**Rest API Testing**

Introduction to API

What is an API? The term API is an acronym, and it stands for “Application Programming Interface".

An API is a particular set of rules ('code') and specifications that software programs can follow to communicate with each other. It serves as an interface between different software programs and facilitates their interaction, similar to the way the user interface facilitates interaction between humans and computers.

**Use Case 1:** Consider an API as a waiter in a restaurant. Suppose you have a menu of your favorite food and the kitchen is the system where your order is made. But how do you take your order till the kitchen? Correct, you call a waiter, give him/her the order, which in turns takes your order till the kitchen and then your order is made there and then finally, the waiter comes back with your delicious ordered food. Thus, the API is very much similar to the waiter. API is the messenger that takes your order(waiter) and tells the system(kitchen) what to do (to prepare food) and in return gives back the response you asked for (waiter returns with the ordered food). To get more clear understanding, lets shift to another example that will clear your remaining doubts too.

**Use Case 2:**Whenever you book a flight ticket, you go to various websites like MakeMyTrip , GoIbIbo , Yatra, etc. You hardly ever go to specific airlines website to book a ticket. Still, you get the plane at the same Air Plane service that you might have booked through the airlines website. So ever wondered how did this happen? Yes, Again, it’s because of the API.

For example, when you search for flights in makemytrip, the site sends a search request/API call which includes the necessary details such as the date of journey, destination, number of passengers....etc to an API which is hosted by the airline or a third party.  In response, the API sends back the response which contains the flights available on that date to makemytrip.

**What is a Web Service?**

A Web service is an API that facilitates the communication between two applications over a network. Web service implementation allows two web applications developed in different languages to interact with each other using a standardized medium like XML, SOAP, HTTP etc.

Web Services Architecture

**Features of web services**

 As web services are based on open standards like XML, HTTP so these are operating system independent

 Likewise, web services are programming language independent, a java application can consume a PHP web service

 Web services can be published over the internet to be consumed by other web applications

 The consumer of web service is loosely coupled with the web service, so the web service can update or change their underlying logic without affecting the consumer

**Types of Web Services**

**1. SOAP Web Service**

SOAP stands for Simple Object Access Protocol, it is a standardized protocol for message exchange between web applications. The message format supported by SOAP is XML. A web service that is based on SOAP protocol is called SOAP web service.

**2. RESTful Web services or REST API**

REST stands for Representational State Transfer, it is an architectural style that describes some constraint for web service development. A web service that satisfies these constraints is called RESTful web service.

The **six**REST architecture constraints are

**Client-Server** – Client and server are separated by a uniform interface and are not concerned with each other’s internal logic

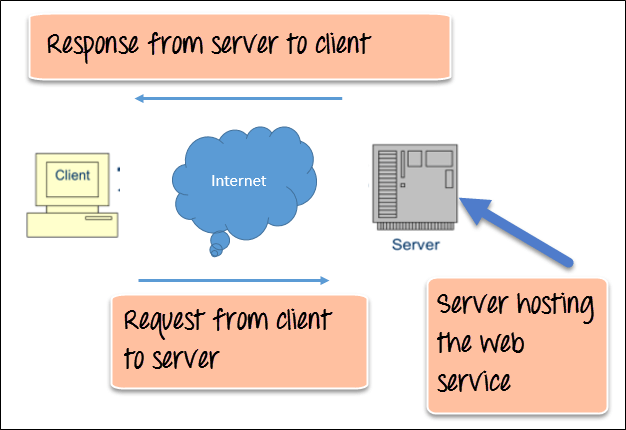
**Stateless**– Each client request is independent and contains all the necessary information required to get executed. No client data is saved at the server.

**Cacheable**– Client should have the ability to cache the responses

**Layered System**– A layered system having multiple layers wherein each layer communicates with adjacent layer only

**Uniform Interface**– A uniform interface design requires each component within the service to share a single and uniform architecture

**Code on Demand**– This constraint is optional. It extends client-side execution of code transfer of executable scripts like javascript from the server.



RESTful APIs implements the following types of HTTP methods

**GET**– HTTP GET method is used to retrieve some information  
**POST**– HTTP POST method submits and creates new resources  
**PUT**– HTTP PUT is used to update an already existing resource  
**DELETE**– HTTP DELETE is used to delete a resource

Introduction - REST API

By the end of the lesson, you will learn  
1. What is API?  
2. What is web service?  
3. What is Rest API?  
4. What is Soap based API?

