mobileNum" : [988000000, 9880000011]
}
```

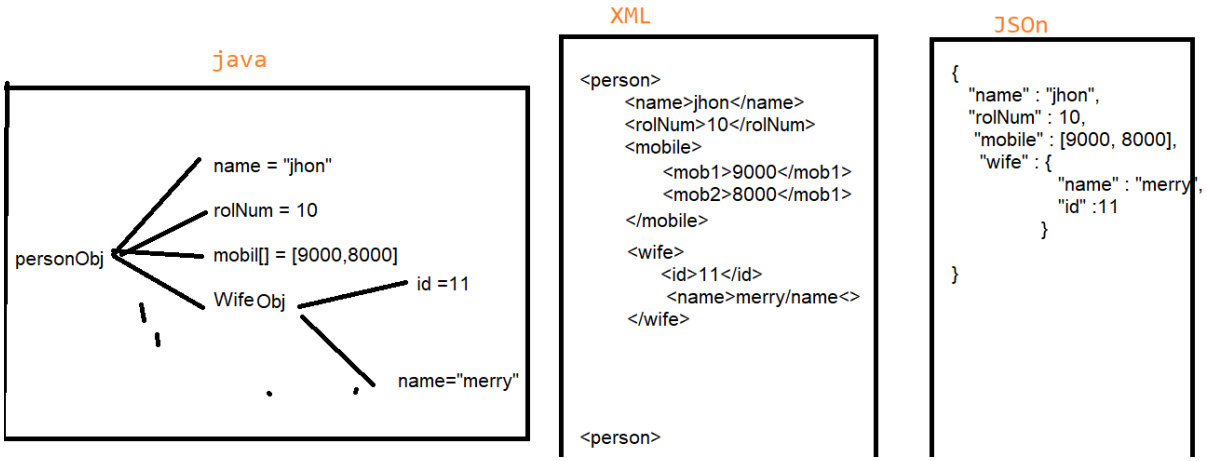
```
<person  id='123'>
     <name>jhon</name>
     <mobilenum>9900099</mobileNum>
      <girlFriend>null </girlFriend>
     <wife>
           <name>merry</name>
            <mobileNum>99999</mobileNum>
     </wife>

<person>
```

```
{
   "name" : "jhon",
   "mobilenum" : 9900099,
   "girlFriend" : null,
    "id" : 123
   "wife" : {
              "name" : "merry",
              "mobileNum" : 999999
            }

}
```

EG : JAVA Object / Vs / XML / JSON

```
class Person{
      int rolNum;
      String name;
      long[] mobile;
      public Person(int rolNum, String name,
long[] mobile) {
            this.rolNum = rolNum;
            this.name = name;
            this.mobile = mobile;
      }
}
```



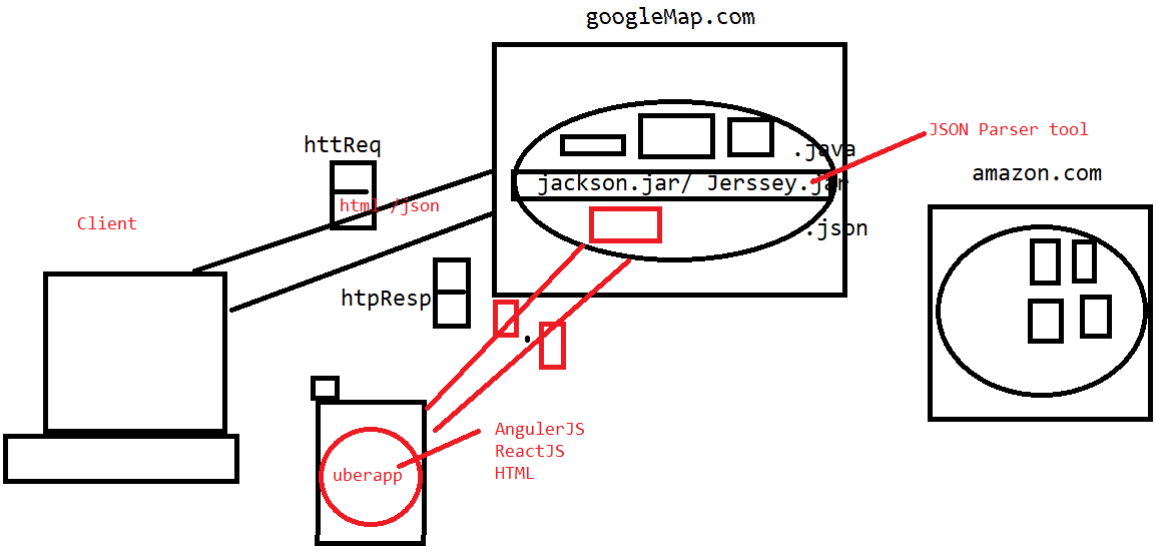


JSON /XML

JSON	XML
JSON data has a data type , light weight	XML data is type less
JSON types: string, number, array, Boolean, null ,Object	All XML data should be string
Data is readily accessible as JSON objects	XML data needs to be parsed.
JSON is supported by most browsers.	Cross-browser XML parsing can be tricky
Retrieving value is easy	Retrieving value is difficult
A fully automated way of DE serialization/serialization JavaScript.	Developers have to write JavaScript code to serialize/ de-serialize from XML
Native support for object.	The object has to be express by conventions - mostly missed use of attributes and elements.
It supports only UTF-8 encoding.	It supports various encoding.
It doesn't support comments.	It supports comments.
JSON files are easy to read as compared to XML.	XML documents are relatively more difficult to read and interpret.
It is less secured.	It is more secure than JSON.

JSON Parser

- JAX-RS, JACSON, JERSEY is a Java API helps us to convert Java Object to JSON& vice-versa
- The Process of converting Java Object to JSON is called as "Marshalling" OR "Serialization“
- The Process of converting JSON to Java Object is called as "Unmarshalling" OR "Deserialization



Write Java Parsing Program to convert JAVA To JSON

Or

Write Sterilizations java program to convert java to JSON

OR

Write Marshalling program to convert JAVA to JSON

Answer

Step 1 : add Jackson dependency in POM.xml file

EG :

```
<dependency>
  <groupId>org.codehaus.jackson</groupId>
  <artifactId>jackson-mapper-lgpl</artifactId>
  <version>1.9.13</version>
</dependency>
```

Step 2 : create Java business class [Pojo Class]

```
   package pac;

public class Employe {
      int rolNum;
      String name;
      boolean mstatus;
      int[] moobileArr;
      public Employe(int rolNum, String name, boolean mstatus, int[]
moobileArr) {
         this.rolNum = rolNum;
           this.name = name;
           this.mstatus = mstatus;
           this.moobileArr = moobileArr;
      }
      Employe(){

      }

      public int getRolNum() {
            return rolNum;
      }
      public void setRolNum(int rolNum) {
            this.rolNum = rolNum;
      }
      public String getName() {
            return name;
```

```
	}
	public void setName(String name) {
		this.name = name;
	}
	public boolean isMstatus() {
		return mstatus;
	}
	public void setMstatus(boolean mstatus) {
		this.mstatus = mstatus;
	}
	public int[] getMoobileArr() {
		return moobileArr;
	}
	public void setMoobileArr(int[] moobileArr) {
		this.moobileArr = moobileArr;
	}


}
```

Step 3 : create one Java Object to Pojo class

Eg :

```
int[] arr = {988777,98887};
	Employe emp = new Employe(10, "yogesh", false, arr);
```

Step 4 : write parsing program using JACKSON

```
ObjectMapper objMapper = new ObjectMapper();
		objMapper.writeValue(new File("./employe.json"), emp);
```

OUTPUT :

employe.json

Write Java Parsing Program to convert JSOn to JAVA

Or

Write deSerilizations java program to convert JSON to JAVA

OR

Write UN Marshalling program to convert JSN to JAVA

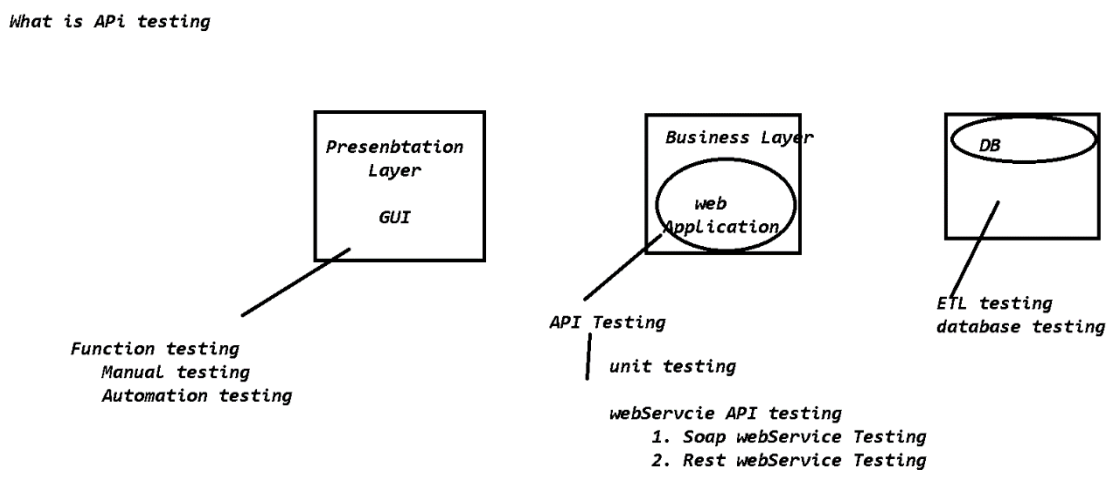
Step 4 : Parsing program to convert JSON to JAVA

```
ObjectMapper objMapper1 = new ObjectMapper();
Employe e1 = objMapper.readValue(new File("./employe.json"),
Employe.class);

