* Environment/VPN Setup
* QA Handover (Functional Understanding/VPN Setup)
* Test Case Planning (Test case Writing/Upload in Test Repository)
* Test Execution Cycle-1 (Defect Reporting, Test Case Execution)
* Test Execution Cycle-2 (Defects Retesting, Round-2 Test execution)
* Final Sanity after Round 2
* Test Summary and Sign-Off Process.
* Load Testing(Env Setup)(If required from Project)

As per my skills and experience I can do the job and deliver great results. I have working experience in all the shifts(US/UK, Middle east,ROW). I delivered almost all the projects before the deadline. I am Quick learner and team player. I have communicated with almost all stakeholders with a positive attitude and provided the best results. Can be able to handle multiple projects simultaneously. You will not disappoint after hiring me.

In the next five years, I aspire to work in a company that challenges me to perform better than my capacity. I see myself learning and growing with every experience and contributing to different projects with sharp analytical and innovative skills for achieving the best results.I would like to see myself in a position to propagate great innovative ideas and lead my team towards such ideas.

Web Applications :

Client -> eg Browser (Is a computer, from where we are trying to access the application)

Server → Software actually installed (Web pages)

Contains actual application and through client web application these application

Between client and server we need to established the connection, i.e through internet

Server usually located at remote location, not at the local(it can be but not always)

Client server architecture : 3 Types

Web applications usually followes 3 tier architect

1 tier → no database , no internet, client and server(file system) in the same machine/single system

2 tier → Multiple client machine accessing the same DB, and this DB server is placed somewhere in remote machine

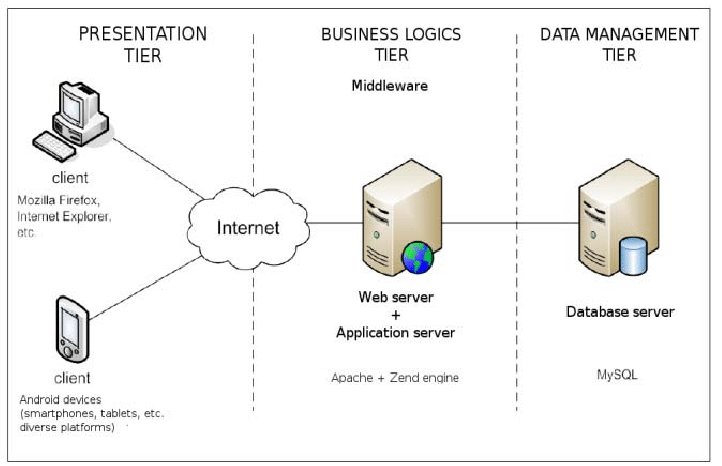
Eg, Hospital

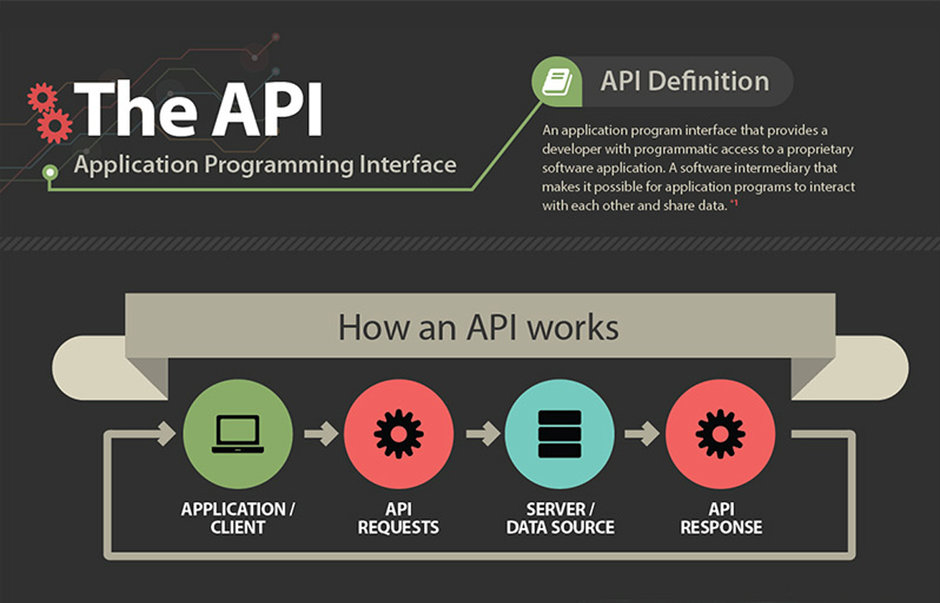
3 Tier → All web application comes under 3 tier architecture

Client

Business logic layer

Database layer





To fetch the data from backend and display it to the frontend , i.e. on web browser

API is act as intermediary between client and server, with serves the request and response between the two and maintain the communication active.

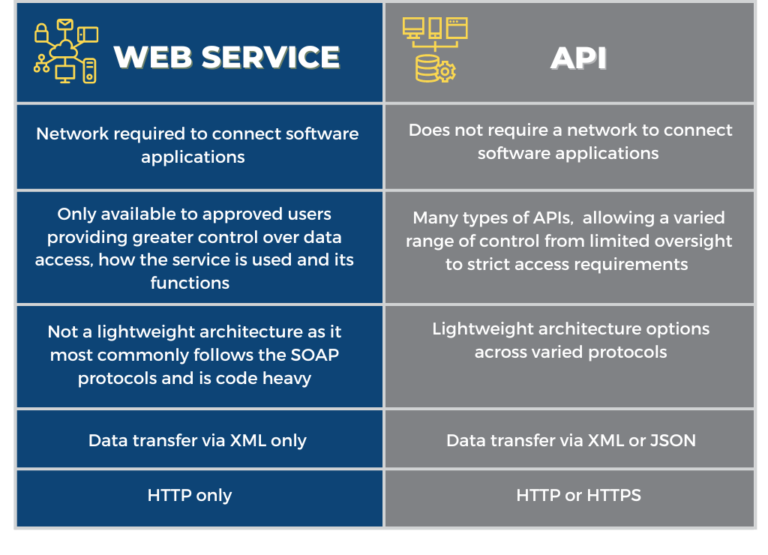
Types of API

1. SOAP - simple object access protocol
2. REST - Representational state transfer

API vs WebServices :

