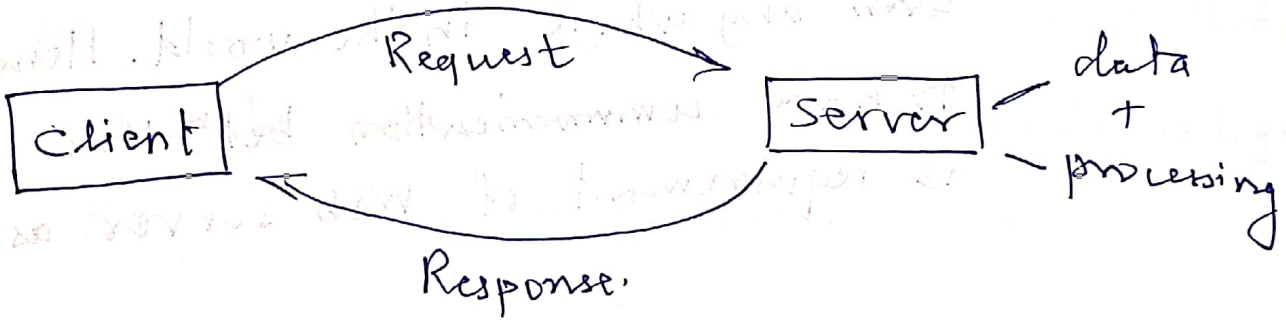
Web Services Testing.

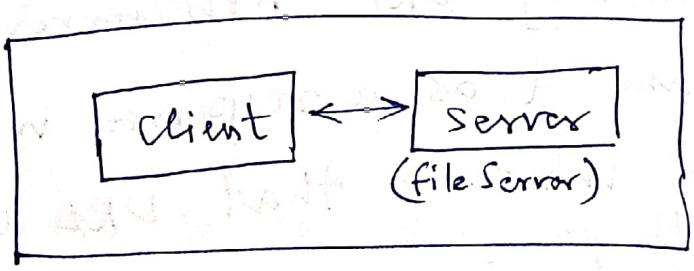
- **Client** - A client is a computer hardware device or software that accesses a service made available by a server.

- **Server** - A server is a physical computer dedicated to run services to serve the needs of the other computers. Depending on the service that is running it could be file server, database server or web server.



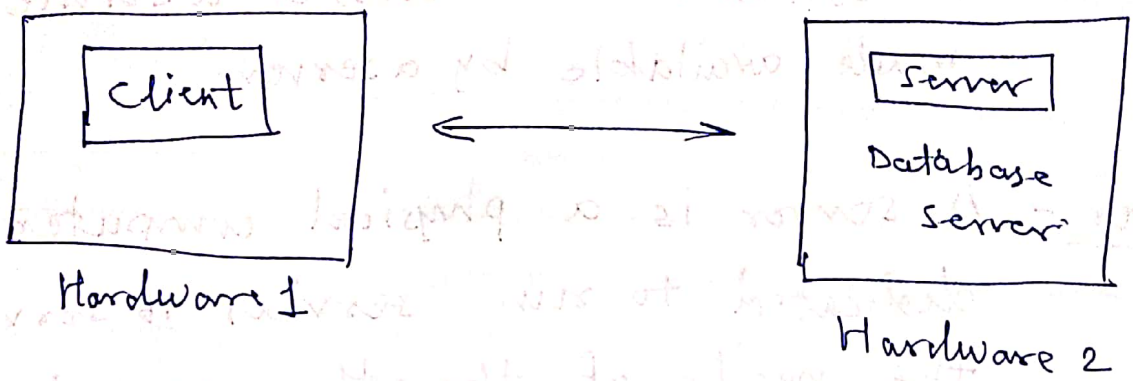
- **Client / server Architecture :-**

a) **1-Tier** - Both server & client located in same device / hardware
- e.g. Ms Excel, Word etc.



