d

SQACIRCLE
CONNECT AND STEP INTO THE FUTURE

API Testing

- **Introduction to API**
- **What is an API**
- **What is API Testing**
- **Why is API Testing Important**

Created By: IQRA ILYAS

Introduction to API:

What is an API:

Acronym of API is **Application Programming interface**. An API allows to applicants to talk to each other. **Interface**: It allows Communication between two separate things.

It is defined as a set of functions, methods, guidelines which allow communication and exchange of data between various software components.

Why are API Useful:

Save effort as developers can make use of available functionality without writing code themselves.

Enhance an application by adding other application features.

Allows user to access data from other application.

What is API Testing:

It is simply a type of software testing where we are validating several aspects of an API.

Aspects of API:

- **Verify Functionality:** Verifying functionality of an API which means that we are validating that API is working as its expected too.

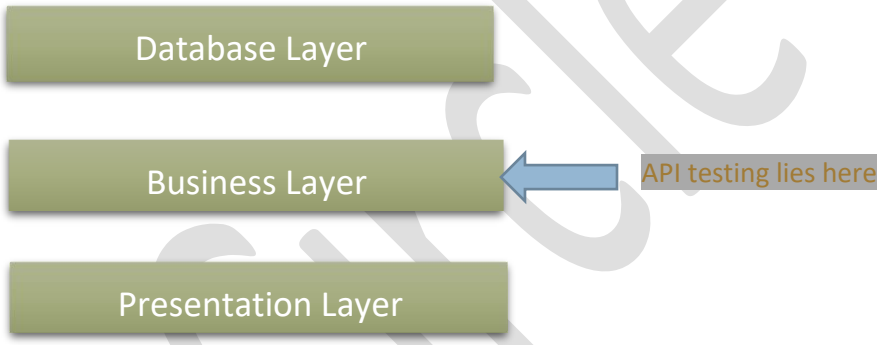
Example: If we request some data from server then in response, we get data in some format like JSON Format which is our response body and we get response code of 200.

- **Verify Reliability:** Verify reliability by ensuring that API is producing consistent results. We could have different environments for testing and we may specify a period of time for testing. So, we nee to ensure that API we are testing is functioning consistently across all different environments and for that specified period of time.

- **Verify Security:** We also verify security of an API by validating access controls and ensuring that our API endpoints are secure.

- **Verify Performance:** We verify performance of an API by generating load on our server to check if API is functioning and are not taking too long to respond or load on the server is not cause application to crash.

3 Layers of Application:



Presentation Layer:

This is also known as UI Layer. This layer is where we make use of elements of user interface like text inputs and buttons to interact with an application. This layer is visible to user.

Business Layer:

This layer contains all of the business logic of the application. Whenever we perform any action like if we want to post some data to the server.

Example: We are filling and submitting a sign-up form after we click on submit, this layer takes all of our data which is entered and processed that data.

Database Layer:

This layer contains all of data related to application.

Note: Business Layer connects the presentation layer to business layer.it takes an input from the presentation layer, which is entered by the user, and then it communicates this input to database layer.

Why is API Testing Important:

Find defects at early stage of development.

1. Focus and test core functionality before running GUI tests.
2. Exposes defects which can be fixed at an early stage.
3. Save effort when GUI is tested later.

Faster than UI Tests.

1. API test take much less time than GUI tests.
2. Saves a lot of time and effort.

Language independent:

1. Data is exchanged in JSON or XML format.
2. These formats allow any language to be used.

Reduced Cost:

1. Early detection and fixing of defects.
2. Faster test execution.

SQACircle

Section 2:

- **Types of Web API.**
- **API Request.**
- **API Response. • HTTP Methods.**

Types of Web API:

- Rest
- SOAP
- GraphQL

REST API:

1. Acronym for representational state transfer.
2. Architectural pattern for API's.
3. When an API request for a resource is made, what we get in response is the state of the resource in some form of representation JSON.
4. XML and HTML can also be used, but they are not as common as JSON.

SOAP API:

1. Acronym for simple object access protocol.

2. Protocol which uses XML for communication among applicants.

GraphQL:

1. Data query Language.
2. Allows you to specify what data you want.
3. Only data which is specified in request query is returned instead of the whole data. **Data Formats:**

Format is used to represent structed data. There are two data formats commonly used.

- **JSON:** JavaScript object Notation. Data is shown as object.

  Example:
  {
  Username : "Iqra Ilyas", Password :
  "iqra.113"
  }
- **XML:** Extensible markup language. XML looks like HTML, but in XML format we define our own tags.

  Example:

  <message>

  <warning>

  HELLO WORLD

  </warning>

  </message>

API Request:

Request Components:

HTTP Methods:

HTTP Method of an API determines what operation do we need to perform using an API even communicating with an application.

