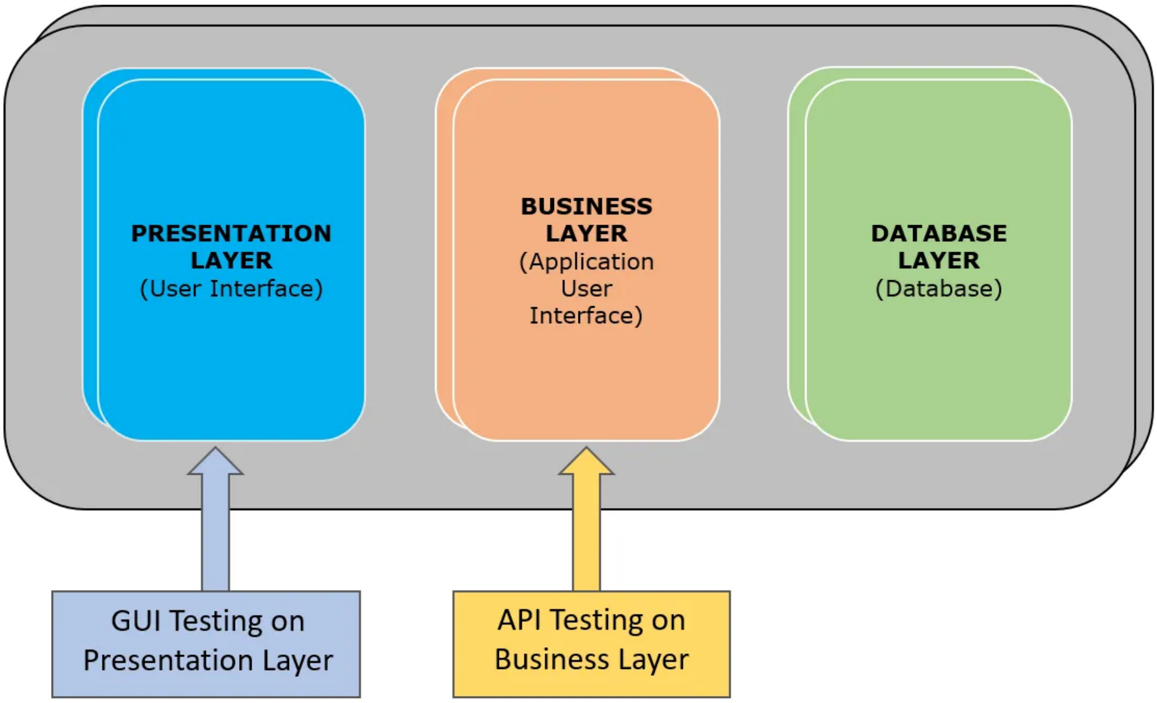
API

● API stands for Application Programming Interface.
● API is a set of routines, protocols, and tools for building Software Applications.
● APIs specify how one software program should interact with other software programs.
● Or We simple word we can say that communication between the two different software

API Testing:

● API testing is a type of software testing that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application.
● In API Testing our main focus will be on a Business logic layer of the software architecture.
● API testing can be performed on any software system which contains multiple APIs. API testing won't concentrate on the look and feel of the application.
● API testing is entirely different from GUI Testing.


PRESENTATION LAYER
(User Interface)
BUSINESS LAYER
(Application User Interface)
DATABASE LAYER
(Database)
GUI Testing on Presentation Layer
API Testing on Business Layer

Advantages:

- API Testing is time effective when compared to GUI Testing. API test automation requires less code so it can provide faster and better test coverage.
- API Testing helps us to reduce the testing cost. With API Testing we can find minor bugs before the GUI Testing. These minor bugs will become bigger during GUI Testing. So finding those bugs in the API Testing will be cost-effective to the Company.
- API Testing is language independent.
- API Testing is quite helpful in testing Core Functionality. We can test the APIs without a user interface. In GUI Testing, we need to wait until the application is available to test the core functionalities.
- API Testing helps us to reduce the risks.

What exactly needs to be verified in API Testing?

Basically, on API Testing, we send a request to the API with the known data and we analyse the response.

- Data accuracy
- HTTP status codes
- Response time
- Error codes in case API return any errors
- Authorization checks
- Non-functional testing such as performance testing, security testing

REST Vs SOAP

SOAP	**REST**
Simple Object Access Protocol	Representational State Transfer
SOAP supports XML format only	REST support Plain Text, XML, JSON etc
More Secure than REST	Less Secure as compare to SOAP
Heavy weight as it requires more resources and bandwidth	Light weight as it requires less resources and bandwidth
Low Speed	High Speed
SOAP defines standards to be strictly followed	REST does not define too much standards like SOAP

HTTP Methods:

1.GET
2.POST
3.Patch
4.Put
5.Delete

1.GET Method:

- ***HTTP GET*** method used to retrieve information from the REST API.
- We should not use this method to change any information or the resource

2.POST Method:

- This method in the *REST API* should only be used to create a new resource.

3.PUT Method:

- If we want to update an existing resource,PUT method should be used for this operation

- If we want to update more fields or all fields from existing data then we will use PUT method

4.Patch Method:

- Partial update to a resource should happen through the PATCH method.
- When we want to update particular field then we will use patch method

5.Delete Method:

- DELETE method should be used to remove a given resource

Status Codes

● Status Codes are issued by a server in response to a client's request made to the server.

● Status Code is 3 digit.

● Http response code grouped into five classes.

1. **1xx(100 to 199):Informational responses**
2. **2xx(200 to 299):Successful responses**
3. **3xx(300 to 399):Redirectional Messages**
4. **4xx(400 to 499):Client error responses**
5. **5xx(500 to 599):Server error responses**

Below are some important status codes that are required while doing API testing.

1.200: OK

● The request succeed.

● The client can read the result of the request in the body and the headers of Response

2.201: Created

● The request succeeded, and a new resource was created as a result. This is typically the response sent after POST requests, or some PUT requests.

3.202:Accepted

- The request ha been accepted for the processing but the process has not been Completed

4.204:No Content

- The request has succeeded but returns no message body.

5.400:Bad Request

- The request could not be understood by the server due to syntax error

6.401:Unauthorized

- Although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response.

7.404:Not Found

- The server can not find the requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist
- This response code is probably the most well known due to its frequent occurrence on the web.

8.405:Method Not Found

- The request method is known by the server but is not supported by the target resource. For example, an API may not allow calling DELETE to remove a Resource.

9.429:Too Many Requests:

- The user has sent too many requests in a given amount of time ("rate limiting").

10.500:Internal Server Error

- The server has encountered a situation it does not know how to handle.

11.503:Service Unavailable

- The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded.

- Note that together with this response, a user-friendly page explaining the problem should be sent.
- This response should be used for temporary conditions and the Retry-After HTTP header should, if possible, contain the estimated time before the recovery of the service

12.504:Gateway Timeout

- This error response is given when the server is acting as a gateway and cannot get a response in time.

API Request Is Break Down in two parts
1.Base URL
2.END Point

Example:
URL:https://reqres.in/api/users?page=2
Base URL:https://reqres.in
End Point:/api/users?page=2

NOTE: For End Point some people refer as Tail URL