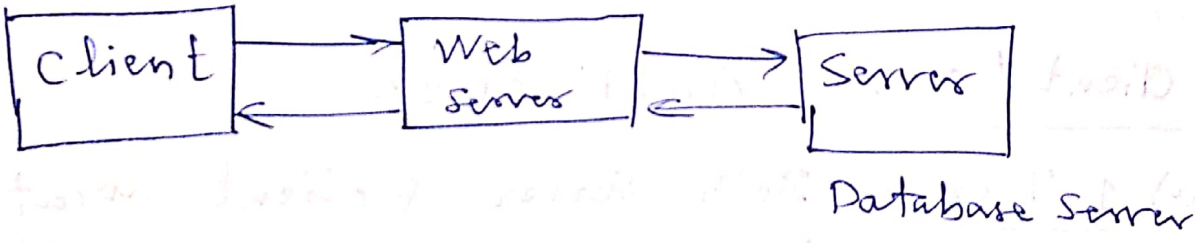
Device 1.

b) 2-Tier :- - Client can be at some other place & server will be at some different place.



- e.g. client-server application., using LAN we communicate.

c) 3-Tier :- Client & server are at remote location ~~even~~ any where in the world. Hence to have communication betn them there is requirement of Web server. as well.



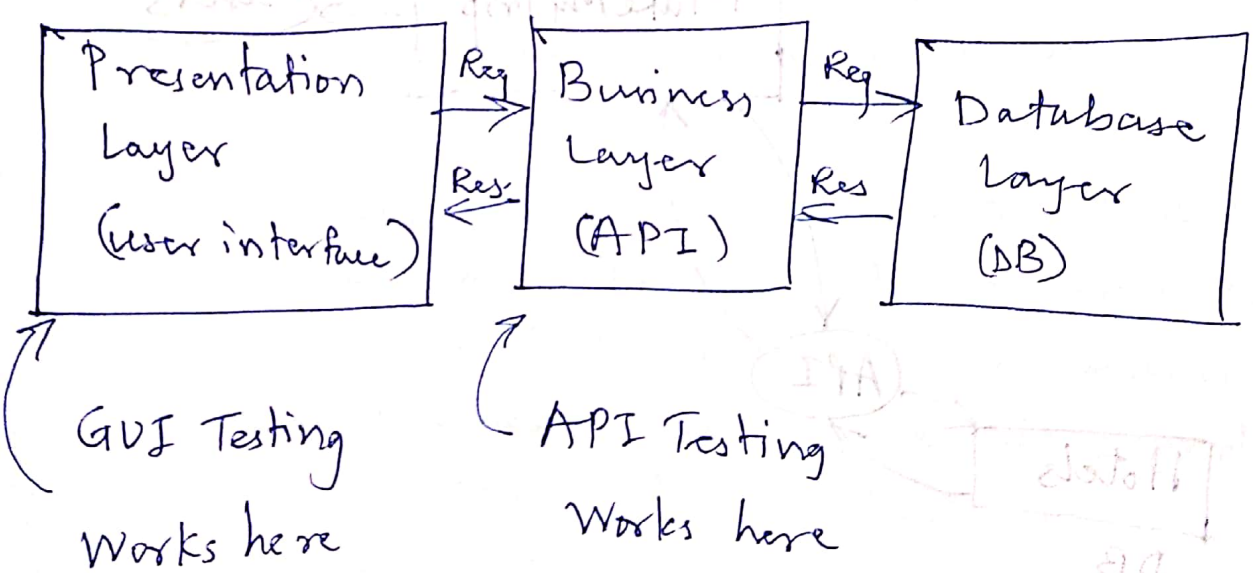
- e.g. All internet applications.
- Whenever we hit any URL web server sends request to database & as a response we get the home page or whatever that URL contains.
- So in web server bussiness logic is present

→ What is an API?

- API (Application Programming Interface)
- API are application interfaces, meaning that one application is able to interact with another application in a standardised way.
- It enables communication & data exchange betn two seperate software systems.
- As we have GUI, which helps user to interact with software, the API helps one software to interact with other.

→ What is API Testing :-

- API Testing is different than the GUI Testing & mainly concentrates on the buisness logic Layer of the software architecture.

