{ api }

TheTestingAcademy

API Testing Using Postman

Mastering API Testing.

POSTMAN

Scrolltest.com

About Me

5 Years into Software Testing and Test Automation Field.

@promode_data

techdutta

Website : https://scrolltest.com
Website : https://thetestingacademy.com

pramoddutta@live.com

Let’s Start Learning…..

{ api }

What you'll learn..

POSTMAN

- Advance your software testing abilities by learning API Testing.
- Good understanding of the APIs & API Testing(HTTP Basics).
- How to test API using the POSTMAN(Chrome extension and monitor them using Jenkins)
- How software tester in the company test API and validation used by them
- Web Fundamentals & Understanding Continuous Integration Delivery in Big Companies.

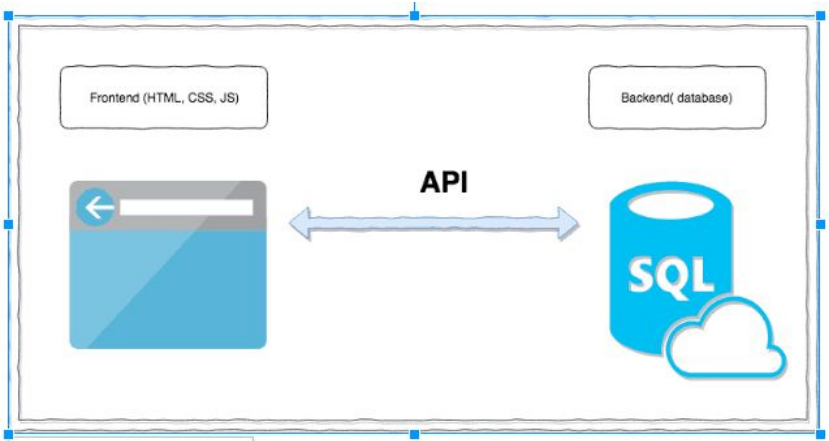
{ api }

In this Tutorial...

- What is an API?
- What is API Testing?
- SOAP vs REST.
- What are common Types of Bugs in API Testing?
- REST Explained.
- Installing Postman

What is an API?

- API stands for the Application Programming Interface,
- They are basically a collection of functions and procedures which allows us to communicate two applications or libraries.
- In short, It is like a **connector** between two services as shown in the picture.

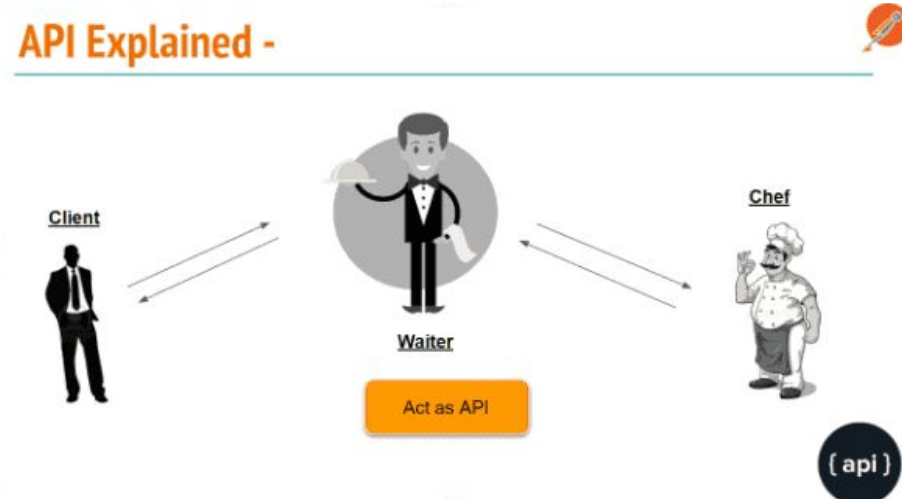


In one line, API is its an interface between different software programs or service.

What is an API?

Suppose you go to a restaurant.

API is the messenger(waiter) that takes your order from you and tells to the chef in (kitchen), what food to be prepared and after some time waiter returns with the ordered food.



In one line, API is its an interface between different software programs or service.

Type of APIs :-

- SOAP
- JSON RPC
- XML RPC
- REST

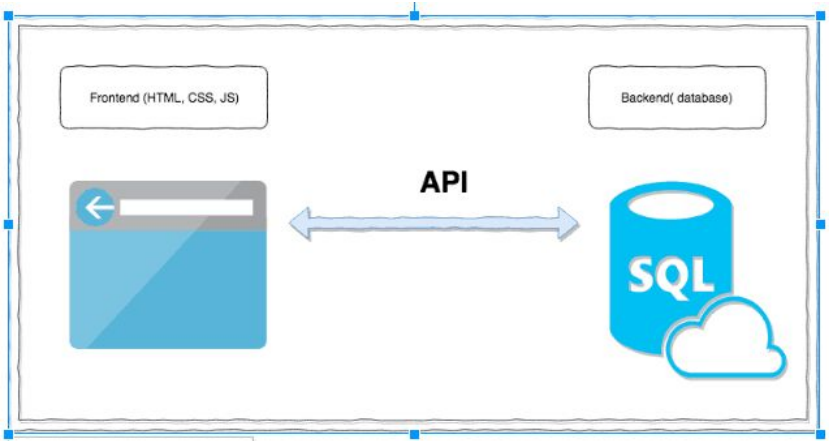
We are only Concern about the Web API

Simple Object Access Protocol
Remote Procedure Call
Representational State Transfer (REST).

etc.....