API is nothing but predefined methods which are designed to utilize the business logic without touching actual code over secured network which are called as web services or API’s

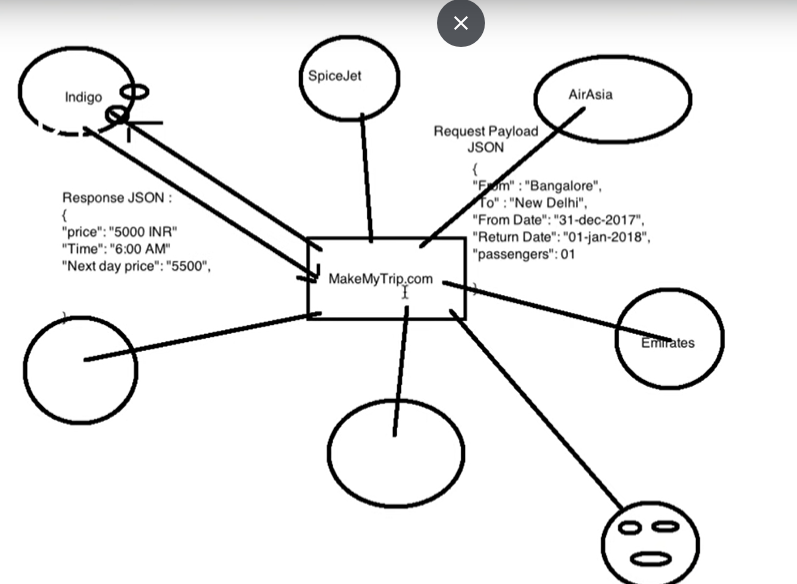
API establish a communication between two different applications to utilize a particular service

In API testing we will tests the business logic of the application which is also called as back end testing

In api business logic will not be visible, but we can utilize the business logic and only client can utilize with proper authorization. Because if the client directly interact with business logic there will be a chance of manipulation in the code.

In api’s the data will be transferred either in JSON or XML format only, because they are standard, light weight

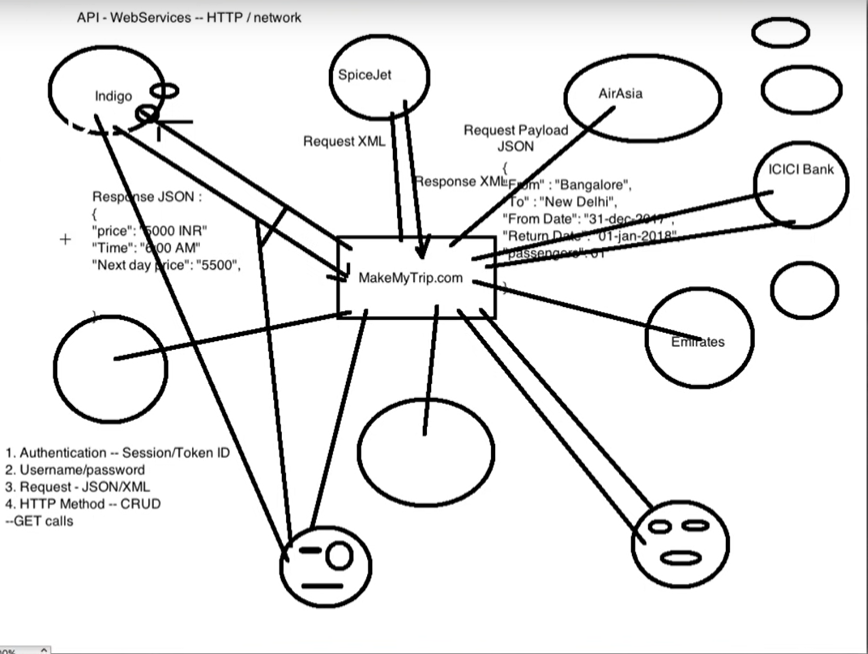
Even selenium also works with apis only. Selenium uses few services on the web service through the webdriver to communicate chrome driver

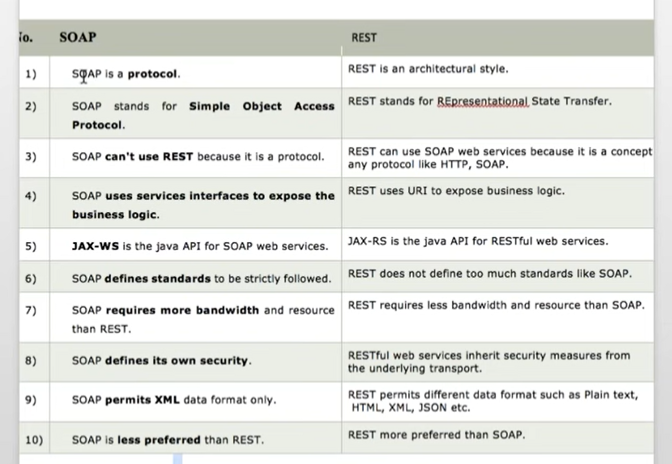


To prevent direct interaction with business logic in the application, application owner creates api by that way the client can utilize the business logic but they can not access the code. This is called webservice or API. The request is made from the client to the API over a secured network with the JSON/XML format to the Web application and that json is converted in the application to the application code format and process it and the response again shared to the client in the format of JSON/XML format

Operations in API:

CRUD 🡪 create, retrieve, update, delete





JAX-WS is the security system for SOAP, JAX-RS is the security system for REST.