  System.out.println(e1.getName());
```

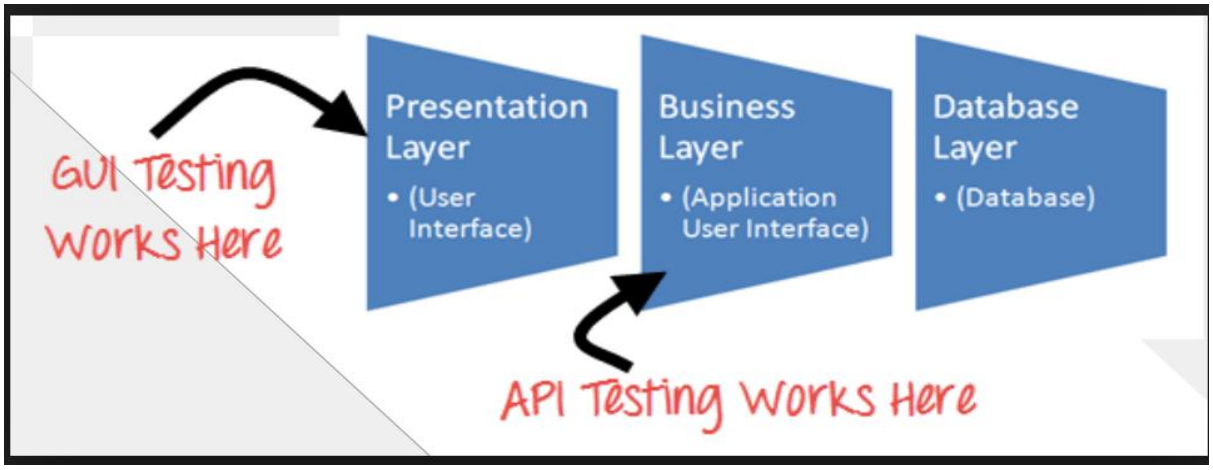
Web Service introduction

API testing

- API testing is a type of software testing where application programming interfaces (APIs) are tested to determine if they meet expectations for functionality, reliability, performance, and security.
- Testing the application in Source code layer (Business layer) is called API testing
- Testing interface between two Application is also called API testing
- Testing is done without browser is called API testing



SOA [Service Oriented Architecture]



Difference between Frontend & Backend Testing

Front End testing	*Back End testing*
Front-end is always performed on GUI	*Back-end testing is done on Business layer and engineer should have knowledge on Data base and backend*
Testing becomes slower because it have to wait for browser rendering time	*It is faster and saves time.*
Performance testing Is not easy in front-end	*Performance testing is easy and faster*
Adhoc, usability and compatibility testing are possible	*Adhoc, usability and compatibility testing are not possible*
Manual & selenium automation will be done at front end(BBT)	*white Box, Webservice testing will be done in the backend (WBT. GBT)*
End to end testing is easy	*End to end testing is difficult*
Tools: selenium, QTP, test complete etc	*SOAP UI, POSTMAN, RESTCLIENT, RESTASSURED, testNG , JUNIT*

Difference between Unit & API Testing

UNIT testing [WBT]	API testing [Webservice testing]
Done by development team	Done by manual testing , API testing team or SDET team
Should have knowledge on source code & database	Should have knowledge on expectations and agreements of API [API –Swagger Doc]
Test each and every source code of the application	Test the data flow between 2 applications
It is a white box testing	It is a kind of grey box testing
Tools: eclipse+testNG	Tools: SOAP UI, POSTMAN, RESTCLIENT, RESTASSURED

API testing Types

Types of API testing

1. Whitebox testing/ Unit Testing
2. WebService Testing
    2.1 Soap WebService API-testing
    2.2 RESTFull WebService API-testing

Unit Testing [White Box testing or Unit API testing]

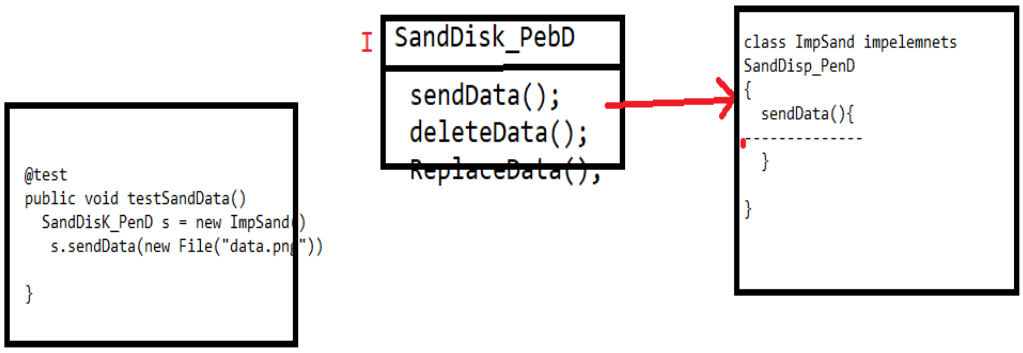
- Testing the business logic of the source code using another program is called unit testing
- White Box testing is also called as unit testing
- In order to automate unit test case, we have to go for Unit testing tools like Junit / TestNG
- Few java developers say Unit testing is also part of API Unit test

EG: Antivirus application, device drivers, JDBC, WebDriver

API [Application programing interface]

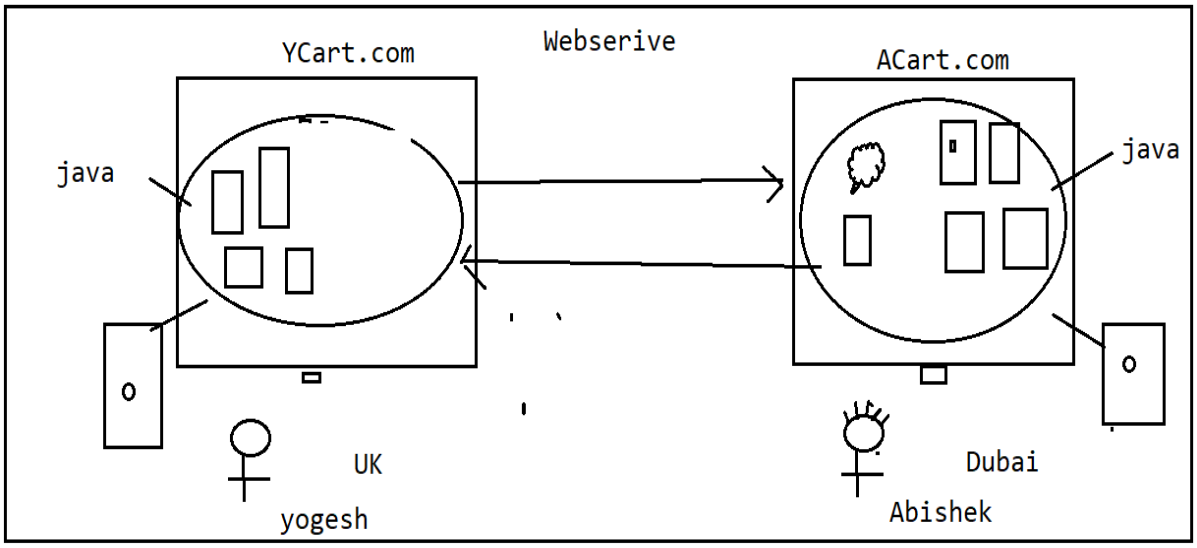
EG :

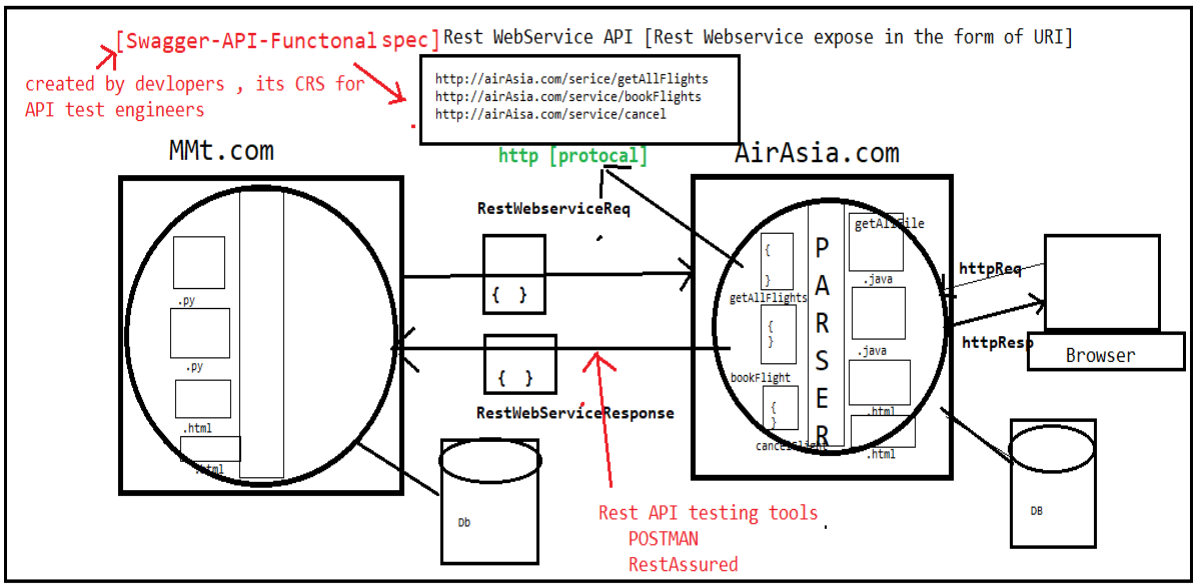
Pen Drive [its hardware storage device , its contain device driver software which used to store & retrive the data]

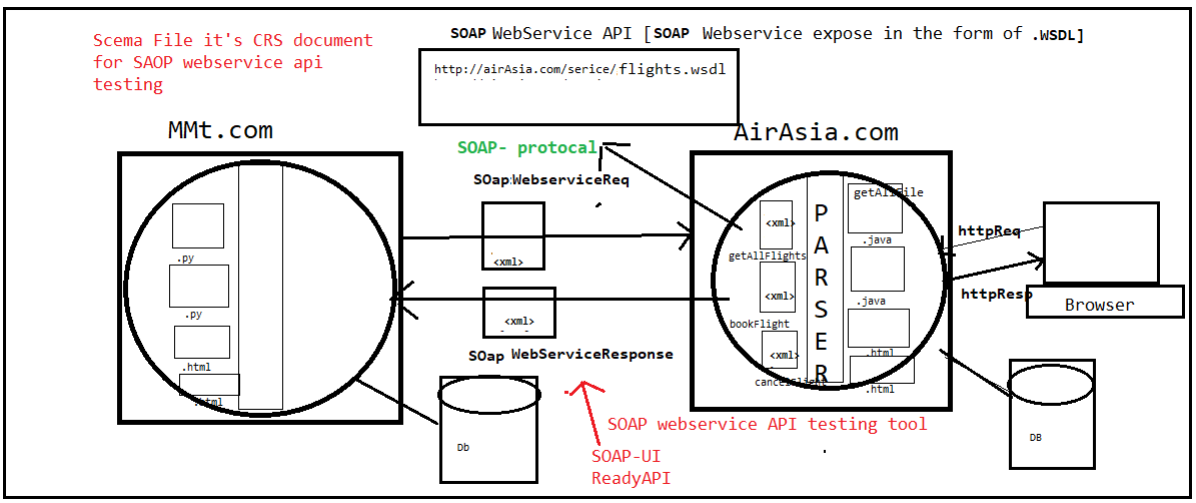


Web service

- Web Services is the mechanism or the medium of communication through which two applications / machines will exchange the data irrespective of their underline technology
- Web services help 2 application exchange information with each other when application running in same or different platform.
- Any service available on web is called webservice
- All Web service are API, but API are not webservice
- WebService help us to share the functionality of one application to any other application without sharing the source code & database data, even though both applications running in same or different platform.


YCart.com
Webserive
ACart.com
java
java
UK
yogesh
Dubai
Abishek


[Swagger-API-Functonal spec]
Rest WebService API [Rest Webservice expose in the form of URI]
created by devlopers , its CRS for API test engineers
http://airAsia.com/serice/getAllFlights
http://airAsia.com/service/bookFlights
http://airAisa.com/service/cancel
MMt.com
http [protocal]
AirAsia.com
RestWebserviceReq
RestWebServiceResponse
PARSER
getAllFlights
bookFlight
httpReq
httpResp
Browser
Db
DB
Rest API testing tools
POSTMAN
RestAssured


Scema File it's CRS document for SAOP webservice api testing
SOAP WebService API [SOAP Webservice expose in the form of .WSDL]
http://airAsia.com/serice/flights.wsdl
MMt.com
SOAP- protocal
AirAsia.com
SOap WebserviceReq
SOap WebServiceResponse
PARSER
getAllFlights
bookFlight
httpReq
httpResp
Browser
Db
DB
SOAP webservice API testing tool
SOAP-UI
ReadyAPI

Why WebService Testing

- The purpose of *Web Service Testing* is to verify that all of the APIs exposed by your application working as expected or not? WRT functionality & performance, security
- All web service is exposed via API
- Testing request and response of the API is called webservice testing
- Webservice provider has to test all the API's, because to make sure all the functionality which is exposed via web service is working or not?

What is SOAP

- SOAP is a simple XML-based protocol to let applications exchange information over HTTP.
- SOAP it uses XML to exchange information between applications.
- SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.
- SOAP stands for Simple Object Access Protocol
- SOAP is a communication protocol
- SOAP is for communication between applications
- SOAP communicates via Internet
- SOAP is platform & language independent
- SOAP is based on XML
- SOAP is simple and extensible

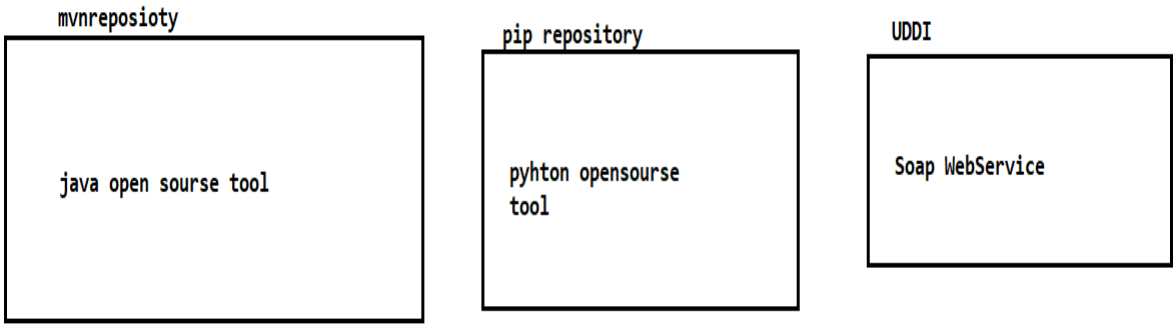
What is WSDL

- An WSDL document describes a soap web service. It specifies the location of the service, and the methods of the service, using these major elements:

Element	Description
<types>	Defines the (XML Schema) data types used by the web service
<message>	Defines the data elements for each operation
<portType>	Describes the operations that can be performed and the messages involved.
<binding>	Defines the protocol and data format for each port type

UDDI

- UDDI is an XML-based standard for describing, publishing, and finding web services.
- UDDI stands for **Universal Description, Discovery, and Integration.**
- UDDI is a specification for a distributed registry of web services.



Soap WebService API Testing

- When two application exchange information via **soap** protocol, which is based on xml, testing those request and response is called Soap webservice testing
- SOAP WebService exposed via .WSDL file
- EG: SOAP API

http://www.dneonline.com/calculator.asmx?WSDL

http://map.google/api/soap/service.wsdl

What is Rest

- REST stands for Representational State Transfer.
- It means when a RESTful API is called, the server will transfer to the client a representation of the state of the request
- REST is an architectural style for developing web services. REST is popular due to its simplicity and the fact that it builds upon existing systems and features of the internet's http resource.
- In order for an API to be RESTful, it has to adhere to 6 constraints:

Why JSON is Popular in RestFull Webservice

- Java Script Object Notation
- It's a Programing language to exchange information between 2 applications
- The JSON format is syntactically identical to the code for creating JavaScript objects.

  Because of this similarity, a JavaScript program can easily convert JSON data into native JavaScript objects.
- JSON is a lightweight format for storing and transporting data
- All Browser & Mobile UI can easily consume json language
- No SQL databases can directly store the data in the form of JSON (Mango DB, Casendra).
- Platform independent.

RestFull WebService API Testing

- When two application exchange information via http protocol, which is based json/xml/text/html, testing those request and response is called Rest webservice testing
- Rest API Webservice are exposed via URI
- EG: Rest API

http://map.google/api/getlocation

http://map.google/api/gettraffice?src='value'& dest=value

http://map.google/api/gettraffic

Difference between SOAP/ Rest

SOAP Web Service	Rest Web Service
SOAP is a protocol	REST is Architectural style
Simple Object Access Protocol	Representational State Transfer
SOAP can't use REST Web services, it is a protocol	REST can use soap as well since it is concept
SOAP expose the services (Business Logic) via .WSDL file	REST expose the service (Business Logic) via URI
SOAP designed using too much standards	REST does not define too much standards
SOAP permits XML data format only	REST permits different data formats- Plain text, html, JSON, XML ,JS
SOAP requires more bandwidth and resources	REST requires less bandwidth and resource than SOAP
SOAP defines its own security	Inherits security measures from underlined transport Protocol (ouath-1.0 outh-2.0 , Bearer token)
SOAP Less preferred than REST	REST more preferred than SOAP

Advantages of Web Service

1. Web Services Interoperability (WS-I)

- Web Services are "Application, Platform and Technology Independent"
- Ex: Uber / OLA and Google Maps shares the data among each other

2. Loosely Coupled

- Each application is independent of one another. Hence changes done to one application will not impact the "unrelated areas"

3. No need of re-inventing the wheel

- Web Services reduces the software development time
- This helps the other business partners to quickly develop application and start doing business
- This helps business to save time and money by cutting development time
- Ex: Uber / OLA can make use of Google Maps

4. Business Opportunity

- Web Services will open the door for new business opportunities by making it easy to connect with partners
- Ex: Dominos can get the order from Food Panda / Swiggy along with getting orders from its own site

5. Service Reuse

- Web Services takes code reuse a step further
- Ex: An organization can have a "Single Payment Gateway service" which helps other web applications of that organization to interact

Pre requist for RMG yantra Application :

1. MySql database

   password should be root

2. Create a database with a name projects

syntax: create database projects;

Running RMG yantra application:

open cmd> java -jar <rmgyantra.jar> (--spring.datasource.password=)

Port already used:-

java -Dserver.port=8085 -jar <rmgyantra.jar> (--spring.datasource.password=password)

username: rmgyantra

password: rmgy@9999

WebUrl:

protocol://servername:portNumber/resourcepath?query=value

http://localhost:8084/

http://localhost:8084/swagger-ui.html#/project-controller

HTTP methods:

GET:- Used to retrieve the information from the service

POST:- Used to create a new resource inside a service

PUT:- Update the existing resource completely

PATCH:- Update the existing resource partially

DELETE:- Delete the resource inside service

CRUD operation:

C-creation R-Retrieve U-Update D- delete

Headers: -

POST

http://localhost:8084/addProject

ContentType: JSON

Body/Payload:


```
{
  "createdBy": "string",
  "createdOn": "string",
  "projectId": "string",
  "projectName": "string",
  "status": "string",
  "teamSize": 0
}
```

GET

http://localhost:8084/projects

==========

Difference between put and patch

PUT: send complete payload/body doing updation

PATCH: sending partial payload or body for updation

Delete Project

http://localhost:8084/projects/{project_id}

Environment variable: -

Instead of Hardcoding common data like URL, We can store it as Environment variable

and we can change it one time if any change required.

How do u call environment variable in request

syntax:

{{variableName}}

Eg:{{url}}

Global variable: -

These variable can be accessed for any collection in workspace

syntax:

{{variableName}}

==========

Different ways to create post request

1. Body->raw-> added body

Type->Json

2. sending a json file

body-> binary-> select file

==========

Parameters:-

1. Path parameter:- In order to fetch particular resource inside a service or parameterizing the path

{pathParameter}

2. Query parameter:-Query parameter is used to sort or filter the response

syntax: ?query=value

what is the difference between uri and url

uri: uniform resource identifier (localhost:8084) name

url: uniform resource location (localhost:8084/projects) address

Authentication

1. Basic Auth:- username and password

2. Barrier Token:-

scope:-read and write

3. Oauth 1.0:-

4. Oauth 2.0:-

client id

client secrete

grant type

scope

auth url

token url

==========

github api

base uri:-https://api.github.com

barrier token: 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

==========

Oauth 2.0

client ID:- 12d1468e9feab7eca9c6

client secrete:- c601df3de8105f875e86579cd38321881a7a52c1

redirect url: https://github.com

access token:

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

gho_j4POPAvcSzK937Sp6k0bUB2LkzBVPI0e

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

==========

validations:

To verify that the response is as per our exception

1.status code

2. cookies

3. contentType

4. response Time

5. response body

PostMan snippets:

1. status code: status code is 200: - verifying status code

2. Response header: contenttype header check

3. Response Time less than 200ms

4. Response body json value check

Request chaining

Dependency between 2 API request

Calling one request to another request

Steps:

1.Response json value we have store in environment variable or global variable

2. Calling the environment or global variable in another request

Parameterization/Data driven: -

Reading the data from the external resource

Note: - PostMan support only csv and json

==========

Running a collection in batch in Postman

-> run collection -> click on Run Collection Name

==========

Running the collection in command line using newman

Pre-request: Node js should be installed

To check node js installed in pc

cmd> node -v

install newman= npm install -g newman

1. Export entire collection in json format

    -> collection->export-> give extension as .json

2. Export environment variable in json format

    -> environment variable -> edit-> export-> extension .json

3. In command promt

newman run <collection.json> -e <environment.json>

PostMan –Response- Validation

➔In order the validate response in postMan, we have to go for snippets

➔Assert / Snippets(tests): Assertion is feature available in all postMan tool which is used to validate expected result of the API response or API test Case

Assertion available in postMan

1. status code is 200: used to verify the status Code the get request

2. status code for Post Request 201: used to verify the status Code the Post request

3. status code name has String: used to verify the status Code String
   EG: 200 Successful
   201 Created
   204 No Content
   422 Unprocessable Entity
4. Response Body Contains String: used to verify the expected data at least available in response body

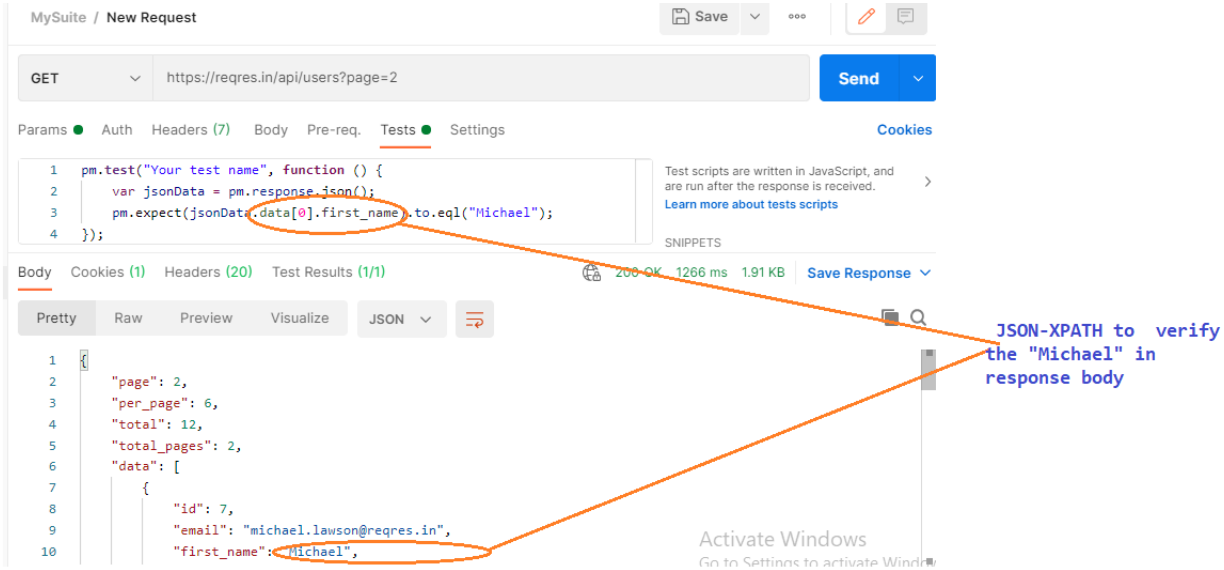
5. Response body Json value Check: used to navigate to specific location using JSONXpath & verify the expected data available or not?

6. Response time is less than 200: used to the time taken between the request & response

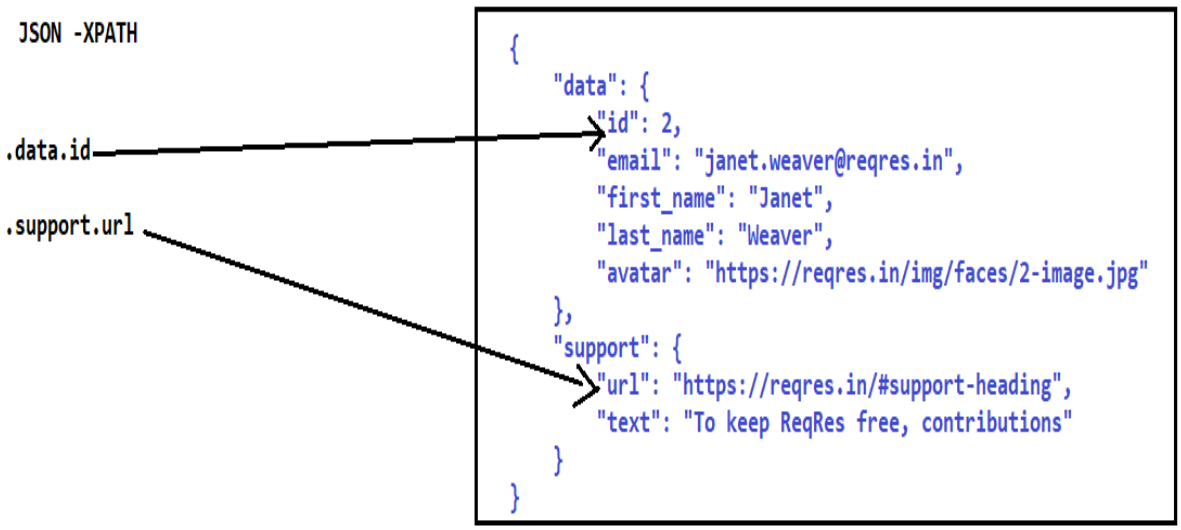
7. Send Request: used to perform request chaining

How to verify the Complex Response?

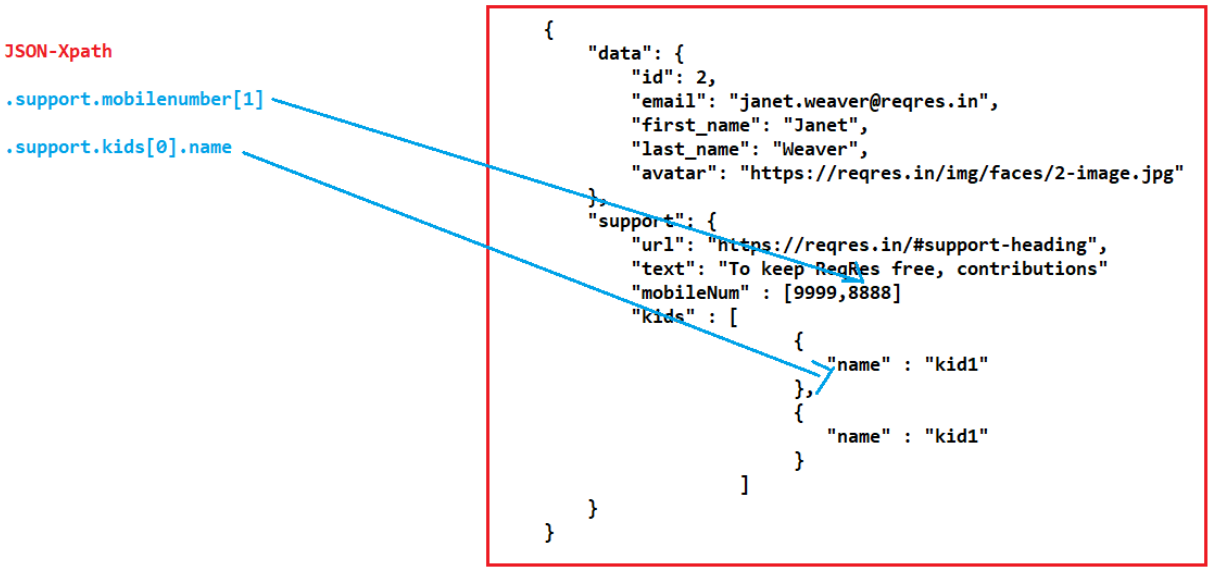
In order to verify the complex response, we have to write “JSONPATH” to navigate to specific data in response body then verify the expected result using “JASON value check” assert



EG 2:



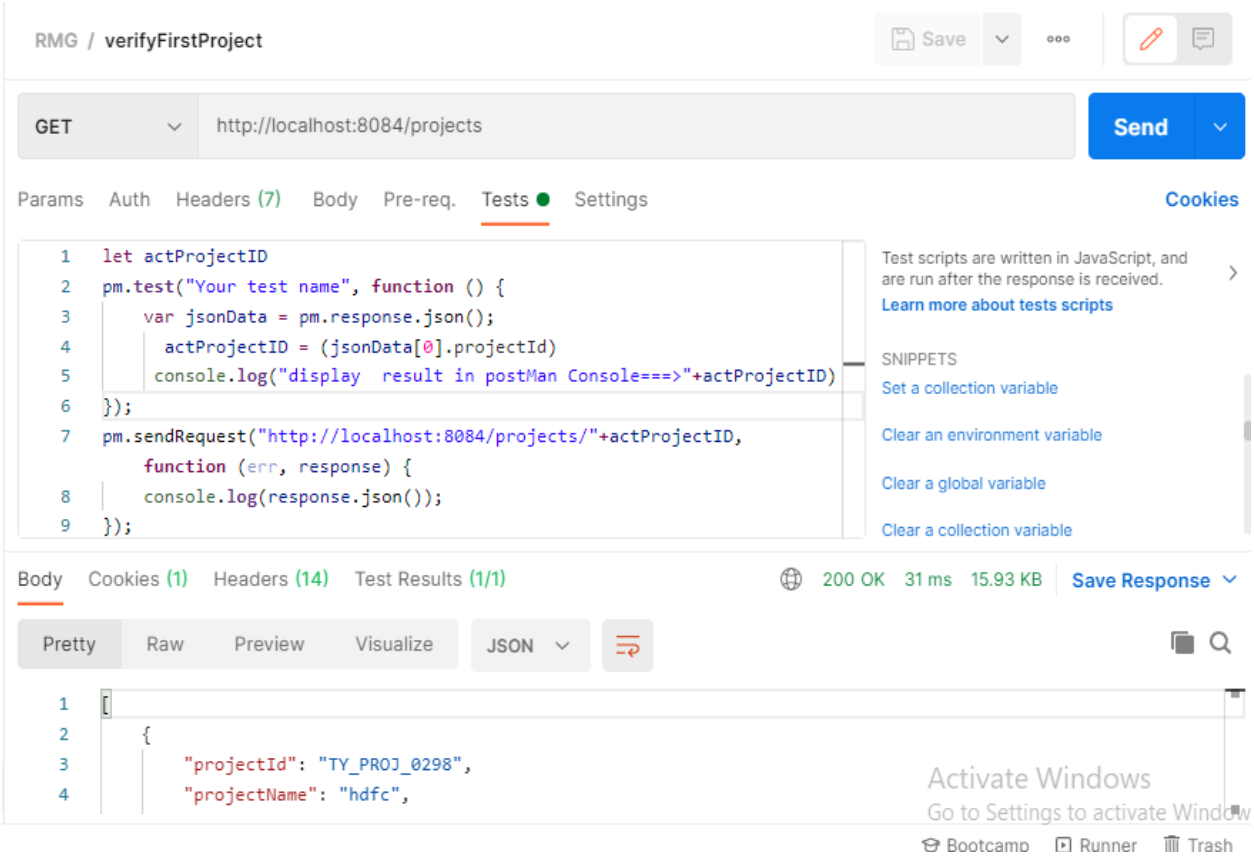
EG 3:



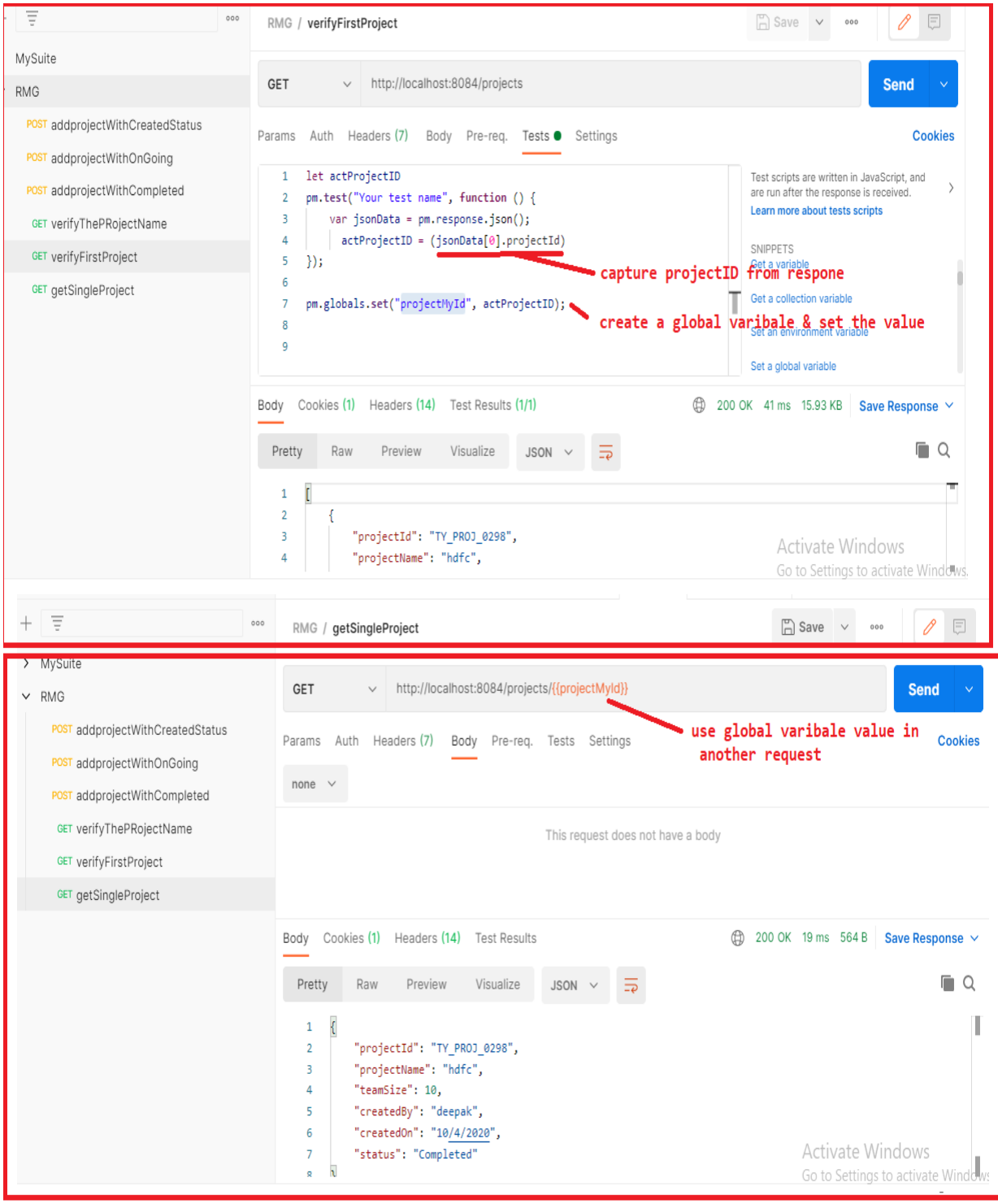
What is Request chaining or Properties transfer or API Chaining?

How to Automate End-2 End Scenario in API testing?

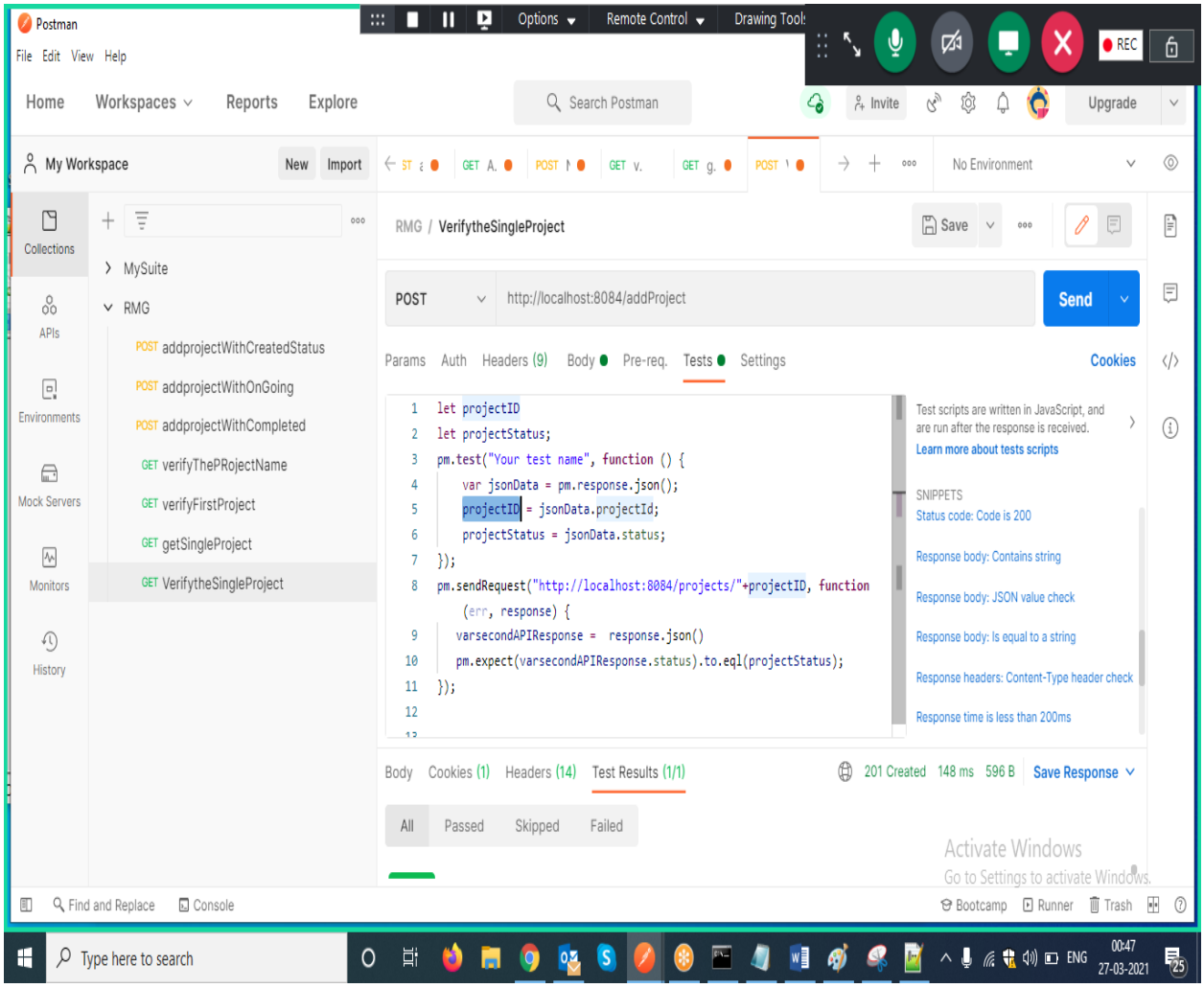
Getting the data from one response API & pass same data to another request API is called Request Chaining



EG : 2

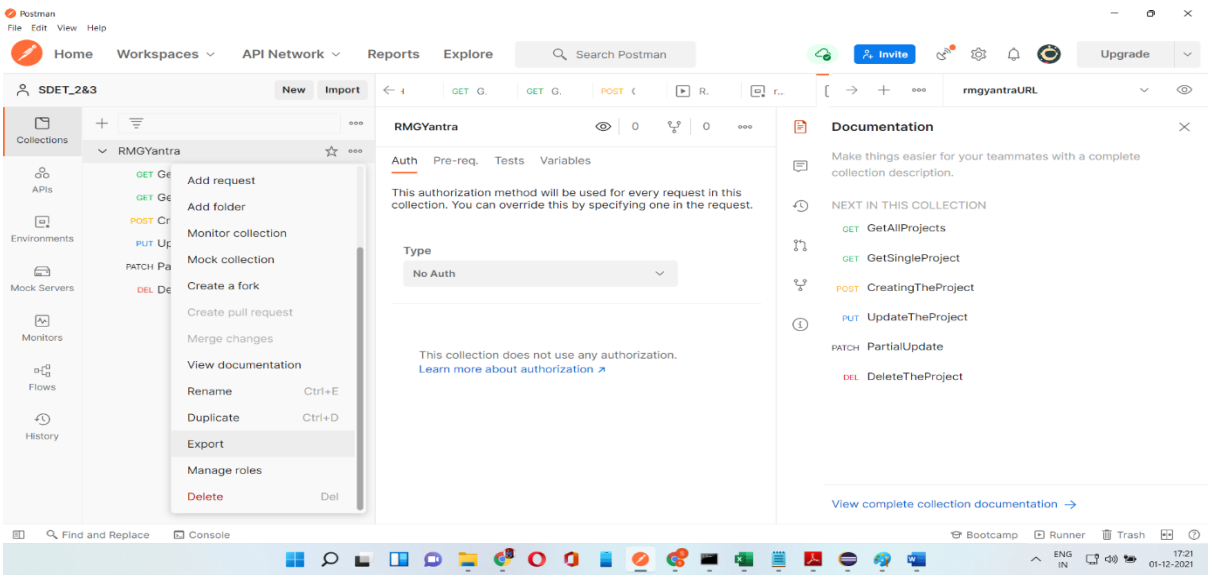

MySuite
RMG
POST addprojectWithCreatedStatus
POST addprojectWithOnGoing
POST addprojectWithCompleted
GET verifyThePRojectName
GET verifyFirstProject
GET getSingleProject
RMG / verifyFirstProject
Save
GET
http://localhost:8084/projects
Send
Params Auth Headers (7) Body Pre-req. Tests Settings
Cookies
let actProjectID
pm.test("Your test name", function () {
var jsonData = pm.response.json();
actProjectID = (jsonData[0].projectId)
});
pm.globals.set("projectMyId", actProjectID);
capture projectID from respone
create a global varibale & set the value
Test scripts are written in JavaScript, and are run after the response is received.
Learn more about tests scripts
SNIPPETS
Get a variable
Get a collection variable
Set an environment variable
Set a global variable
Body Cookies (1) Headers (14) Test Results (1/1)
200 OK 41 ms 15.93 KB
Save Response
Pretty Raw Preview Visualize JSON
[
{
"projectId": "TY_PROJ_0298",
"projectName": "hdfc",
Activate Windows
Go to Settings to activate Windows.
RMG / getSingleProject
Save
MySuite
RMG
POST addprojectWithCreatedStatus
POST addprojectWithOnGoing
POST addprojectWithCompleted
GET verifyThePRojectName
GET verifyFirstProject
GET getSingleProject
GET
http://localhost:8084/projects/{{projectMyId}}
Send
use global varibale value in another request
Params Auth Headers (7) Body Pre-req. Tests Settings
Cookies
none
This request does not have a body
Body Cookies (1) Headers (14) Test Results
200 OK 19 ms 564 B
Save Response
Pretty Raw Preview Visualize JSON
{
"projectId": "TY_PROJ_0298",
"projectName": "hdfc",
"teamSize": 10,
"createdBy": "deepak",
"createdOn": "10/4/2020",
"status": "Completed"
Activate Windows
Go to Settings to activate Windows

EG :3

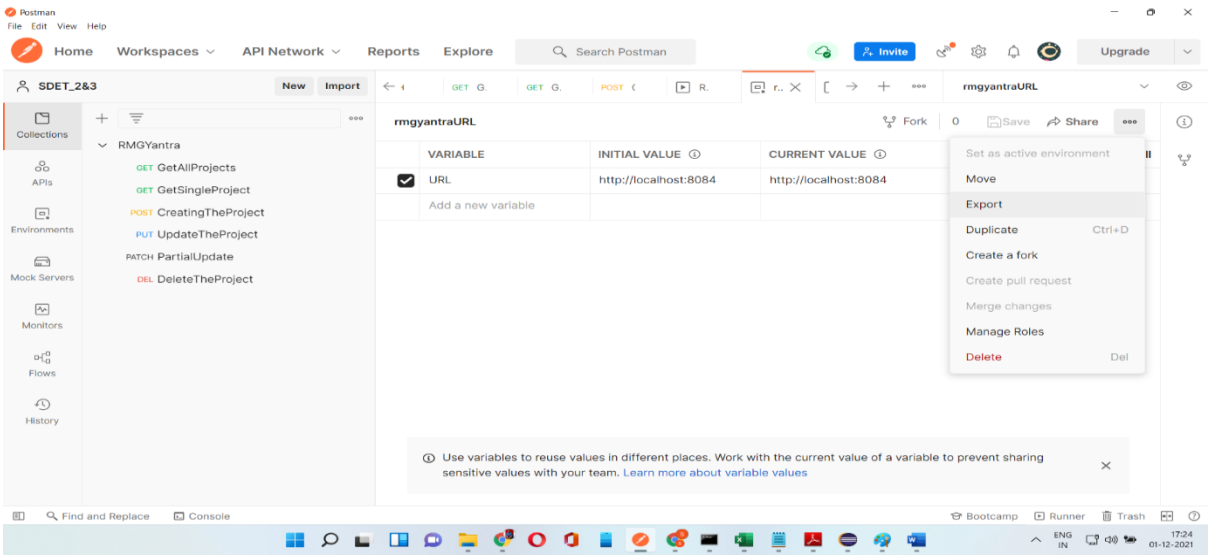


How To Run The Collection In Command Prompt

- Install Node JS → Search on Google “Node JS Download”.
- Download the .msi file.
- Install Node JS.
- Open Command Prompt →npm -v & node -v
- Install New Man → npm install -g newman
- Export The collection From Postman



- Save The file with .json
- If Environment Variable is there then Export those with .json



- Type the command in command prompt
  newman run <collection Path> -e <Environment Variable Path>

Request Authentication/ Authorization