What is API Testing?

- API testing is testing that APIs and its integration with the services.
- It is one of the most challenging types of testing If we miss the certain cases in API Testing that can cause a very big problem in production after full integration and it will hard to debug in the production environment...



{ api }

Why you should perform API Testing?

- Many of the services that we use every day rely on hundreds of different interconnected APIs, if any one of them fails then the service will not work.
- Right now, Internet uses millions of APIs and they should be tested thoroughly.
- Developers make mistake and they create buggy APIs…
- Validation of APIs is very important which are going live to production.

{ api }

What to Test in API Testing?

- Validate the keys with the Min. and Max range of APIs (e.g maximum and minimum length)
- Have a test case to do XML, JSON Schema validation.
- Keys verification. If we have JSON, XML APIs we should verify it’s that all the keys are coming.
- Verify that how the APIs error codes handled.

{ api }

What is SOAP?

SOAP (Simple Object Access Protocol)

It is a messaging protocol that allows programs that run on disparate operating systems or services like frontend or backend to communicate using Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML).

SOAP Example

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/1999/XMLSchema">
    <SOAP-ENV:Body>
     <ns1:sayHelloToResponse
      xmlns:ns1="Hello"
      SOAP-ENV:encodingStyle="
      http://schemas.xmlsoap.org/soap/encoding/">
          <return xsi:type="xsd:string">Hello John, How are you
               doing?
          </return>
     </ns1:sayHelloToResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

{ api }

What is REST?

As REST is an acronym for

REpresentational State Transfer, statelessness is key. An API can be REST if it follows the below constraints.

- The REST architectural style describes six constraints. These constraints, put on the architecture, were initially communicated by Roy Fielding in his doctoral dissertation and defines the basis of RESTful-style.

- Uniform Interface
- Stateless
- Cacheable
- Client-Server
- Layered System
- Code on Demand

{ api }

SOAP vs REST

#	SOAP	REST
1	A XML-based message protocol	An architectural style protocol
2	Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
3	Invokes services by calling RPC method	Simply calls services via URL path
4	Does not return human readable result	Result is readable which is just plain XML or JSON
5	Transfer is over HTTP. Also uses other protocols such as SMTP, FTP, etc.	Transfer is over HTTP only
6	JavaScript can call SOAP, but it is difficult to implement	Easy to call from JavaScript
7	Performance is not great compared to REST	Performance is much better compared to SOAP - less CPU intensive, leaner code etc.

Who Users REST APIs

Who Uses REST

- Twitter API
- LinkedIn API
- Slack API
- ..... many more.....

REST Constraints Explained. - Uniform Interface

The first constraint of the REST API states that the Client and server has to communicate and agree to certain rules based on resources(they should communicate with same resource like json, xml, html , txt) and with proper encoding like UTF-8 extra.

REST Constraints Explained. - Stateless

- APIs in REST is stateless and Client and server don’t worry about the state of the request or response..
- Stateless means the server does not remember anything about the user who uses the API.
- It doesn’t remember if the user of the API already sent a GET request for the same resource in the past

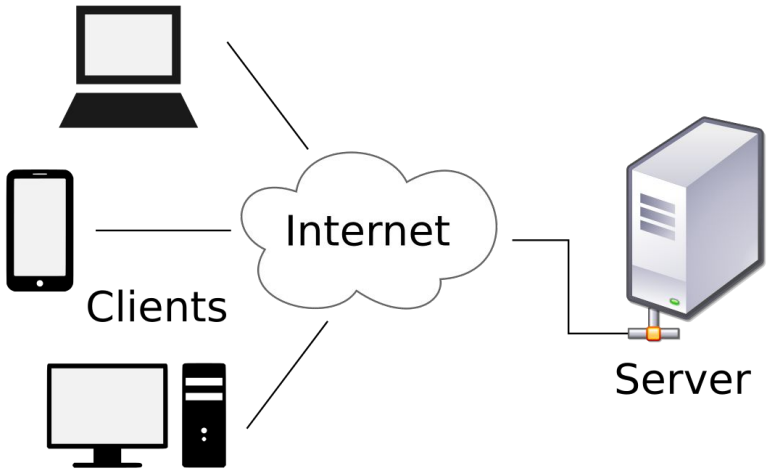
{ api }

REST Constraints Explained. - Cacheable

- According to the World Wide Web, clients can cache responses. Responses should, therefore, implicitly or explicitly, define themselves as cacheable. It’s up to server when they want the cache to expired etc.
- This means that the data the server sends contain information about whether or not the data is cacheable.
- If the data is cacheable, it might contain some sort of a version number. The version number is what makes caching possible

REST Constraints Explained. - Client-Server

- Client and Server are two different entity, It means that servers and clients may also be replaced and developed independently, as long as the interface is not altered.



REST Constraints Explained. - Layered System.

It means that the between client and server there can be any number of layered systems it does not matter.

REST Constraints Explained. - Code on Demand

The server can store the Code or logic to themselves and transfer it whenever needed rather client-side logic.

Let’s Install Post and Basic Overview.

Click to Subscribe

Scrolltest.com

TheTestingAcademy.com

Thanks for Watching...

Next...Understanding HTTP Methods for API Testing.