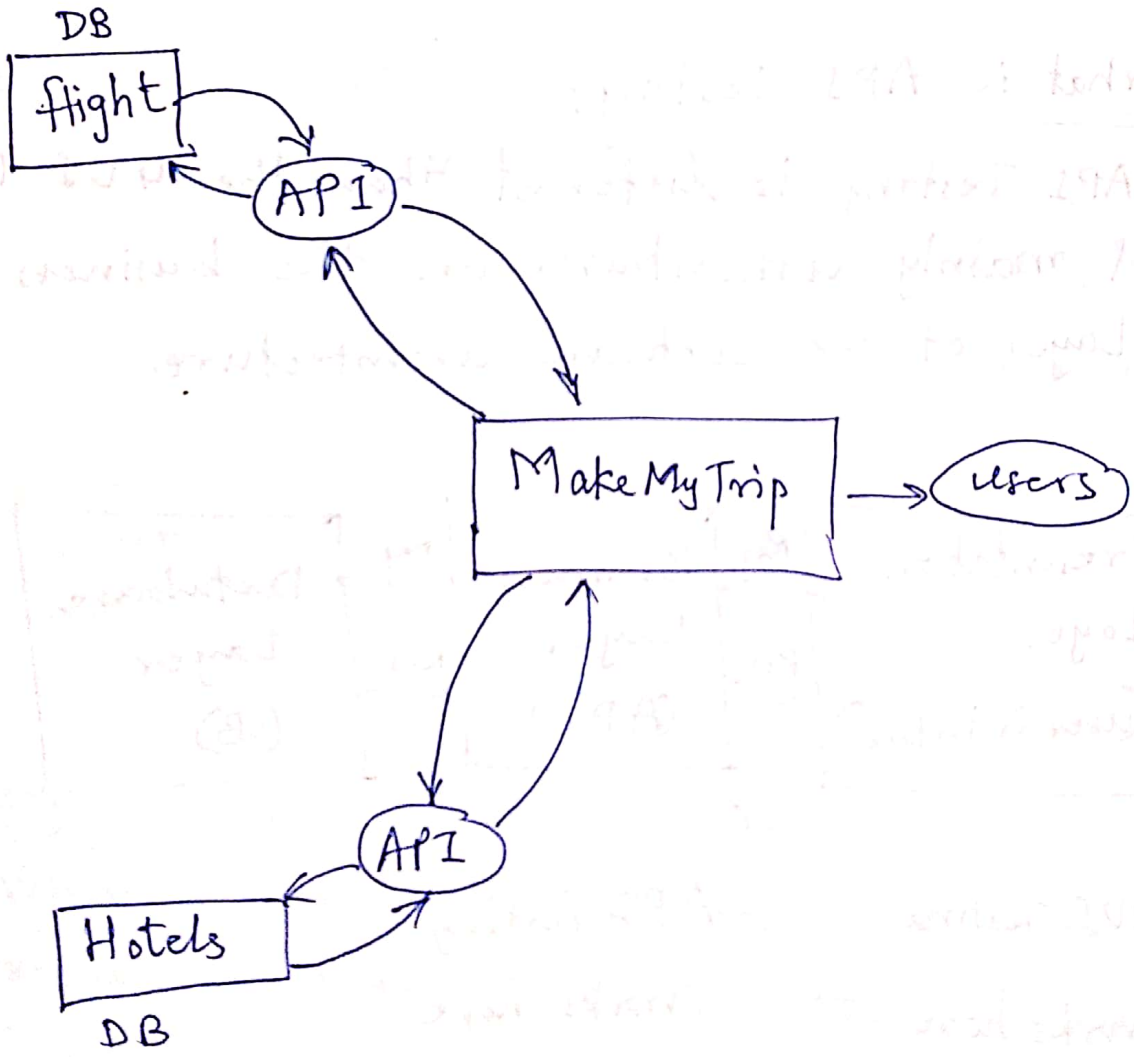


- In API Testing, you use s/f to send request to API & get o/p which is the system response

→ What is Web services?

- It is the service available over the web.
- It enables communication bet^n applications over web.
- If API uses internet then it becomes web services.
- If API does not require internet then both client & server will be on same device & it becomes stand alone applications API.

— × — × — × —



→ Difference betn API & Web services.

- Web service is an API wrapped in HTTP.
- All web services are API but all APIs are not web services.
- Web services need a network while an API doesn't need network for it's operation.