Authentication: gives those users to access a resource via Rest / SOAP API

Authorization: gives role /permission for the resource

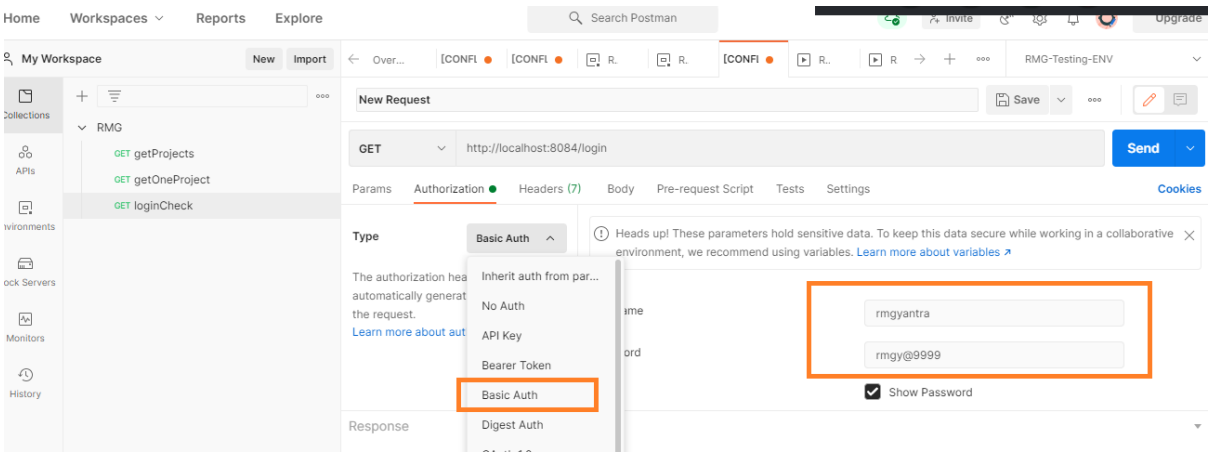
- Almost every REST API must have authorization to check whether you are authorized client or not? before allowing access to the resource
- Authorization is present in header of the request.

There are 4 types of authorisations:

1. **Bearer token**: Required "TokenID" to access an API
2. **Basic auth**: Required "Username & Password "to access an API
3. **OAuth 1.0**: Two level authentication
4. **OAuth 2.0**: One level authentication

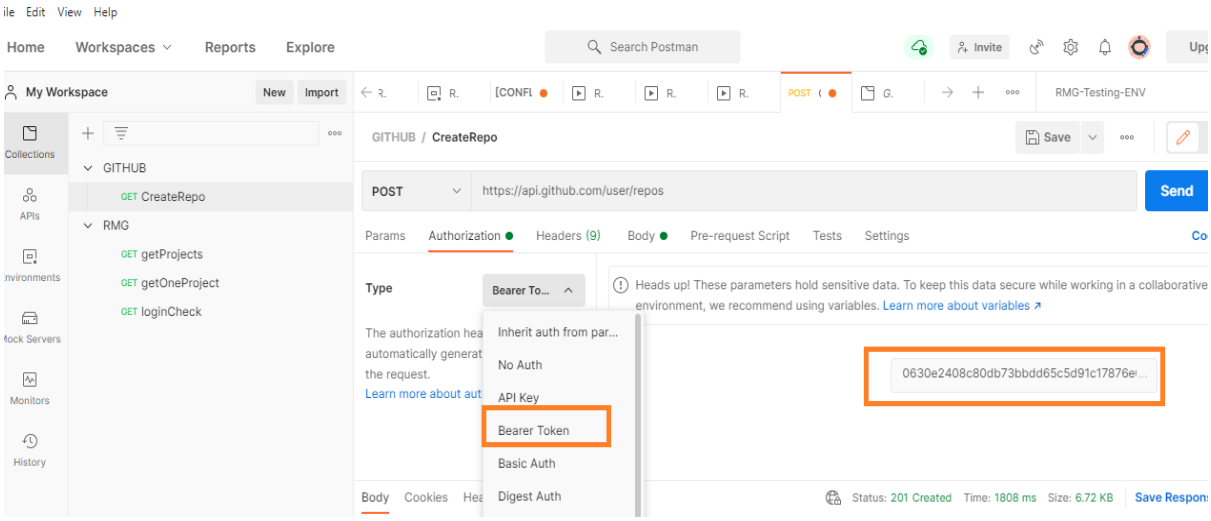
OAuth: is an Open Standard for Authorization protocol, widely used in web application, which allows API services to accesses user data without sharing is password.

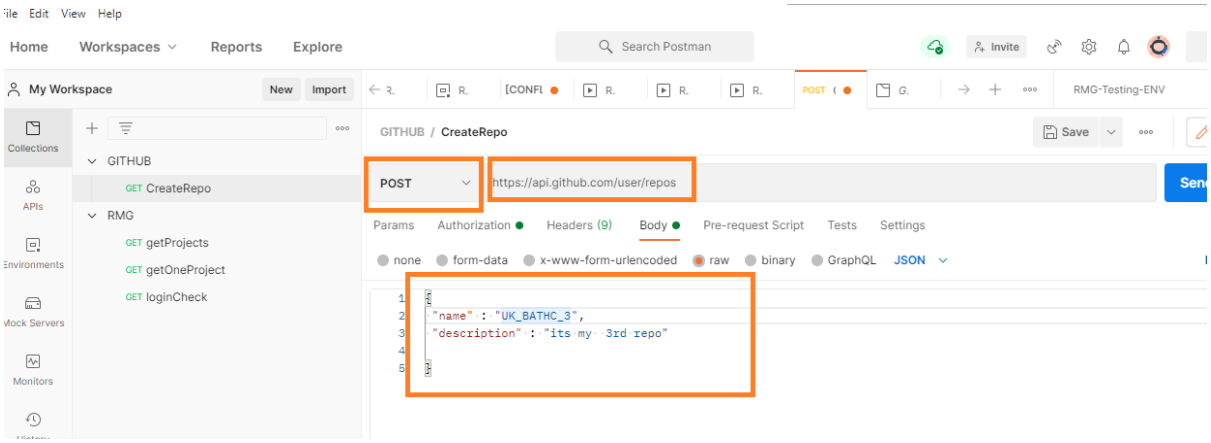
Basic Auth: used to access the resource (API) via username / password is called basic auth



Disadvantages: sharing the username /password is not advisable via API, some body can hack your Data

Bearer Token: used to access the resource via bearer token ID, that is unique Id to access all the API resource.

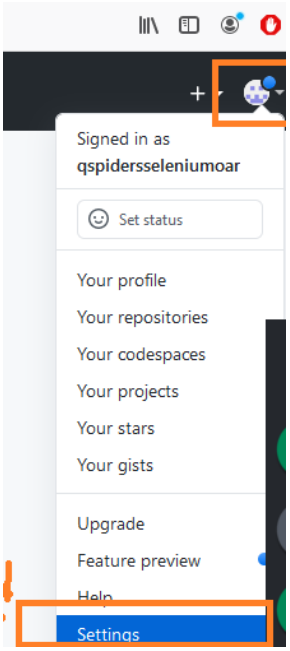




Disadvantages: bearer can also share to other team members, & bearer token id will be fixed until developer changes.

How to create Bearer Token in GITHUB?

1. Go to gitHUB URL http://github.com
2. Create an Account with GitHUB
3. Login to Git HUB
4. Go to user Menu ➔ click Settings



5.click on “Developers settings”
6. Click on “personal Access token”
7. Click on “Generate new Token” button
8. Enter the name & select “Repo” CheckBox
9. Click on “Generate a Token”
10. Copy the Token ID

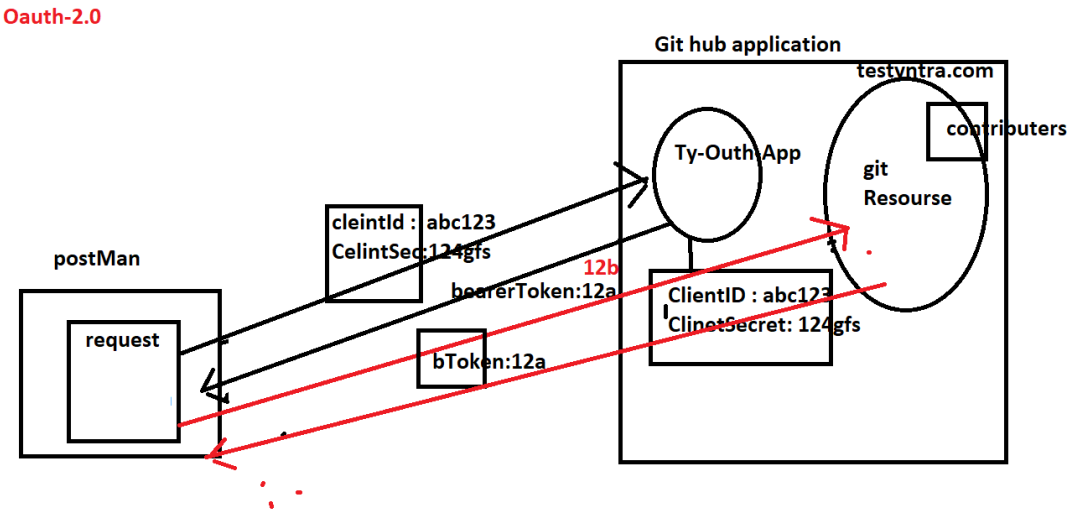
How to get API for create Repository in GITHUB

1. Go to GIT API Functional Spec Document
   https://docs.github.com/en
2. Copy the Base URI
   https://docs.github.com/en/rest/overview/resources-in-the-rest-api
3. Copy the End PONT & get the JSON BODY structure
   https://docs.github.com/en/rest/reference/repos#create-a-repository-for-the-authenticated-user

OAUTH-2:

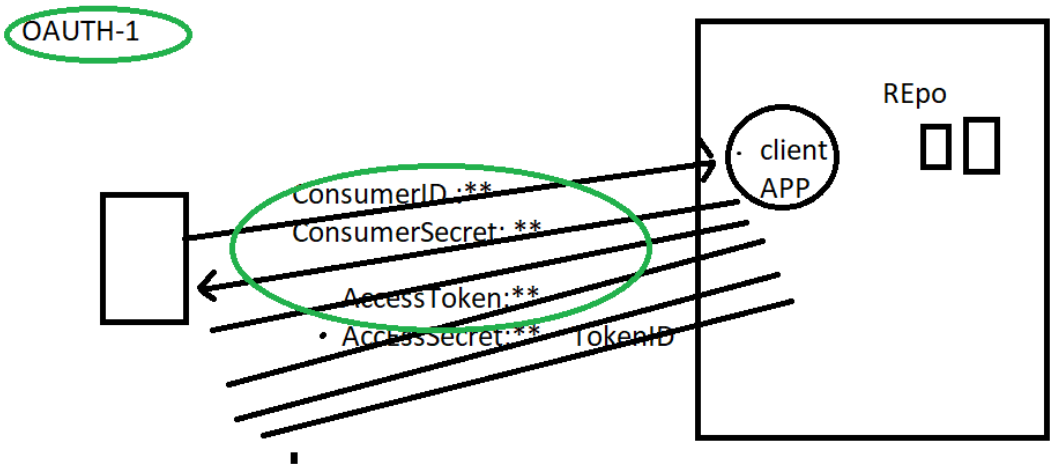
- Whenever your application requests private user data, it must send an OAuth 2.0 token along with the request. Your application first sends a client ID and, client secret to obtain a token. You can generate OAuth 2.0 credentials for web applications, service accounts, or installed applications.

- One level of Authentication
- Client application includes “client secret” with every request.



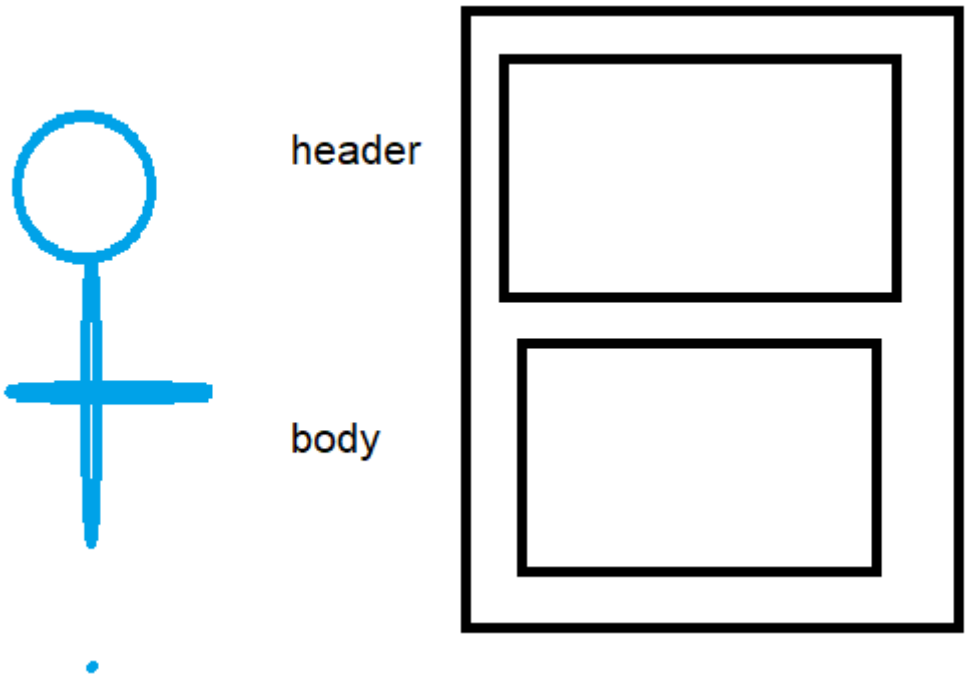
OAUTH-1 :

- **Consumer key and secret & Access Token / Access Secret**: These two strings are used to identify and authenticate the client application
- **Two level of Authentication**

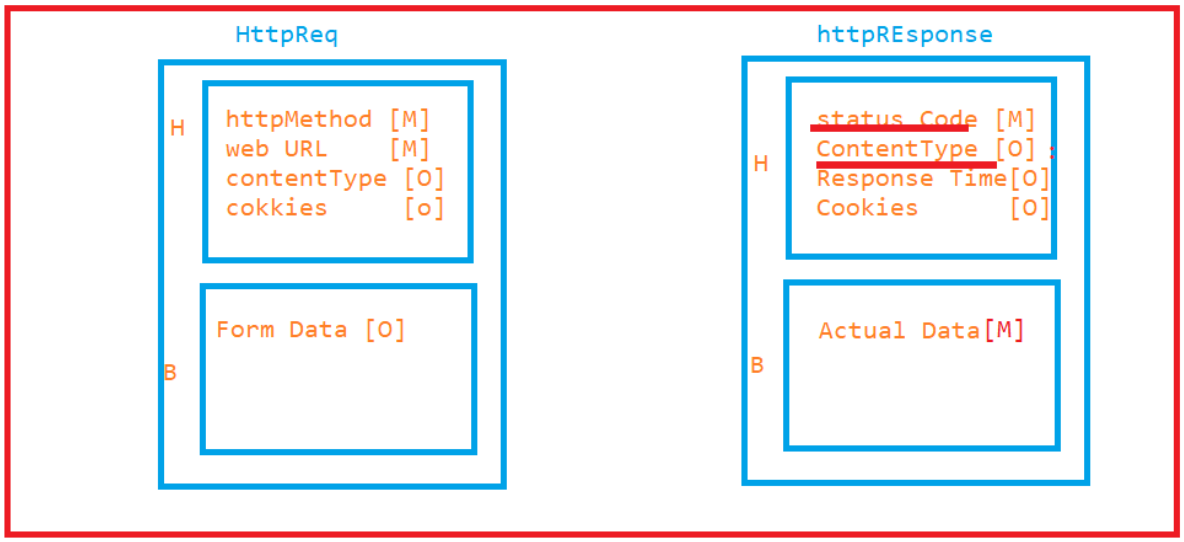


Http Structure

- Http protocol is also a structure and it consists of header & body part separated by empty line.
- In order the view the http request & response structure we go for Fiddler tool (developer debug tool).



Key Elements of Http-request & Http-response



Http method:

- It is a mandatory information present in the header of the http Request
- http method is the first element in the http request, used to specify the type of the request which is sent by client(Browser) to Server(Application)
- http-1.0 has 3 methods, but in http-1.1 five more methods got introduced

➢ Get → get resource from the server

➢ Post → create resource inside the server

➢ Put → complete update resource inside the server

➢ Patch → partial update the resource inside the server

➢ Delete → delete the existing resource inside the server

➢ Trace

➢ Option

➢ Head

➢ Connect

Note: First http methods play major role in WebService testing

WebURL:

- Mandatory information in header of the http request
- It's used to identify the specific web Resource inside the web application
- In get : Query will be present in URL
- In Post : Query will be present in the Body of the Request

Syntax

protocal://<domainName>:port/resourcePath?queryString#fragementId

EG :

http://localhost:8888/index.php

Status Code	Description
100	Continue
200	Server successfully handled the request
300	Re direction request
400	Client side error(requested resource not found at server side)
500	Server side error(Server encounter an unexpected condition)

http://google.com/search?q=iphone11

http://172.217.160.142/search?q=iphone11

Form data :

- Data collected using htmlForm is called form data **or** data collected from browser to server
- Size of the form will vary form to form
- Get -> will not have form data
- Post-> will have form data

Status Code

- It is a mandatory information present in header of the response.
- Status code represent the status of the http request

EG

Content type:

- Its an optional information present in the response body
- Used the specify the content of the data present in the response body
- Every file will have dedicated contentType (MimeType)

EG :

application/json : .json

application/xml : .xml

application/zip : .zip

image/png :.png

……

Actual data:

- It's an optional information present in the response body
- Actual data responded by server, if in case request is successful. otherwise server provides error message in the body of the request

Fiddler Tool [API debugging tool]

- ***The Fiddler tool helps you debug web applications by capturing network traffic between the Internet and test computers. The tool enables you to inspect incoming and outgoing data to monitor and modify requests and responses before the browser receives them.***
- ***It is a debugging tool where we can view http request and response data/property.***

Download from

https://www.telerik.com/download/fiddler/fiddler4

Mangerial_Interview_Questions_On_API_Testing

1. What is your Backend Roles & Responsibility?
    a. When we get API-Functional-spec[Swagger-Document] document from the development team, we go through all the End-points understand the Functionality.
    b. Involved in Understanding JSON schema [JSON Structure & Mandatory & non mandatory fields] for all endpoints.
    c. Write positive & Negative API Test scenario for every End points.
    d. Execute them manually using Postman.
    e. Get the latest Rest Assured framework from the GitHub.
    f. Involved in CRUD Operation.
    g. Involved in performing Complex post operation using POJO Class/JSON File.
    h. Involved in HTTP Request creation.
    i. Involved in HTTP Response Validation.
    j. Involved in Complex response validation using JSON Path.
    k. Good Knowledge on All type of Authentication.
    l. Used all type of Parameter in API.
    m. Involved in E-2-E workflow Automation using Request Chaining.
    n. Used BDD to perform CRUD Operation.
    o. Good Knowledge of Status Code.

2. Can you write few API URL, which u used in your Previous Project?

Add Course

POST http://skillrary.com:7078/addCourse

GET(All Course) http://skillrary.com:7078/courses

GET http:skillrary.com:7078/courses/{Appium}

PUT http://skillrary.com:7078/ courses/{API}

PATCH http://skillrary.com:7078/courses/{Selenium}

DELETE http://skillrary.com:7078/courses/{Selenium}

3. How Many API Test cases you wrote in your previous Project?

