**API Testing**

**Tool:** Postman

Main Goal of all testing (Such as API, Performance---- etc.) 🡪 to release the quality product or software according to the customer satisfaction.

API🡪 Comes Under the backend side testing. Also (Database (SQL) Testing)

Web Testing 🡪 Comes Under the Front End Testing (Tool🡪 Selenium)

**For different type of testing we used different Tools**

API Testing can be performed by both Way 🡪 Manually and Automation

Manually 🡪 Postman

Automation 🡪 Rest Assured

Pre-requisite 🡪 (Programming Language 🡪 Java)

**Client** 🡪 Software or hardware where we use or access different services from the server.

Suppose through **browser we browse or access different pages.**

Where we are trying to access the application or services.

**Server** 🡪 where the actual software is installed

Remote server access through **URL** but in some organization Server are placed in their **local Environment.**

**Three Component of Web Application**

Presentation Layer 🡪 Application Layer 🡪 Data Layer

(UI View) Client Business Logic (Server) Data

**Business Logic contains the API** 🡪 Suppose first we send request to the **middle layer** which are basically API’s and these API’s Send request to the **Server (Database Layer)** Fetch the data and display the data to the user.

**API’s** are the who are taking the request and responding to us according to the request.

API’s Act as Intermediate between Presentation and data layer. API is used to communicate two different applications.

**For Example:** Through our google account we can be logging into FB, Instagram, LinkedIn Etc. So all these apps send API request to google to verify that either the following google account is available or not.

**Two Types of API:**

SOAP (simple Object Access Protocol) and REST (Representational state Transfer) API

Traditional Type of API New and Advance Type of API

**For Design and Development Testing we don’t need Web services like (Internet)**

But when we conduct whole testing and everything is running fine and smoothly then we move it to production where we need Web Services (Internet)

When we test API on our local system then we called it Just API but when moved to production and make it available to the public then we called it web services.

**All API are not Web services but ALL Web Services are API**

Moved it to web Environment we called it web services

**Rest API Have Different Type of Methods:**

A lot of method we used but mainly we used four methods:

Get 🡪 when we are trying to access some data on google (Suppose we search something on Google we actually get data.

Post 🡪 Suppose we want to store some data on database we simple create data and it actually store in database then this is through post method (we are sending our data to the server which will be stored)

Put 🡪 Put mean we want to update our existing data

Delete 🡪 Delete Mean when we wants to delete some data

These methods are called API Rest Methods or Http Request Methods

HTTP/HTTPS 🡪 HTTP is not secure the data is not in the secure form but https the data is in encrypted form (secure data)

URL (complete)

<https://www.google.com/articlename>

URI

URN

Resource 🡪 Something that is available at Server and we need to request it

**Postman is supported both (Desktop and Web Based)**

* Created Workspace first

Area where we maintained our files (Test Cases and reports) and saved

* Next Step is creating Collection (Containing Number of Folders and https request)
* Without creating collection, we cannot create any number of test cases and https request
* Inside Workspace we can create multiple collections
* After creating collection, we are able to create request (Multiple Request)

To Test the API (Two Important Things (Request and API))

**Request 🡪 API 🡪 Response**

Input Fetched Data from DB Output

We Do Validation On Response Like status code etc.

**Difference between Put/ Patch**

Put🡪 Change the whole data when we want to change the entire data in DB

Patch 🡪 when we want to change the piece of or particular data in database.

Website (Reqres.in)

<https://reqres.in/api/users?page=2> **Query Parameter (Filter data)**

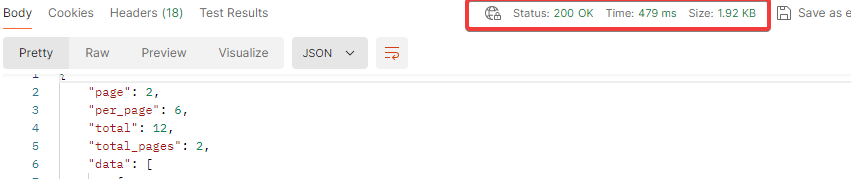
**Host and Domain/** **Path Parameters**

Cookies-🡪 Temporary Files when our browsing something so cookies are created

History or Information is saved on Cookies Form.