→ Components of Web services :

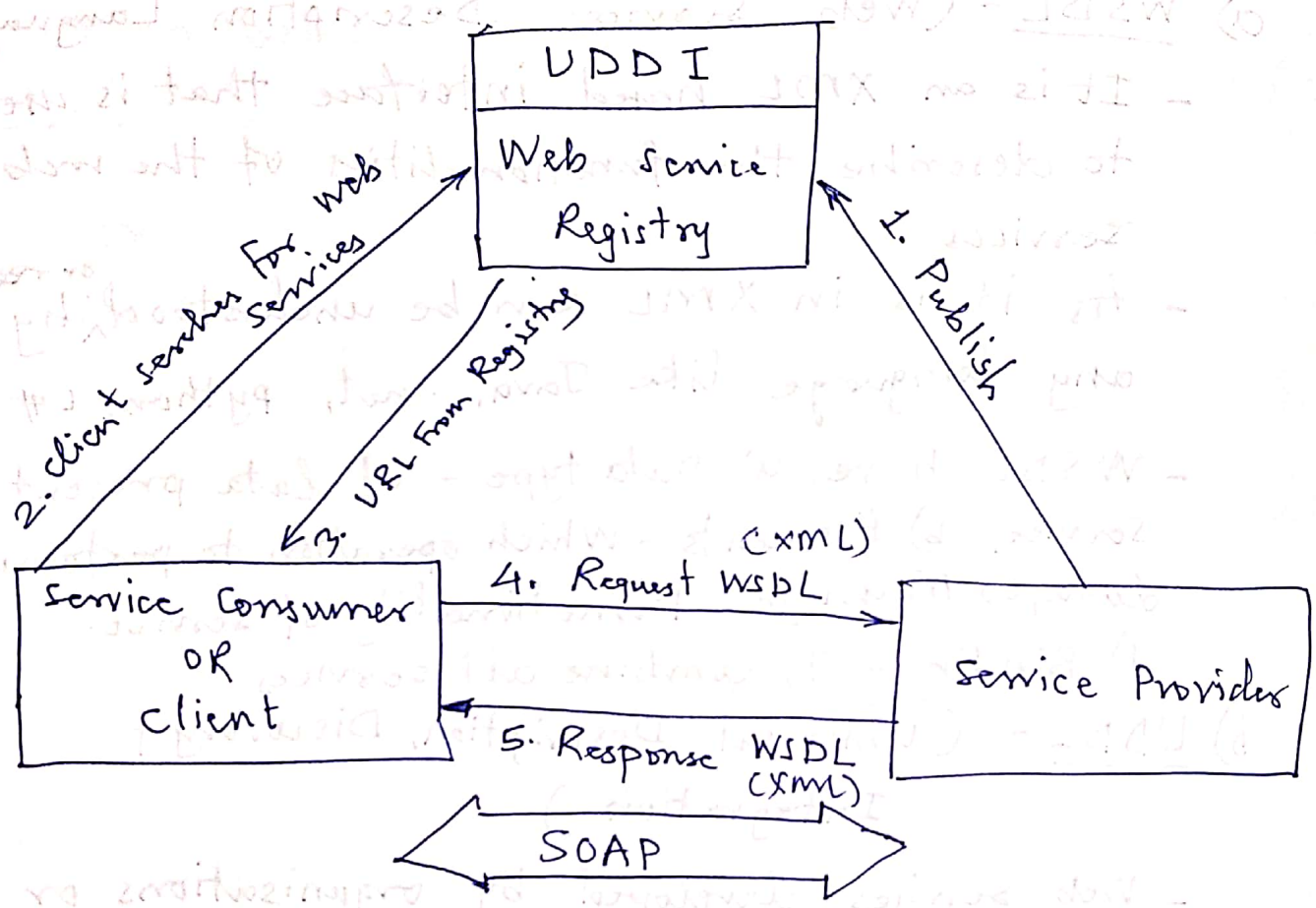
a) WSDL - (Web Services Description Language)

- It is an XML based interface that is used to describe the functionalities of the web services.
- As it is in XML can be understood by any language like Java, .net, python, C#, etc. or read.
- WSDL have. a) Data type - of data present in service, b) Elements - Which operation to perform on data, c) Massage - Functionality of service. d) Binding - To combine all services.

b) UDDI - (Universal Description, Discovery & Integration.)

- Web services developed by organisations or individuals must be known to web service users. So there is need to register the web services so that businesses can discover the desired one.
- UDDI is that catalog which contains information about web services as well as the organisations where they are developed or offered.

- UDDI is plat-form independent & open.
- So every API has WSDL & this WSDL is stored at UDDI.
- We can say that UDDI is repository or file storage for WSDL.
- UDDI is a global web service directory that itself is implemented as a web service.