Module-1→ 6 API's → 140 Test Cases

Module-1 + Module-2 → 250 Test Cases

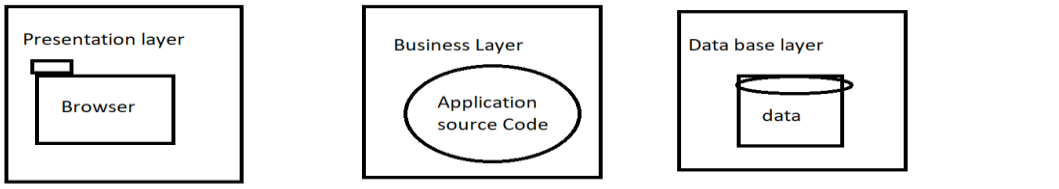
4. Can you write one POST Request body in your Previous Project?

5. Can you write one complex GET Response in your Previous Project?

6. What was not Automatable in your Previous API Testing?
   a. Adhoc testing Test cases
   b. Compatibility Testing
   c. Usability Testing

PostMan_API_Interview_Questions

1. What technology your application was developed?



Front End Technology
- html-5 + javascript
- anguler-js
- reactjs

Backend Technology
- servler+ java
- Jsp+java
- SpringBoot+ java

database Technology
- SQLDB ==> mysql, orcale10g, db2, sybase, mssql
- NoSQlDb ==> mango db , cassandra, redis

2. Difference between Unit testing & API testing

Unit Testing	API testing(web service)
Done by development team	Done by testing team or SDET team
Should have knowledge on source code	Should have knowledge on expectations and agreements of API (Api-Functional Spec)
Test each and every source code of the application	Test the data flow between 2 application
It is a white box testing	It is a grey box testing
Tools: eclipse+testNG	Tools: SOAP UI, POSTMAN, RESTCLIENT, RESTASSURED

3. Difference between JSON & XML language

JSON	XML
JSON object has a type	XML data is type less
JSON types: string, number, array, Boolean, null, Object, arrayofObject	All XML data should be string
Data is readily accessible as JSON objects	XML data needs to be parsed.
JSON is supported by most browsers.	Cross-browser XML parsing can be tricky
Retrieving value is easy	Retrieving value is difficult
A fully automated way of DE serializing/serializing JavaScript.	Developers have to write JavaScript code to serialize/de-serialize from XML
Native support for object.	The object has to be express by conventions - mostly missed use of attributes and elements.
It supports only UTF-8 encoding.	It supports various encoding.
Structure is simple	Structure is complex
It doesn't support comments.	It supports comments.
JSON files are easy to read as compared to XML.	XML documents are relatively more difficult to read and interpret.

4. Difference between SAOP & Rest Webservice

SOAP Web Service	Rest Web Service
SOAP is a protocol	REST is Architectural style
Simple Object Access Protocol	Representational State Transfer
SOAP can't use REST Web services, it is a protocol	REST can use xml as well Web Services since it is concept
SOAP expose the services (Business Logic) via .WSDL file	REST expose the service (Business Logic) via URI
SOAP designed using too much standards	REST does not define too much standards
SOAP permits XML data format only	REST permits different data formats-
	Plain text, html, JSON, etc
SOAP requires more bandwidth and resources	REST requires less bandwidth and resource than SOAP
SOAP defines its own security	Inherits security measures from underlined transport Protocol (ouath-1.0 outh-2.0 , Bearer token)
Less preferred than REST	REST more preferred than SOAP

5. Difference between Frontend & Backend language

Frontend testing	Backend testing
Front-end is always performed on GUI	Back-end testing is done on Business layer and should have knowledge on Data base and business logic of software
Testing becomes slower coz it has to wait for browser rendering time	It is faster and saves time.
Performance testing Is not easy in front-end	Performance testing is easy and faster
Adoc, usability and compatibility testing are possible	Adoc, usability and compatibility testing are not possible
Manual & selenium automation will be done at front end(BBT)	Whitebox , Soap API & rest API testing will be done in the backEnd(WBT. GBT)
End to end testing is possible	End to end testing is not possible
Tools: selenium, QTP, test complete etc	SOAP UI, POSTMAN, RESTCLIENT, RESTASSURED, testNG , JUNIT

6. Which APi testing Tool you have used?

PostMan & RestAssured , RestClient , testOptimize , read API

7. Which type of Webservice testing you worked on

Rest WebServciec testing

8. Types of API testing ?

A. Soap WebServcie API testing
B. REST WebServcie API testing

9. What is API ?

Application programing Interface, testing interface between two applications, or
Testing the application in business layer without browser is called API testing

EG : Rest webservcie is API testing
Soap WebServcie is API testing

10. What is the advantages of the webservice API testing?

a. API testing very Faster (because no to wait for Browser rendering time)

b. testing the functionality without GUI(Browser)

c. testing the functionality early stages (We can start API testing in Sprint-1)

d. find defect in early stages

e. Time effective & fast to release

f. whenever API provider develop an API , every api should be tested the Functionality, performance security, reliability before exposing those API to consumer

6. what is the disadvantages API testing?

a. can't perform negative testing

b. End to end workflow testing is difficult

c. testing should have knowledge on Programing & data base (Backend)

What is Unit testing ?

Testing the source of the application using another program is called unit API testing

11. Why JSON is very Popular?

a. Java Script Object Notation
b. It's a Programing language to exchange information between 2 applications
c. The JSON format is syntactically identical to the code for creating JavaScript objects.
d. Because of this similarity, a JavaScript program can easily convert JSON data into native JavaScript objects.
e. JSON is a lightweight format for storing and transporting data
f. JSON is often used when data is sent from a server to a web page
g. JSON is "self-describing" and easy to understand
h. All Browser & Mobile UI can easily consume json language
i. NO SQL databases can directly store the data in the form of JSON (MangoDb, Casendra)
j. Platform independent

12. What is Soap webservice testing

- When two application exchange information via **soap** protocol, which is based on xml based, testing those request and response is called Soap webservice testing
- SOAP Webservice exposed via .WSDL file
- EG : SOAP API

http://www.dneonline.com/calculator.asmx?WSDL
http://map.google/api/soap/service.wsdl

13. What is Rest Webservcie testing ?

- When two application exchange information via **http** protocol , which is based json/xml/text/html/js , testing those request and response is called Rest webservcie testing
- Rest API Webserice are exposed via URI

- EG : Rest API

  http://map.google/api/getlocation

  http://map.google/api/gettraffice?src='value'& dest=value

  http://map.google/api/gettraffic

14. How many years of Experience in API project?

    2 years

15. Where did you used API in your Previous project & Purpose of the API?

    Product, contacts, Quote, Case

16. How many you have tested

    20 RestFull API

17. Can you Write or list Out few API in your Previous Project

Product

Get Http://autodesk-test-env.com:8084/products

Post Http://autodesk-test-env.com:8084/products/addproduct

Put patch ,delete Http://autodesk-test-env.com:8084/product/{productID}

Get Http://autodesk-test-env.com:8084/products?name =hplabtaop

18. Write on POST API request in your previous Project

    EG:

    Post Http://autodesk-test-env.com:8084/products/addproduct