Some basic HTTP Methods are:

- GET
- POST
- PUT

- DELETE

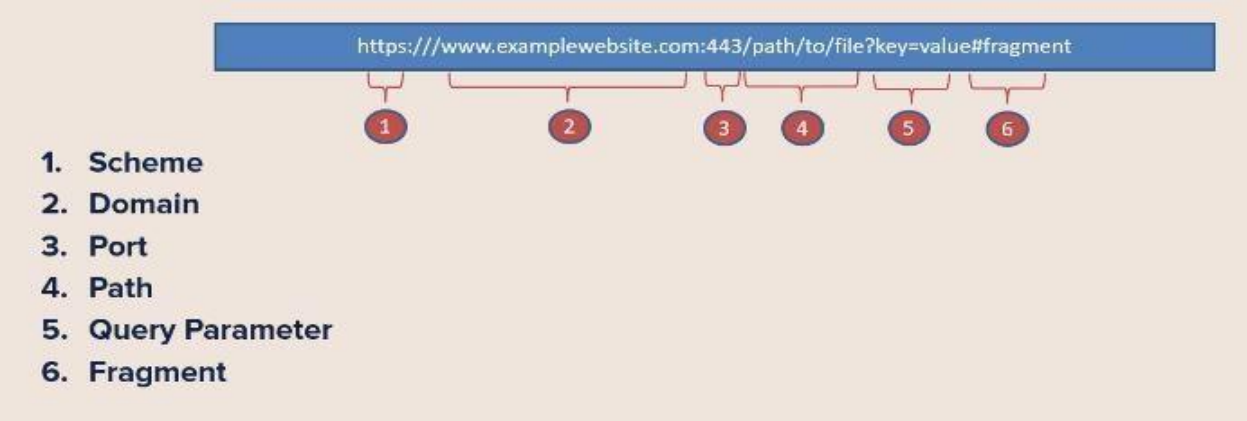
If we want to perform CRUD Operations like Create, Read, Update and delete we will make use of HTTP Methods to perform these operations.

1. **C** For creating data we would make a **POST Call.**
2. **R** For reading data we would make a **GET Call**.
3. **U** For updating data we would make a **PUT Call**.
4. **D** For deleting data we would make a **DELETE Call**.

URL:

URL stands for Uniform resource Locator. It is an address of accessing a resource on the web. So, when you type URL on your browser. You are basically requesting for particular webpage.

URL Structure:



1. **Scheme:** It is the protocol which must be used in for requesting for data. For websites its usually HTTP or HTTPS. Difference between the two is that HTTPS is secured while HTTP is unsecured.

2. **Domain:** This domain name specifies our web server to which we initiate the request. You can also used IP Address of the web server instead of domain name, but the use of domain name is more common and it is easier to remember domain names than IP Address.

3. **Port:** port is specified after the colon in the domain name or IP Address and it is used to determine that which process or specific service the incoming request and traffic should be forwarded to. For HTTP if it is not specified then by default it will be determined as 80, and for HTTPS it is 443.

4. **Path:** It specifies the specific resource that you want to access on the application. It is very important to specify path when making API request as when we get particular resource or add data under some resource.

5. **Query Parameter:** Not all URL has query parameters. Query parameters are used for creating for specific data and getting relevant data according to the data provided in return. When you type something on google search then your text input is passed as query parameter.

6. **Fragment:** Not all URL has fragments. If you provide browser a URL with fragment which is associated with particular heading or subheading, it will not only open the web page for you, it will also scroll you to exact heading which is specified in the fragment.

Request Body:

It is also called as request payload. It typically required when we using a POST API Call to create some data or PUT API Call to update some data. In these cases, we provide data to server for creating or updating purposes.

Request Headers:

It is an additional information related to an API request. It can contain things like authorization, that if an API is secured it would require authorization.

API Response

When we are communicating with application using API's we are on the client end and we initiate a request to perform a particular operation. And then server upon receiving a request gives an appropriate response to that request. Response also has a structured.

Response Components:

- **Status Codes**

200 It means that request has succussed.

400 It means that some error in request.

500 Internal server error issue with server.

Response Code Groups:

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

1. Informational responses (100 – 199)
2. Successful responses (200 – 299)
3. Redirection messages (300 – 399)
4. Client error responses (400 – 499)
5. Server error responses (500 – 599)

- **1XX – Informational**
  100-101 etc…

100 means Continue.

101 means Protocol.

It means that server has received the request and is continuing the process.

- **2XX – Success** 200-201 etc…

200 means OK.
201 means Created.

If your request is successful server will respond with 200 OK.

If you send POST request to server to create data under given resource and instead request is successful the server will respond with status code 201.

- **3XX – Redirection**
  301-302 etc…

301 means Permeant redirection. (URL of resource changed permanently).
302 means Redirection (resource which we are going to access moved temporarily to another location).

If you send a request for resource and that resource exit elsewhere, then you would get redirected by the server.

- **4XX – Client Error** 400, 401, 404 etc…

400 means Bad Request.
401 means . Unauthorized
404 means Not Found .

When you send API request to server incase request is not correct. Server will respond with error code.

400 error occur when a request contains some sort of syntax error.
401 error occur if an API request requires some sort of authorization. 404 error occur when you are provides wrong path on the URL for resource.

- **5XX – Server Error** 500-502 etc…

500 means internal server error .
502 means . bad gateway server error

500 error occur when server has encountered an unexpected issue and cannot handle the request.

502 error occur when server is acting as a gateway or proxy receives an invalid response from the server.

HTTP Methods

GET

- Used for requesting a resource from the server.
- Cannot be used to update data. • No request body.

POST

- Used for adding resources to the server. • Data to be added is sent on request body.

PUT

- Used for modifying resources on server. • Data to be modified is sent on request body.

PATCH

- Also used for modifying resources on server, however it is different from PUT call as it is only used for partially modifying a resource.
- Unlike request payload in PUT Method, in PATCH method we only need to send data we want to change.

DELETE

- Used for deleting a resource on the server.
- No request Body.

.....THE END.....

SQACircle

THANK YOU

SQACIRCLE

CONNECT AND STEP INTO THE FUTURE

Contact: +923164363302
Email: sqacircle@gmail.com
Website: sqacircle.com