**Main Testing/ Validations**

Validation on Status, Size, Time, Response Body (JSON, XML), Header, Cookies



**Hierarchy**

**Create** workspace

Collection

Request (get, Post, Put, Del)

**Get Request:**

Simply Put URL on URL Search Bar and See the status code, Size, Time, JSON Body

**Post Request:**

URL, Set Parameter on Body (JSON Format)

Request Payload (what we input) and verify that 🡪 What Response Payload generated

**Verify both and then see the status**

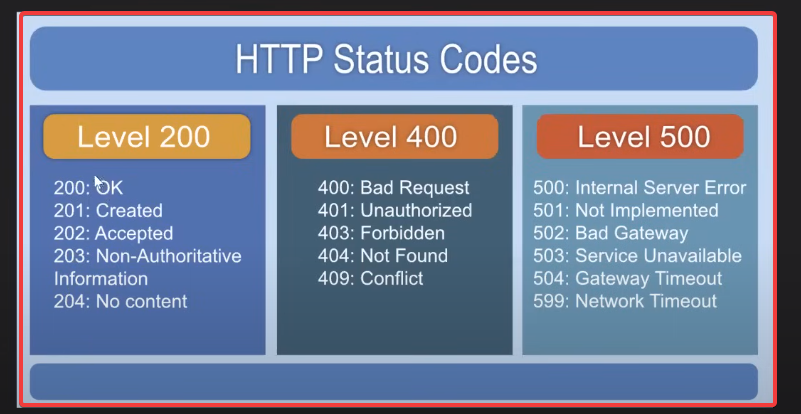
**HTTP Status Code:**

3 Levels

200

400

500



**Create Own API’s**

* Need Pre-installation
* **(Step 1)** NodeJs (By this npm automatically installed)
* NPM (Node Package Manager) (By this we can install the below)
* **(Step 2)** JSON Server (Command: npm install –g json-server **(On Terminal)**)

We can also import API Collection in **Postman**

**(.json** file 🡪 we import these files in Postman)

We can use it by both way (by URL and by Importing to our local system)

**In Local System:**

Create or save file (with the extension **.json)** create **to create json file API we first run on local system then API generated and run on postman to see the result**

We set Authorization Token or main URL against the word in the Environment.

Assertion 🡪 is called validation.

To **Add Assertion** in Postman 🡪 There is a library name **(PM**) 🡪 In PM Library there are various function in it.

For that function we need to understand some basic of JavaScript.

**Write Test cases in Scripting Section**

**Testing Status Code**

**Verifying Status Code:**

To Add **assertion** or **test or validation points** then write it in **Script Section**.

pm.test("Status code is 200", **function** () { Testname

    pm.response.to.have.status(200);

});

Expected Outcome result

And compare it with the actual outcome

**JSON Schema:**

**Schema:** Schema define the type of data which we can store in Json File

Also Known as Meta data (The data about data)

Means What kind of data store and what is value and its type.

Through Schema we can store the data in json file

------------------------------------------------------------------------

We can also generate online Schema by giving response body data. (json to json schema converter)

**Project:**

**OpenCart Application API: (E-Commerce Application)**

Installed Application on the local system. 🡪 Followed the Documentation ( **For Proper Installation) 🡪 because we cannot access API through online or browser.**

**Local system:** Installed Xampp

**Opencart Application URl:** localhost/opencart/upload/index.php 🡪 FrontEnd

**Admin Dashboard:** localhost/opencart/upload/admin

**Pre-Requisite:**

**Main Work:** Go to Admin Dashboard 🡪 localhost/opencart/upload/admin / username: admin, password: admin

**In Admin Section:**

**Go to Users** 🡪 API 🡪 Create API Keys 🡪 then Set IP (To Get to Know About API used cmd (type ipconfig)

**API Username:** opencart12

**API Keys:** zqR7CT25vMxGdRBQeGfwbblU166wrMhNniNtrpBgSKLA7BIB0n6ZHKFV1Fls8XnOC3bCOhzfj1VvhSj4KtH8xr6oTCkojR8HEUGMpDwsk38OQJVefqHme3Rp1ppyL5bclDqOaR3VLZtzDEaPzsjJWMNJEb0ItyghMnrFIIIREhfeoxD0XWWfqNjLcUndagEkDvF81c0t7BCXM7gWzferkbsRGhe7HOjKOcGjRhozXQPyX6MICGKckzkqoqVr6vog

**IP Address:** 192.168.11.131

**OpenCart API Documentation:** <https://docs.opencart.com/en-gb/system/users/api/>

<https://documenter.getpostman.com/view/4924228/UzXM1JmQ>

**Module**🡪 CartModule

**Flow of API Testing of This Module**

Create Session/Token

Add Product to the Cart

Get the Product

**Edit the Product**

**Delete the Product**

**Create Session:**

Sometime we set variable for common thing instead of using whole again and again

Create Collection 🡪 Go to Variable and **Set Ip** and **Base URl variable**

BaseUrl : https://myopencart.example.com/index.php?route