```
{
"ProductName" : "name"
"SerailNum" : 123
"Manfauting date" : ""
"unitPrice"   : 2000INR
"Qty" : 10

}
```

19. Write one API response in your previous Project


```
{
 "name": "name",
 "msg": "successfully added",
 "id": "625",
 "createdAt": "2020-11-02T08:05:40.255Z"
}
```


EG : get all Products

Get Http://autodesk-test-env.com:8084/products

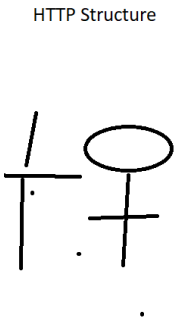

```
{
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "product": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/follettkyle/128.jpg"
    },
    {
      "id": 8,
      "email": "lindsay.ferguson@reqres.in",
      "first_name": "Lindsay",
      "last_name": "Ferguson",
      "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/araa3185/128.jpg"
    },
    {
      "id": 9,
      "email": "tobias.funke@reqres.in",
      "first_name": "Tobias",
```

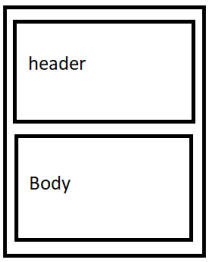
```
            "last_name": "Funke",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/vivekprvr/128.jpg"
        },
        {
            "id": 10,
            "email": "byron.fields@reqres.in",
            "first_name": "Byron",
            "last_name": "Fields",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/russoedu/128.jpg"
        },
        {
            "id": 11,
            "email": "george.edwards@reqres.in",
            "first_name": "George",
            "last_name": "Edwards",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/mrmoiree/128.jpg"
        },
        {
            "id": 12,
            "email": "rachel.howell@reqres.in",
            "first_name": "Rachel",
            "last_name": "Howell",
            "avatar":
"https://s3.amazonaws.com/uifaces/faces/twitter/hebertialmeida/128.jpg"
        }
    ],
    "ad": {
        "company": "StatusCode Weekly",
        "url": "http://statuscode.org/",
        "text": "A weekly newsletter focusing on software development, infrastructure, the
server, performance, and the stack end of things."
    }
}
```

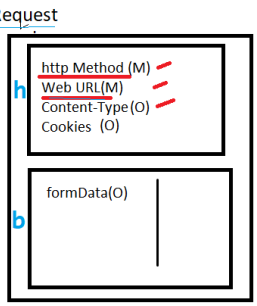

20. How many API test case you written

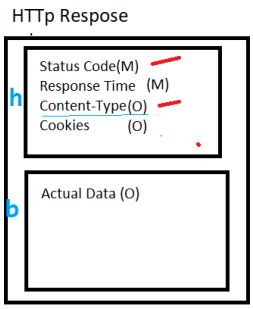
20 API ➔ 100 tc

21. Explain httpRequest / httpResponse structure









http method : its a mandatory information present in the header of the request which is used specify the type of the request clinet is sending to server

1. Get (used to read information/resource from the Server) — C
2. POST (used to create resourse inside the Server) — R
3. PUT (used to completly update the resouce inside the server) — U
4. Patch (used to partail update the resouce inside the server) — U
5. Delete (used to delete the existing resouse avivale inside the server) — D
6. HEAD
7. Connect
8. Option
9. Trace

StatusCode
its a mandatory information present header of the respose , its will tell you the status of the Request

100 : Continuee
200 : Successfull
300 : Redirect
400 : Client Side Error
500 : Server Side Error

Web URL : its is the header of the Request , used to uniquly identify specific WebResourse inside the Server

Syntax :

http://<domainNAme>:portNum/ResourcePath?QueryParemeter

hhtp: // Ty-Deepak-Tr:8080/StudentApp/Login.html?username=admin& password=manager

Base URI — End Point

https://www.google.com/Search?q=jobs&key=hp&key=value

Content-Type:
Its is presnt in the header of the Response
.its Specify the Type of the Content available in body of the Response

MIME TYPE
text/html : html
application/xls : eXcel
application/pdf : PDF
application/xml : XML
application/json : Json
image/png : image

Form Data :
Data collected from clinet to server is called form data , form data will be allways present HTML Request Body
Data collected using htmlForm is called formdata

- Size of the form will vary form to form
- Get -> will not have form data
- Post-> will have form data

Autal Data
1. Its an optional information present in the response body
2. Actual data responded by server, if in case request is successful . other wise server provide error message in the body of the request

HTTP Request:

httpMtd : specify the type of the request

Web URL : uniquely identify the WebResourse inside the Web Application

ContentType: specify the type of content available in body the Request/Response

formData : data collected in html Form / or request body

HTTP Response

Status Code : specify the status of the Request
ContentType
Actaul Data : data available in response body

22. Explain the http method which you used In your previous project

Get = > read resource from the sever
Post => create resource inside the sever
Put=> create resource/ complete update resource inside the sever
Patch => partial update resource inside the sever
Delete => Delete resource inside the sever

23. Explain the status code which you encounter in your previous project