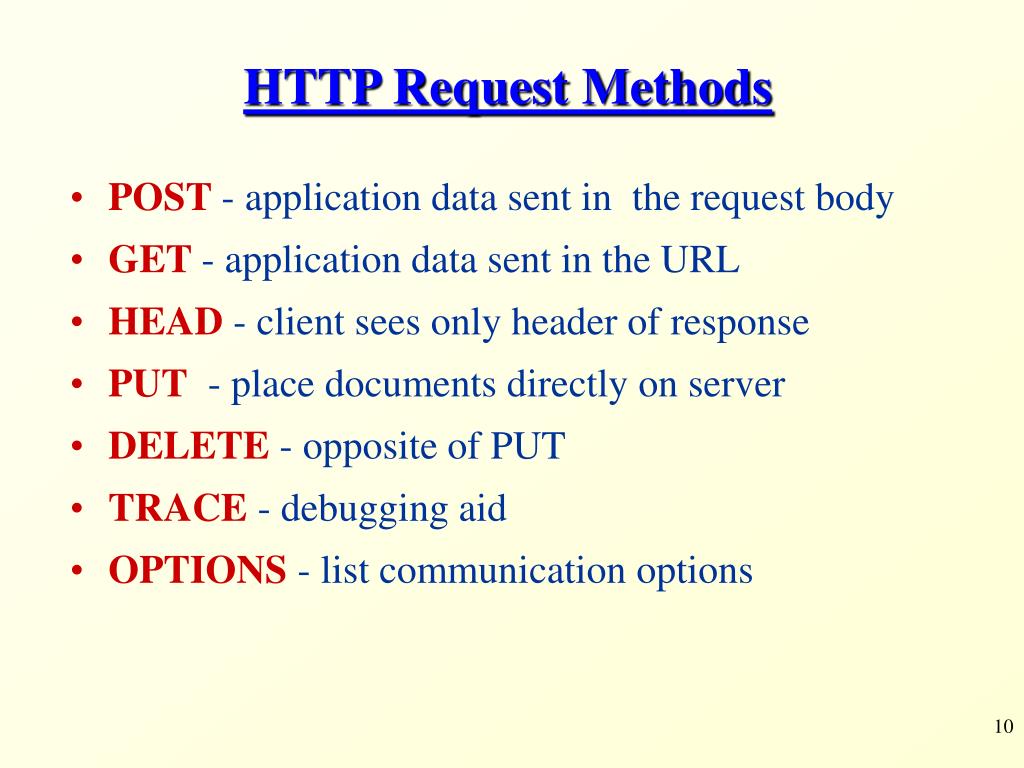
When u place the api on internet and everyone is able to access that apis then it would be termed as Web service.

All web services are APIs but all APIs are not web service



REST API methods : (Or HTTP request)

1. GET → To get the information from the db server
2. PUT → Update / Edit the data from DB
3. DELETE → To delete the data from DB
4. POST → To create something in the db server



Response Validation :

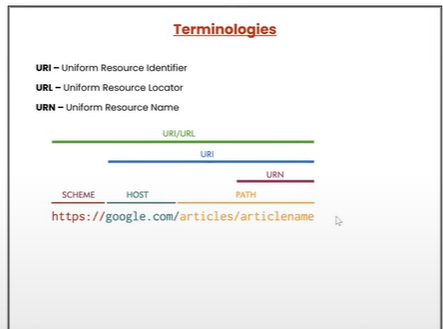
Http vs Https →

Http → data without validation/encryption - hacker will easily access the data or can misuse the data

Https → Data with encryptions , more secured

Payload → Body of the http request and response message

Can be in json/xml format



Website for some sample APIs

<https://reqres.in/>

**POSTMAN**

**Api Testing tool → Manual Testing of apis using postman tool**

1. Creating a new workspace → create, rename, delete

Workspace → Area where we maintain everything - files, test cases,test data or whatever task.

Basically it is space created by google when you create workspace

Created by postman by using your gmail account

Workspace is the first basic prereq in postman. Without it we cannot perform any task

1. Creating new collections: create,run, rename, delete—> collection is nothing but the folder.

Inside the workspace we can create N number of collections. It contains a number of folders and different types of HTTP requests.

1. Request → API → Response

Get, post, put, delete — the types of request

Get – retrieve the resource from service / data from database

Post – create resource on database

Put – edit / update

Delete – delete the data

Patch – Put will update complete resource, but if you want to update the partial details of resource

URL →

URI → protocol + domain information

Path parameter → Location of folder / resource on server

Query parameter → exact query to fetch the data.

Eg → <https://reqres.in/api/users?page=2>

URI → [https://reqres.in](https://reqres.in/api/users?page=2)

Path parameters → [/api/users](https://reqres.in/api/users?page=2)

Query parameter → ?page=2

Validation on response →

Status code

Time taken to retrieve data

Size of data

Response body (XML/Json format)

Cookies (Temporary files created by google when you search/browse something

Or logging for the very first time )

Headers

HTTP Status code

200 Level

400 Level

500 Level/series