→ Types of Web services :-

a) SOAP (Simple Object Access Protocol)

b) REST (REpresentational State Transfer)

a) SOAP -

- SOAP is a protocol which was designed before REST.
- The main idea behind SOAP was to ensure that programs built on diff. platforms & programming languages could exchange data in an easy manner.
- Only we can use ~~"GET"~~ "Post" web service request

b) REST -

- This is designed specially for working with components such as mediia components, files on a particular hardware device
- We can use GET, POST, PUT, PATCH & DELETE web service requests.

SOAP	REST
- Simple ~~Acces~~ object Access Protocol - SOAP services used for web-based application only - SOAP is based on XML	REpresentational State Transfer. - REST services used for web based, Mobile based, Desktop based applications - REST uses HTTPS, JSON, URL & XML

SOAP	REST
1) SOAP - Simple Object Access Protocol	1) REST - RE presentational State Transfer
2) SOAP services are used for web-based applications only	2) REST services are used for web based, mobile based & desktop based applications
3) SOAP services use XML for request as well as response.	3) REST services use HTTP/ URL for request & HTML, Text, JSON, XML for response.
4) For Testing SOAP services WSDL file is required	4) For Testing REST services we require URL
5) SOAP support SSL security	5) REST support SSL & HTTPS security
6] SOAP is a protocol	6] REST is an architectural style.
7] SOAP requires high bandwidth & response time is also more	7] REST requires fewer bandwidth & response time is less (faster

Types of Requests / Methods :- (HTTP Methods)

1] GET Method :- GET is used to get data from a resource - similar to select statement

2] Post Method :- POST is used to send data to a server to create a resource.
- similar to Insert into statatement

3] PUT Method :- PUT is used to send data to a server to update a resource.
- similar to update statement
- Here full update is there.

4] PATCH Method :- PATCH is used to send partial data to server to update.
- similar to update statement
- Here partial update is there.

5] DELETE Method :- DELETE is used to delete the specified resource
- Similar to delete statement

— x — x — x — x — x —

GET → Read data

POST → Create

PUT → upadate

PATCH → partial update

DELETE → delete

What is CRUD ?

- Most of APIs implements CRUD.

C → Create → POST
R → Retrieve or Read → GET
U → Update → PUT / PATCH
D → Delete → DELETE.

Realtime examples of HTTP methods :-

1] GET - Check Account balance. (GET Request)

2] POST - Create UPI for Account (POST Request)

3] PUT - Update UPI (PUT Request)

4] DELETE - Delete Account (DELETE Request)

5] PATCH - Single update in UPI (PATCH Request).

Diffent HTTP status codes :-

1] Successful Responses (2xx)
2] Server errors (4xx)
3] Client Errors (5xx)

1] Successful Responses (2xx) :-

a) 200-OK :- A well functioning URL will respond with a 200 status code.
- The request has succeeded & response body will be present.

b) 201- Created :- The request has been succeeded & a new resource has been created as a result.
- This is typically the response for Post or some PUT requests.

c) 202:- Accepted :- If data is sent to server for storing purpose & get accepted by server.
- PUT / PATCH Request.

d) 204- No Content :- The request is successful but there is no response body.

2] Client Errors (4XX) :-

a) **400 - Bad Request** :- If the Json formate is not proper in request in that case we get bad request.

b) **401 - Unauthorised** :- Generally it comes with non-registered users tries to hit any api.

c) **402 - Payment Required** :- If any api which requires payment and payment not done.

d) **403 - Forbidden** :- User is authorised but not having auess to do some operations.

e) **404 - Not Found** :- If you try to auess URL which ~~is no~~ doesnot exists.

f) **405 - Method Not allowed** :- If we want to update the resourse but by mistake instede of PUT we apply POST Method then we get Method not allowed error.

g) 408 - Request Timeout :- Server is expecting to hit the API within some time but didn't hit. (eg. atm Machine).

3] Server side errors (5xx) :-

a) 500 - Internal server error :- Server cannot proces the request for unknown reason.

e.g. a) Server misconfigured or missing package.

b) Some malware done some changes in code.

b) 501 - Not Implemented :- The server does not support the functionality required to fulfill the request.

c) 502 - Bad Gateway :- The server is acting as a Gateway or proxy server & it is not receiving proper response from backend server.

a) To connect the two servers socket is required & if socket is not at it's location then we get Bad Gateway.

d) 503 - Service Unavailable :- The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

e) 504 - Gateway Timeout :- Server not recieving response within specified time.

e.g. a) Network connection beth servers are weak
b) The backend server is too slow.
c) Time out duration set is too short.