100 Continue

----------Successful----------

200 (OK) : able to read resource from the server
201 (Created) : able to create **resource inside the server**
202 (Accepted) : able to accept the permission request inside the server
204 (No Content) : got response from the server but no content

---------- **Redirected** ----------

300 Redirected : indicates that further action needs to be taken by the user agent in order to fulfill the request. http method is not supported

----------Client-Side Error----------

400 (Bad Request) : The request could not be understood by the server due to malformed syntax
401 (Unauthorized) : request is not authorized to access the resource inside the server
403 (Forbidden) : server understood the request but not authorized person to access
404 (Not Found) : not able to find the resource inside the server which you requested
405 Method Not Allowed

----------**Server Side Error**----------

500 Internal Server Error : problem from server side
502 Bad Gateway : proxy / firewall (network) issue in server side
503 Service Unavailable : The server is currently unable to handle the request
504 Gateway Timeout : server not response even waiting for long time

24.What is Crud operation

Create read update delete

25. What Authorization & Authentication

Authentication to check whether you are authenticated client or not ?
Authorization to check what resource your accesses

26.Challenges you faced in API Testing

a. End to End Scenario testing is challenging because we have to do API Chaining

   EG: Scenario →Search Product + ADD to cart + Billing + Logistic

   API : API -1 for Search Product

   API-2 for ADD to cart

   API-3 for Billing

   API-4 for Logistic

b. API Document is not clear, API testing challenging
c. Negative testing is challenging because in api document will not have complete requirement
d. Validation of Complex response is challenging
e. Deriving api test scenario is challenging ,

27.Prerequisite for APi testing

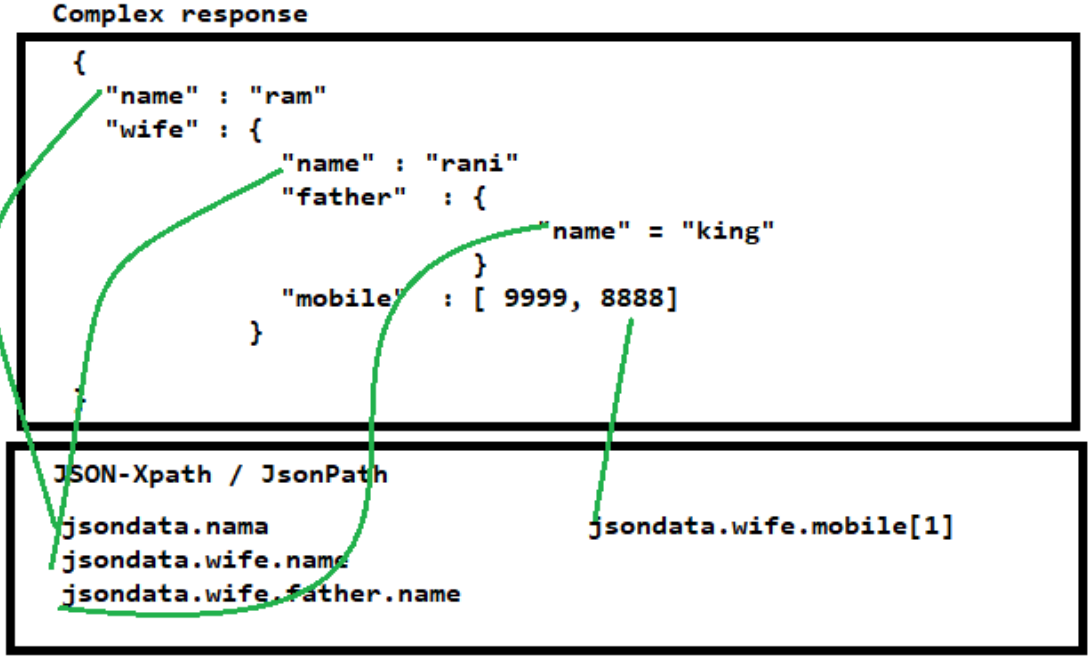
- Knowledge of client server architecture of the application
- Knowledge of HTTP request and HTTP response properties;
- Knowledge of JSON or XML language
- API documentation / API Functional spec (Swagger Document)from Development team
- Get the list of URI (BASEURL + ENDPOINT ➔ URI)
- Knowledge of CRUD
- Write test case for every API that include CRUD operation
- Required tool for API TESTING

  Postman → GUI based API testing tool

  RestAssured → Headless Java libraries for REST API testing
- Collect API authentication if required (like Basic Auth , Bearer Auth , Oauth-1.0, Oauth-2.0

28.How to verify the complex response in API testing?

Will write jsonPath to navigate to specific data inside the response , then will go for "JSon Value checker "Assertions in postMan



29.How many ways we can post a request in POSTMAN

1. Post a request body by JSONOBJECT
2. Post a request body by Key:value pair (hashMAp)
3. Post a request body by JSON File

30.How to post Complex request in postman

We can post json File Itself in postman ➔ body➔binary➔ select File

Note: file extension should be file.json , if content Type in JSON

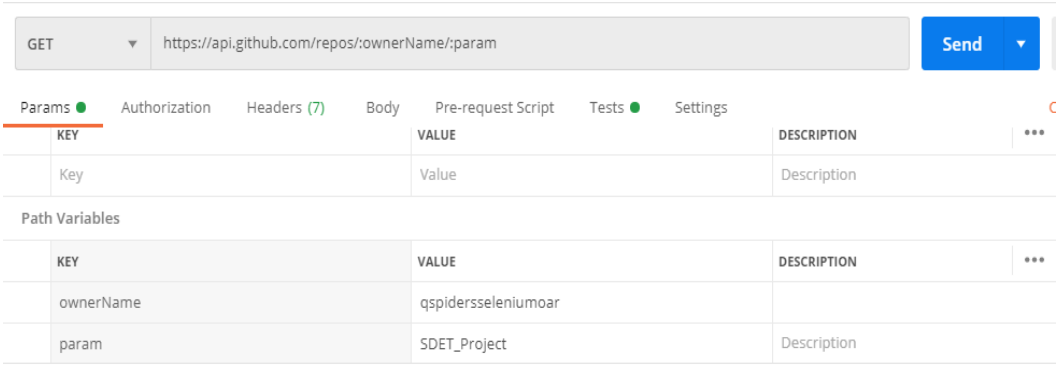
31.Explain the validation available in POSTMAN APi testing

1. ResponseBody : Verify status Code : used to verify the response status Code
2. ResponseBody : time is less than --- : used to verify the Response time
3. ResponseBody : json value Check : used to navigate to particular key
4. ResponseBody : contain string : used to check the particular String is available in entire response body
5. ResponseBody : equal to String
6. **sendRequest : we can send another request with in a request**

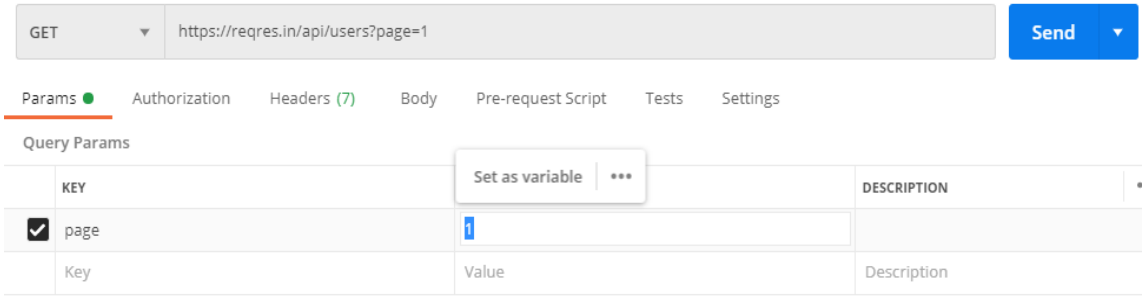
32.What is Parameter & types of Parameter?

To achieve data driven testing we go for Parameter

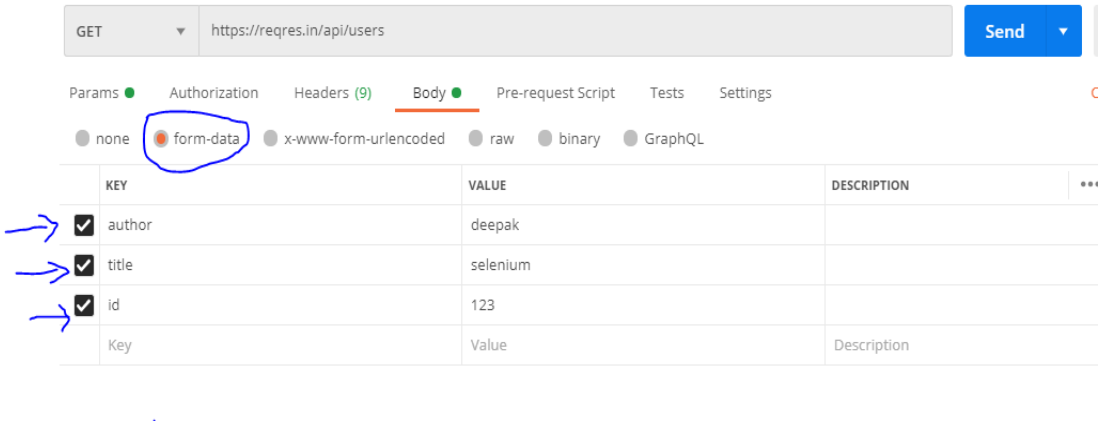
1. Path parameter: it's the Part the ENDPoint, execute same request with different data we go for Path Parameter



2. Query parameter : it's part of the Query String, execute same request with diff filter /search criteria



3. form Parameter : send complete request body in the form of Parameter



33. What Authentication & Authorization

Authentication: used to check whether you are valid user or not?
Authorization: used to check your permission / accesses to the resource

34. Types of API Authentication & which Authentication you have used in your previous project

1. Basic Auth (send a request using username/ password)
2. Bearer Token (send a request using tokenID, but token is fixed)
3. OAuth-1.0(older)
4. OAuth-2.0 (send a request using tokenID , but token ID is dynamic created via Outh-cleintAPP) or (Gmail app , allow grant permission to skillRaray app without sharing your username/password via Oauth-2 Protocol)

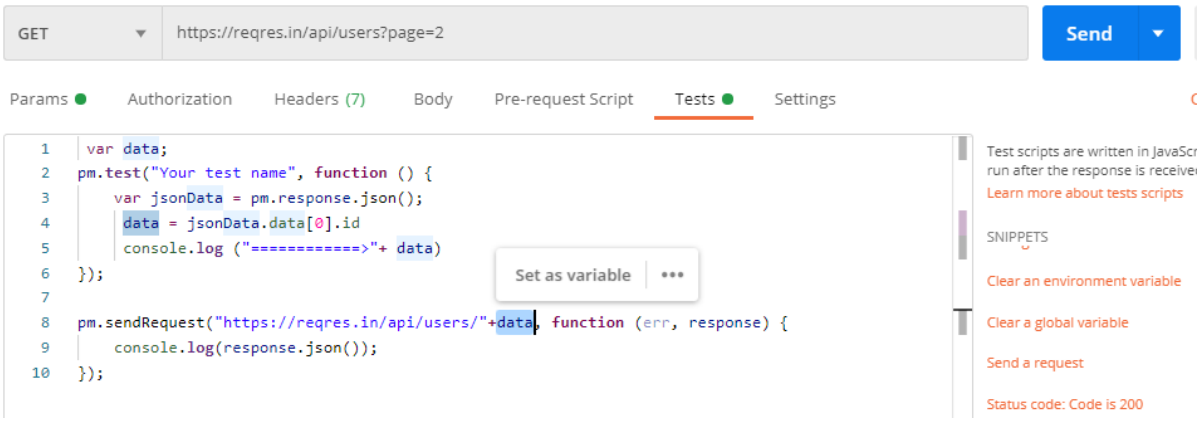
35. What is Request chaining?

Capturing data from one response using jsonXpath() , & pass same data in to another request is called Request chaining .

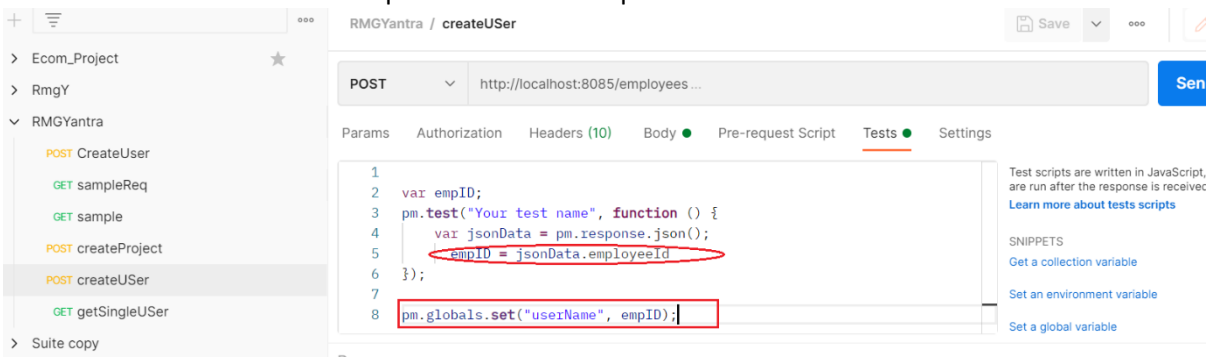
In below example, capture "id" value from the first api request & pass same data in another URL

EG :

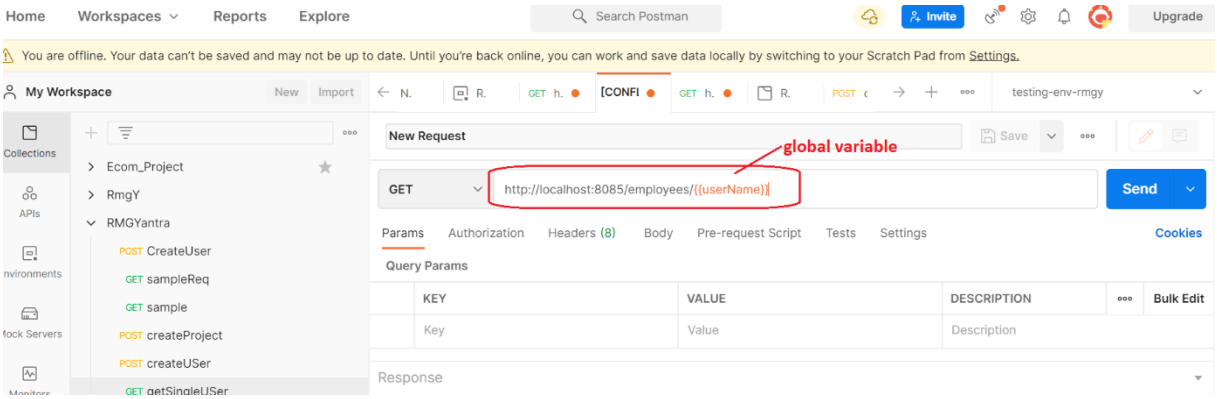
In below example, capture "id" value from the first api request & pass same data in another URL within the same api execution



EG : 2 ➔Send data from one request to Another request



Send "userName" data to another request



36.What Is Collection in PostMan

Collection of API requests is called collection or suite

37.How to execute API collection in command line

1. Export Collection from POSTMAN
2. Download Collection in .json format
3. Go to Command line
4. Install newMAN

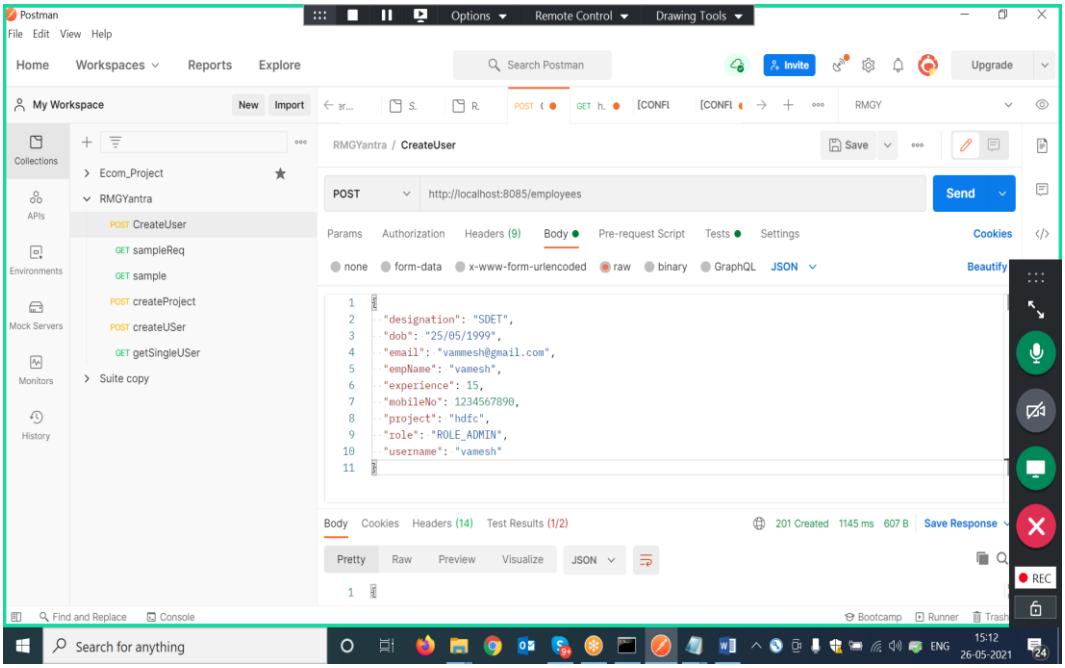
   cmd> npm install –g new man
5. Execute collction in CMD

   Cmd> newman run path

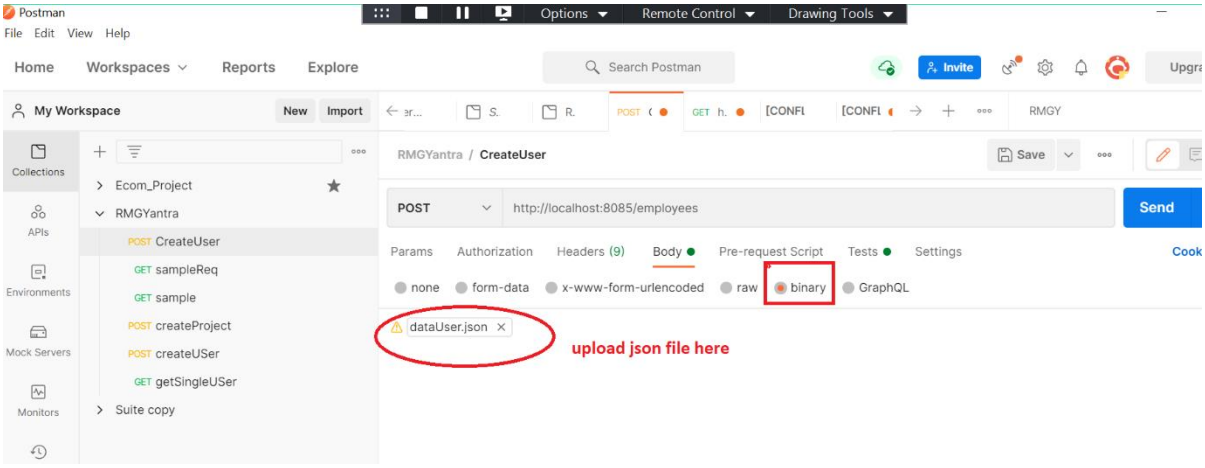
38.How many ways we can post request in JSon in PostMan

In postman we can post request in 3 ways

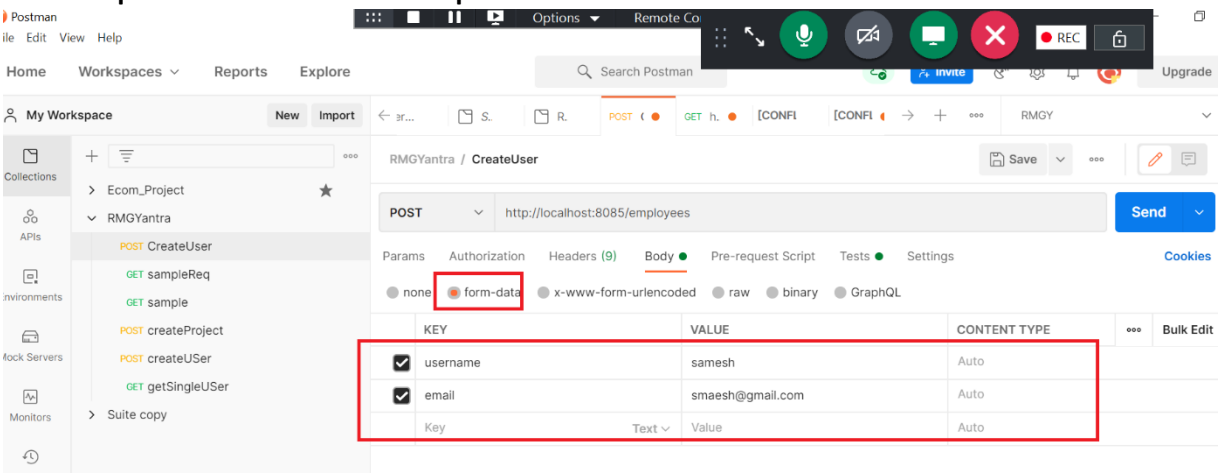
1. **Post a request using jsonObject**



2. **Post a request using Json File**



3. **Post a request in the form of hashmap**



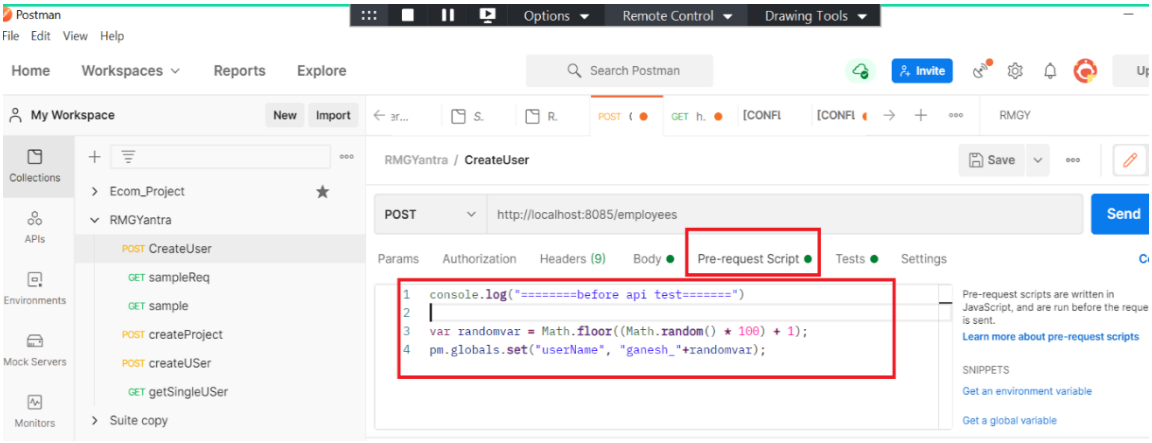
36 How to execute same request with dynamic request body

We can post same request with dynamic data without any changes in the request body,

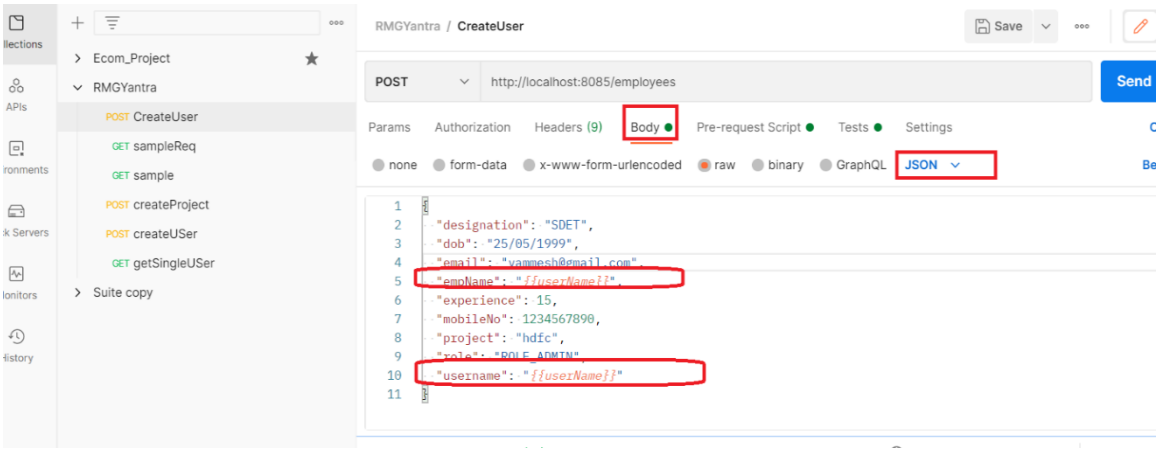
in order the achieve this fallow below steps

1. should write random math function in “pre-request-script” , then store the random data in global variable

   Note: pre-request script will be executed before the sending the api request



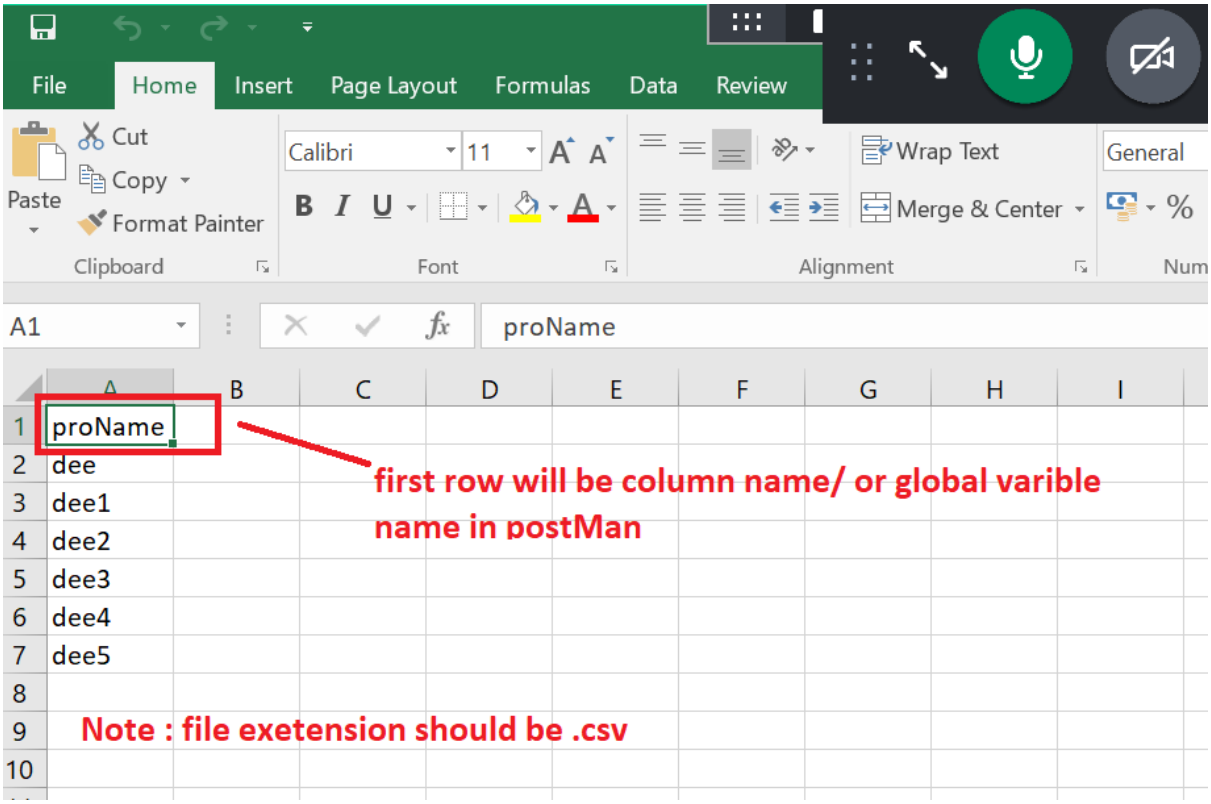
2. use the same random variable in request body



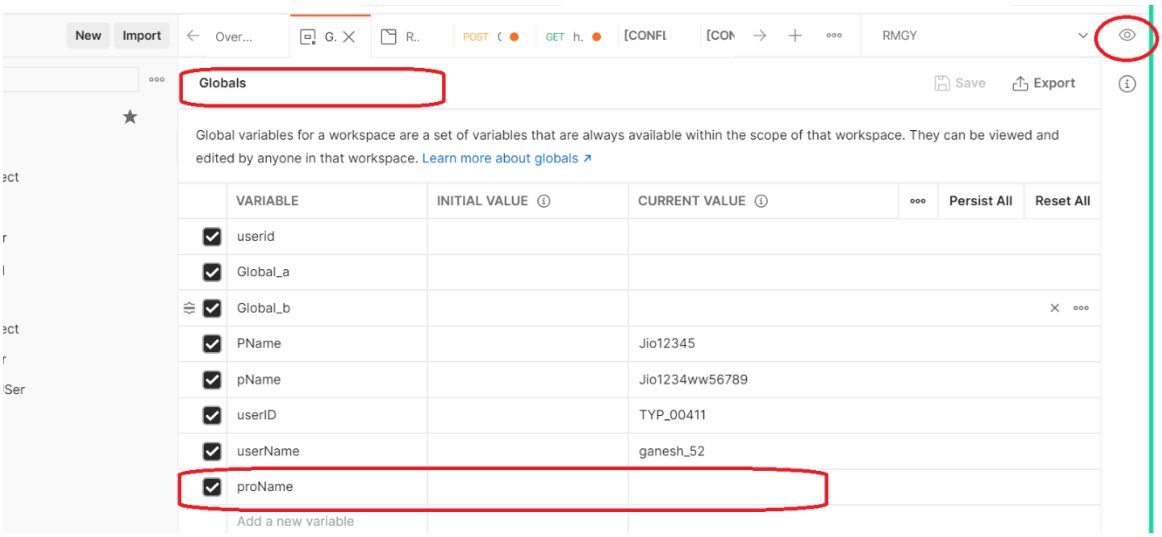
37 How to execute same request multiple time with different data

We can execute same api with multiple data using .csv file , to achieve fallow the below steps

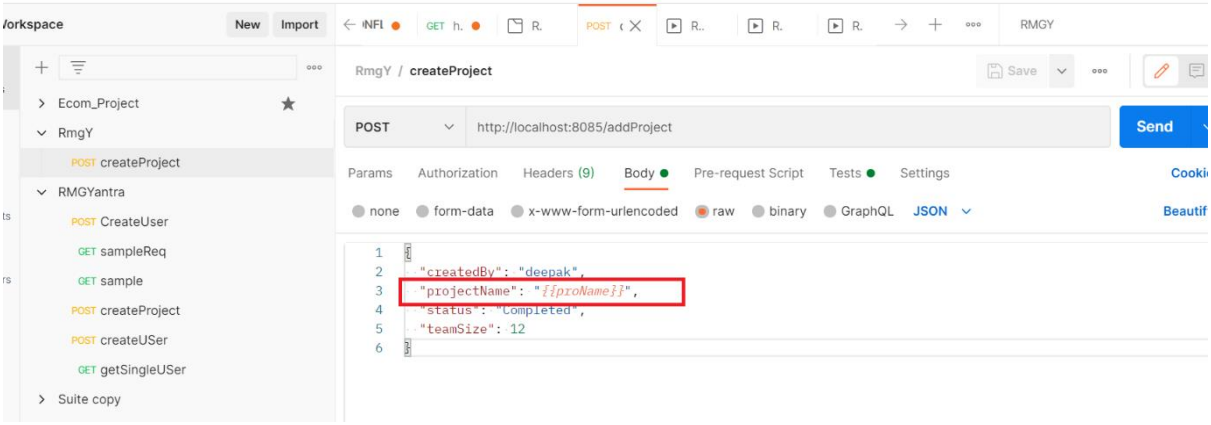
1. **Create .csv file with column name fallowed by data**



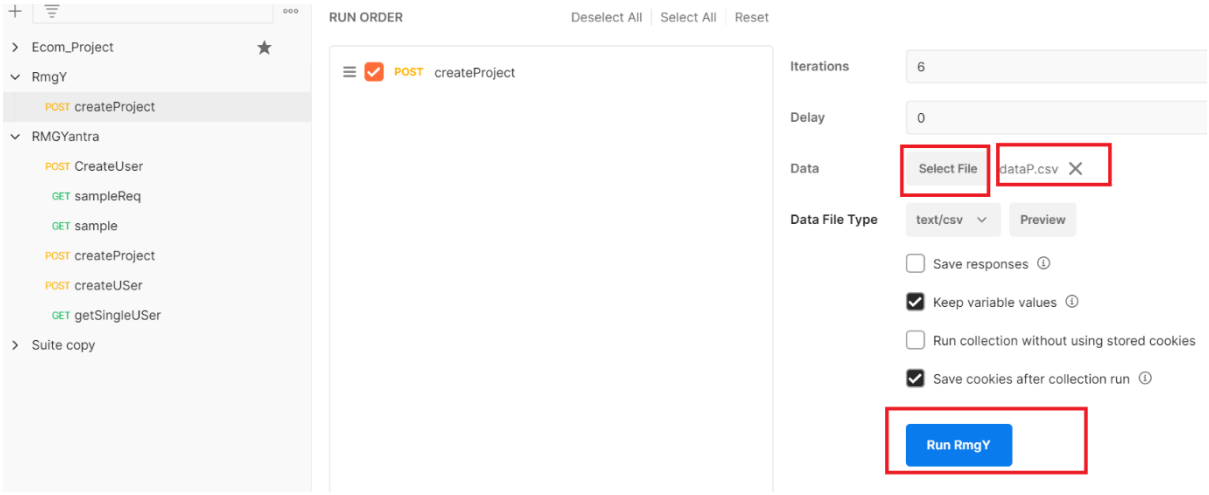
2. **Create on one global variable in postName , name should be same as column name in .csv file**



3. Set the global variable in request body

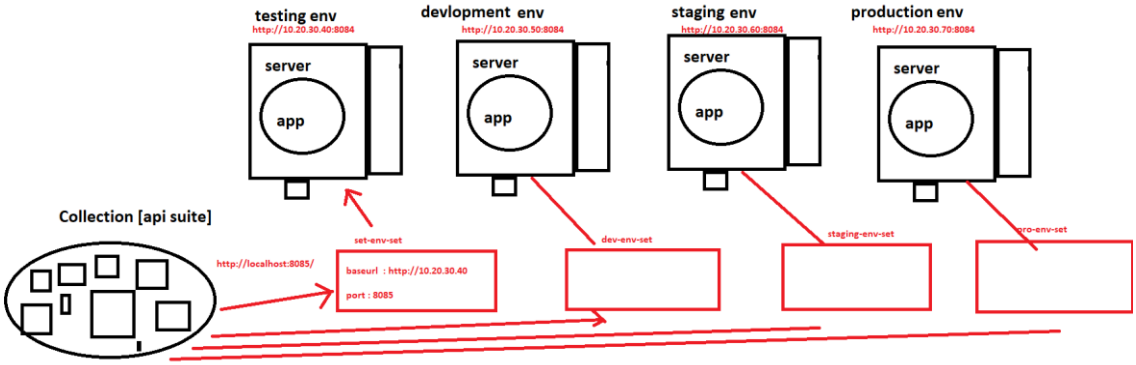


4. Run the collection with .csv file



38. how to execute API collection with different Environment like testing env , development, staging , production

In order to run same collection with different environment, take a help environment variable available in postman



EG : API URI :

Get➔ {{baseuri}}/{{port}}/endpoint

39. Explain the variables available in postman

1. local variable: (scope: it's a data within the api request) can be used with in a request
2. Global variable: (scope: it's a data across the collection) can be used between the api request, outside the collection also
3. Environment variable: (scope with in the collection) it's a configuration data to run the collection

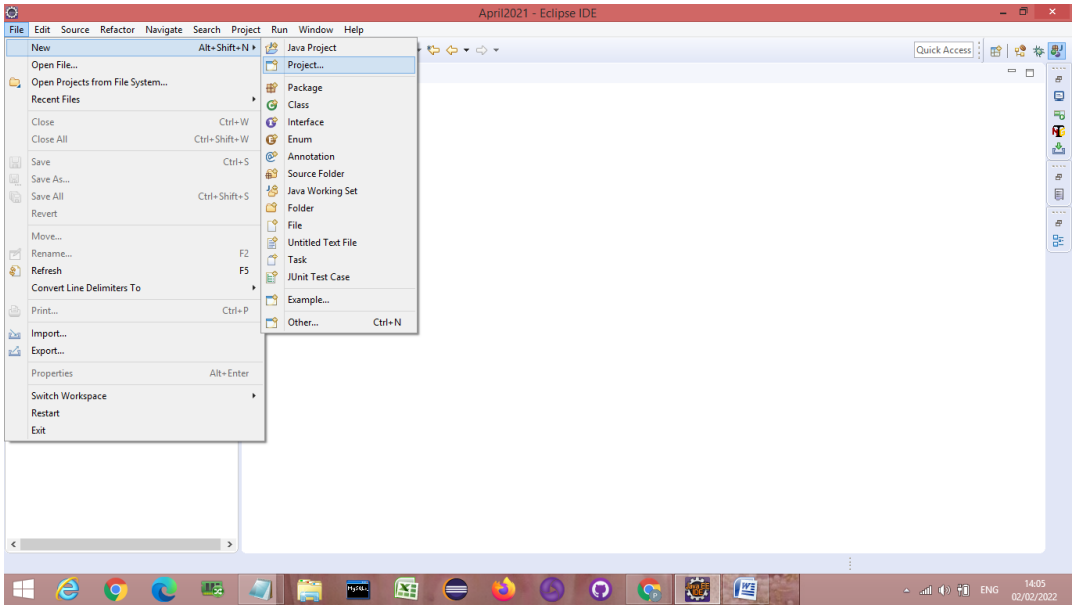
---

39. What is the difference between Oauth1.0 Oauth2.0

Oauth1.0	Oauth2.0
older version of protocol for authentication	latest version of protocol for authentication
Two level authentication required for every api	One level authentication required for every api
complex authentication approach	Simple authentication approach compares to Oath1.0
To get bearer token , should pass consumerID , consumer secret & accessID , access secret	To get bearer token , should pass Client & Client Secret

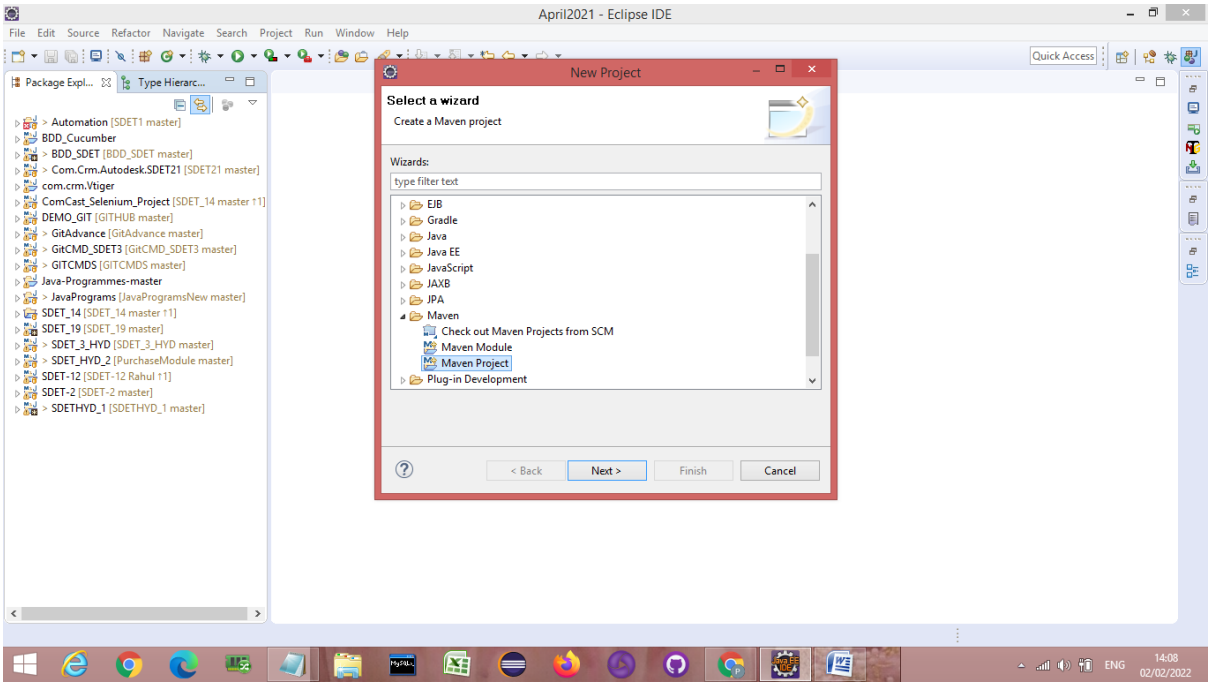
How to create Maven Project in Eclipse ?

➔ Click on File → New → Project



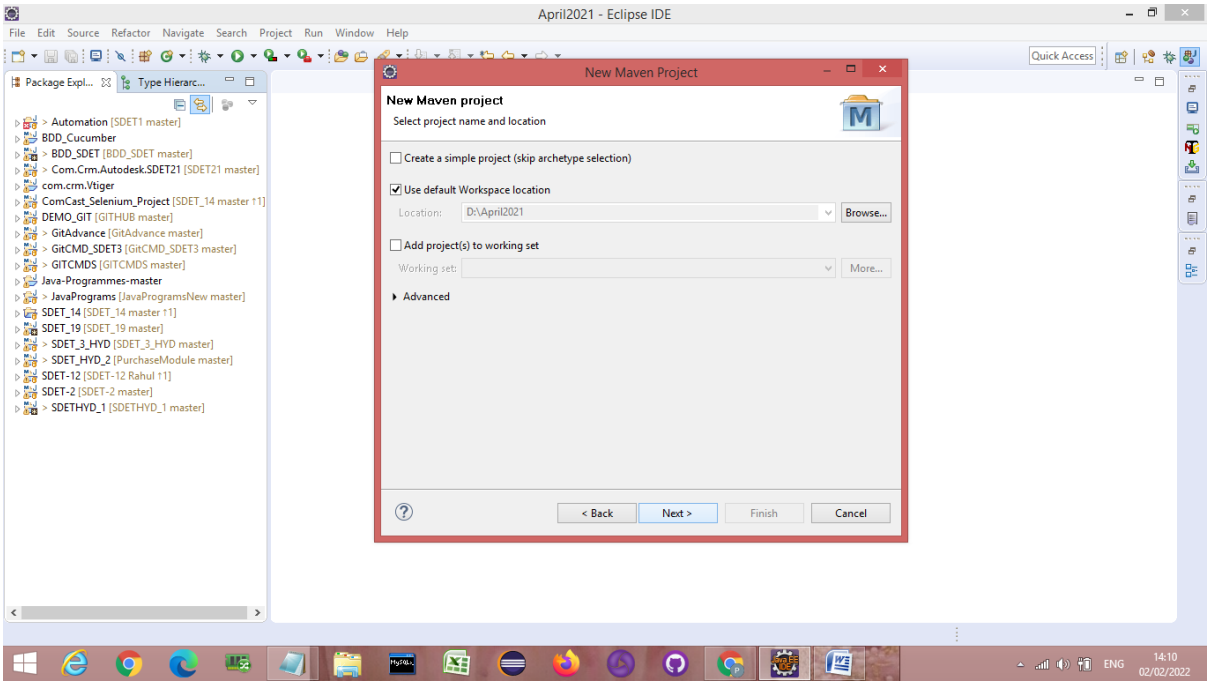
➔ Click on Maven Folder → Maven Project → Click Next

➔

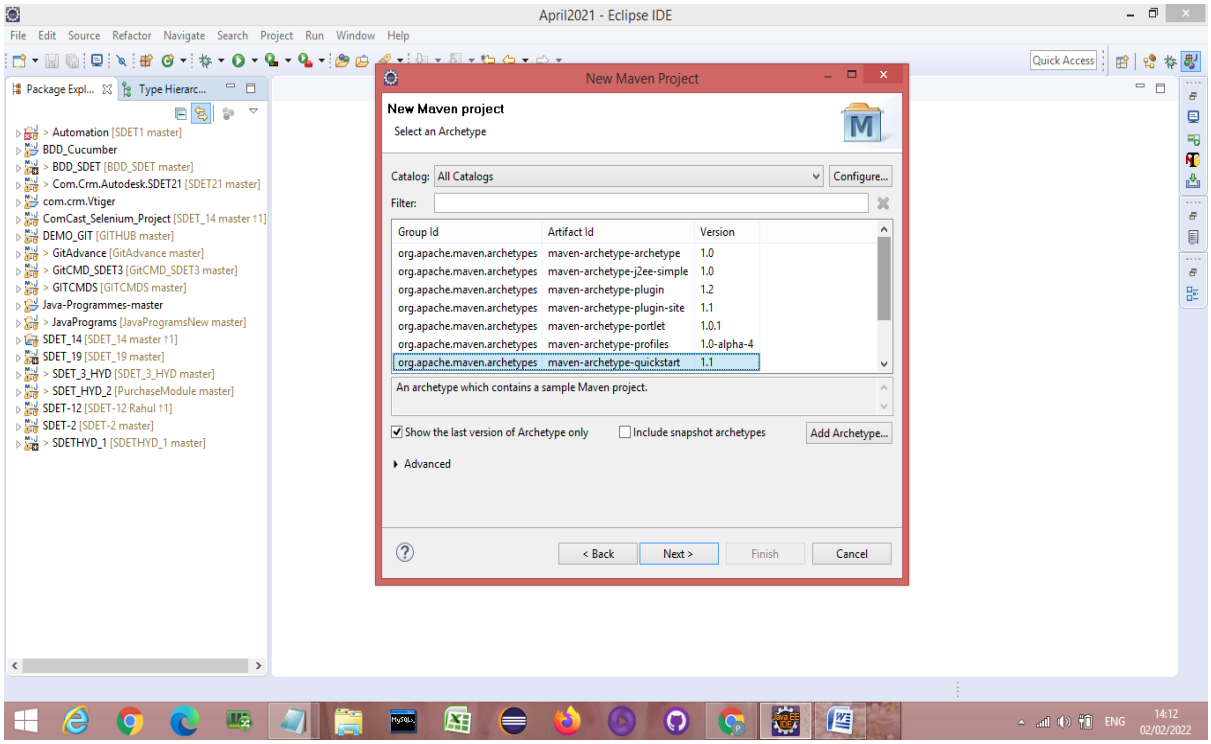


➔

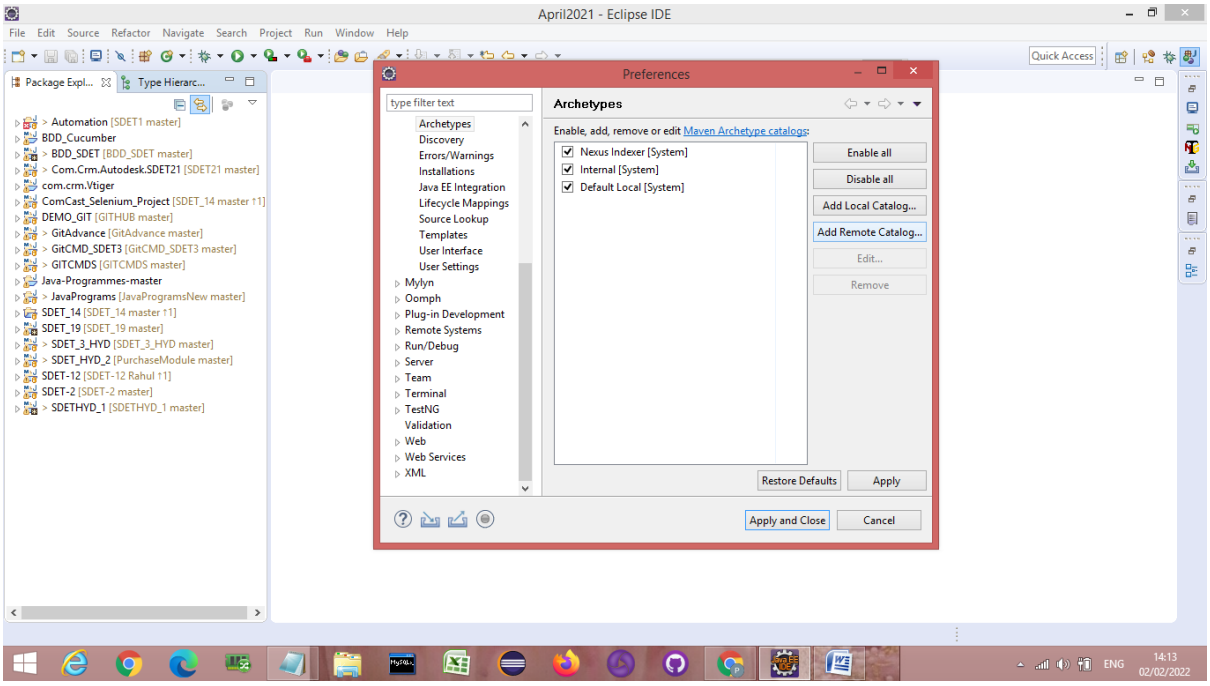
→ Select the Workspace → Don't click on the first checkbox → Click on Next



→Click on Configure →

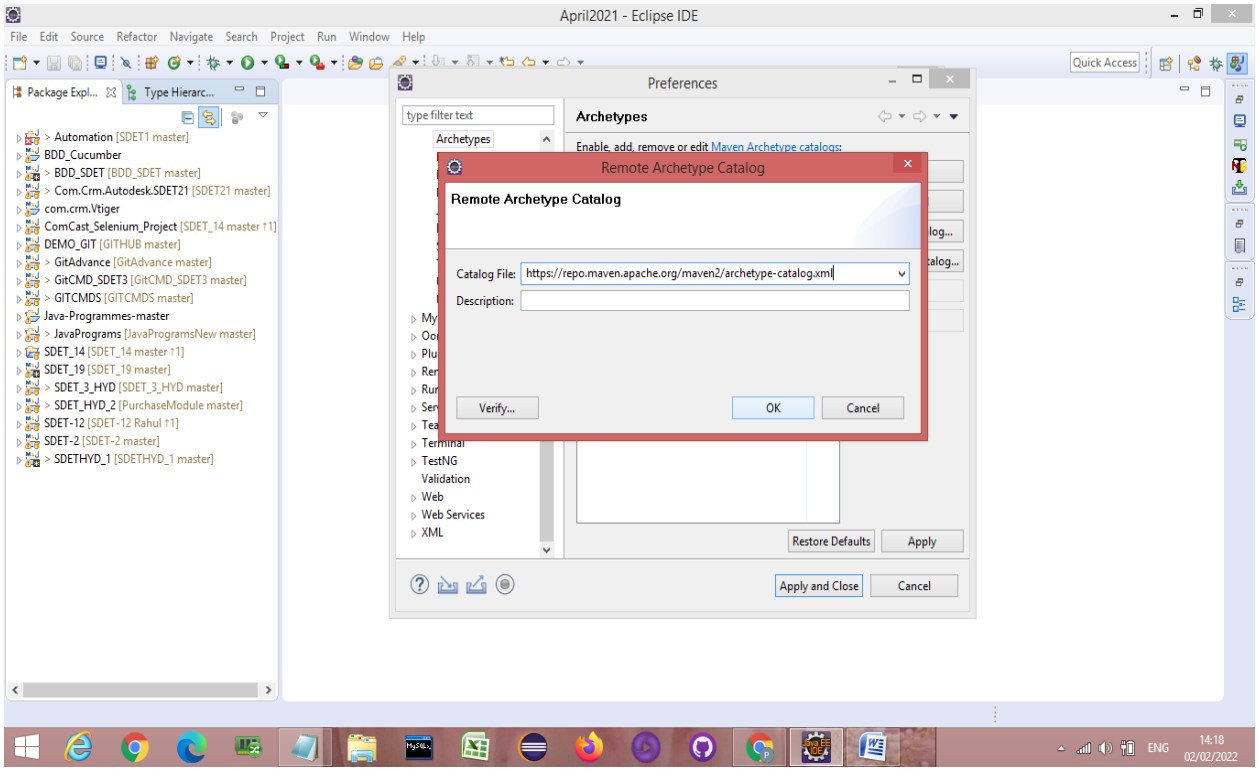


→ Click on Add remote catalog

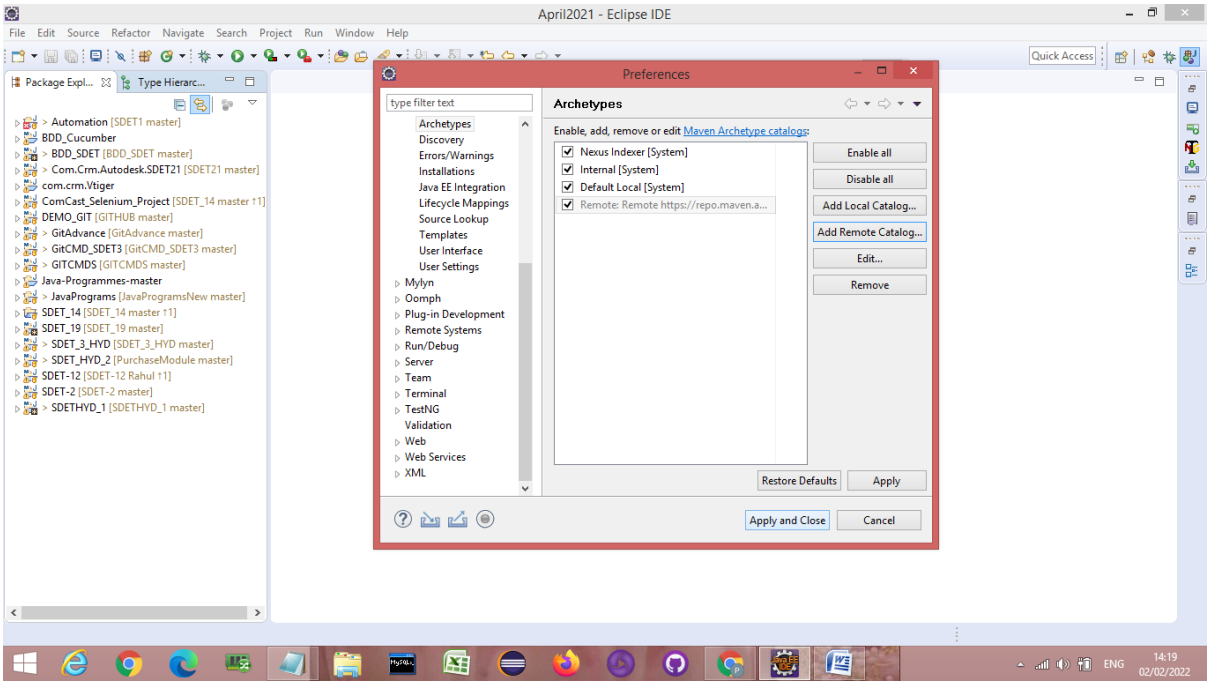


→Add the below link in catalog file textbox → then Click on OK

https://repo.maven.apache.org/maven2/archetype-catalog.xml

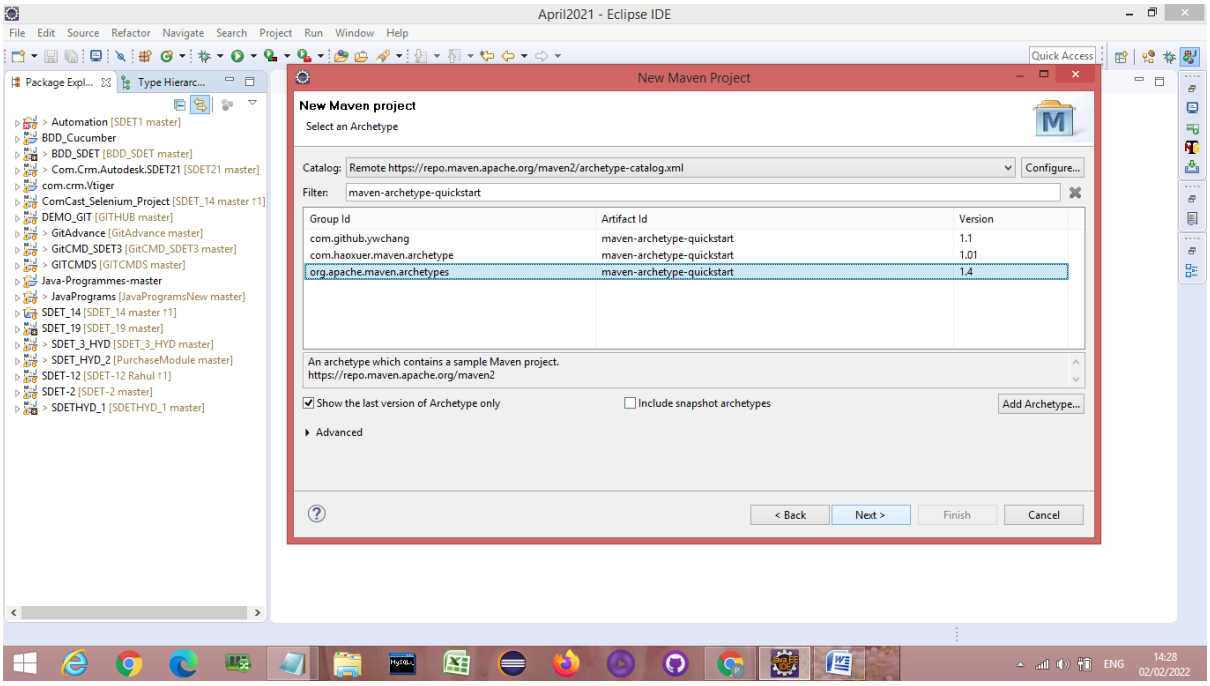


→Click on Apply and Close → This will start downloading Archetype make sure you are connected to good network connection.



→In the filter textbox type →maven-archetype-quickstart → You will get 3 suggestion →Select

maven-archetype-quickstart 1.4→then click on Next



→Fill the below details in textboxes

- **→ Group Id : DEMO_MAVEN_Project**
- **→ Artifact Id : DEMO_MAVEN_Project**

Then click on Finish → Maven Project will be created