CRUD:

POST:

* Used to create a new record/entity/entry in the server
* Need url, payload, autherization (in the format of json/XML)
* Will get success response

GET:

* Used to retrieve the existing records/entity
* Need url, headers, authorization

PUT:

* Used to create new entity as well as to update the existing update records
* Need url, payload, authorization

Delete:

* Used to delete the records
* Need url, payload, authorization

Create a record (POST)

Retrieve a record (GET)

Update a record (PUT)

Delete a record (Delete)

Status codes:

1 XX 🡪 informational 🡪 it means that request has been received and process is running

2 XX 🡪 Success 🡪 It means that action requested by the client was received, understood, accepted

200 ok 🡪 success, 201 🡪 created  
3 XX 🡪 redirection 🡪 It means further action must be taken in order to complete request

Ex: 302 🡪 to look into another url

4 XX 🡪 client error 🡪 it means the request contains incorrect syntax or cannot fulfilled

401 🡪 Un authorized 403 🡪 forbidden, 404 🡪 not found, 405 🡪 method not supported,

400 🡪 Bad request, 408 🡪 request time out, 429 🡪 too many requests

5 XX 🡪 Server error 🡪 It means served failed to fulfill

502 🡪 Bad gateway, 504 🡪 Bad gateway time out, 500 🡪 internal server error

504 🡪 Gateway time out , 503 🡪 service unavailable

REST client:

* As the API doesn’t have any UI we need a platform to test these API’s. So the client which we use to test these rest API’s will be called as REST clients
* Ex: POSTMAN, SOAP UI, Advance Rest client, Jmeter,

URI 🡪 URL (End point url) + Request API

Ex: https://reqres.in[/api/users/23](https://reqres.in/api/users/23) -🡪 URI

<https://reqres.in> 🡪 URL [/api/users/23](https://reqres.in/api/users/23) 🡪 Request API

Parameters:

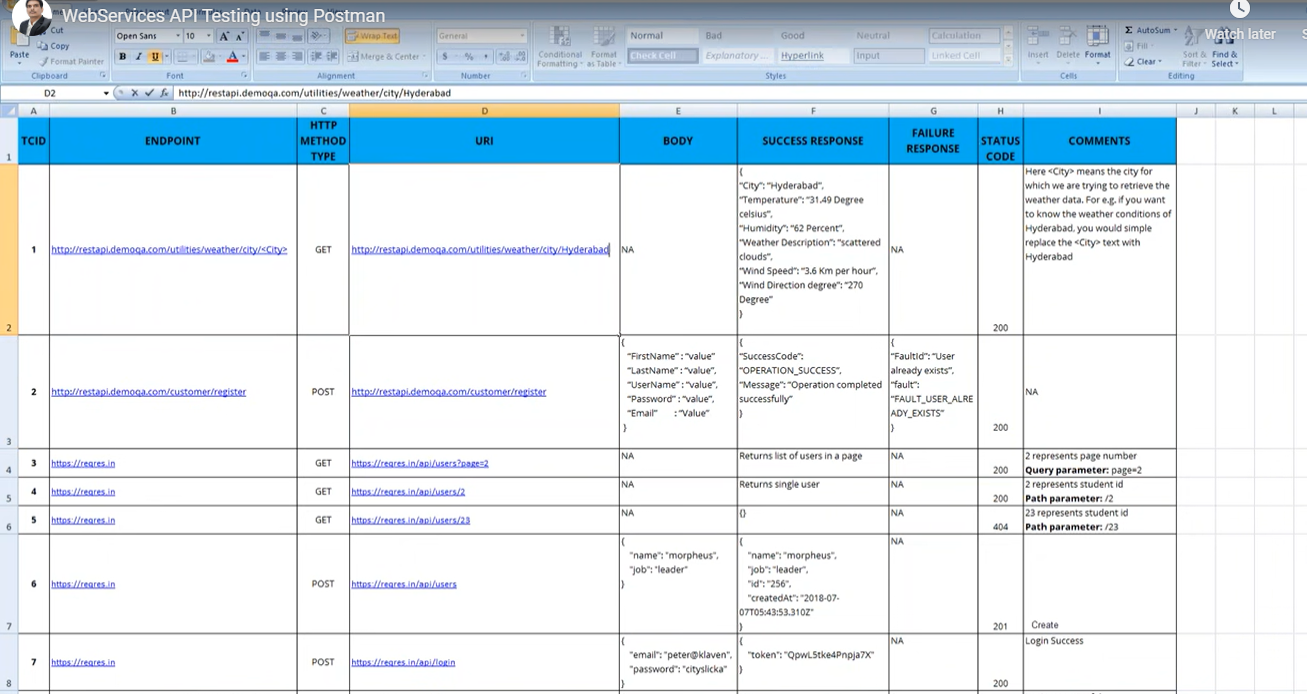
Path parameter 🡪 will always defined by /

Used to get the particular Data

Query parameter 🡪 will always defined by ?

Used to filter out the records in the response

Ex: <Http://services.groupkt.com/country/search?text=un>



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenraio | Test steps | URL | HTTP Method | Body | success response | failure response | status code | comments |  |
| To fetch all public repos in github | 1.select end point for get operation for git in git end point documentation  2.paste it in postman and select GET method and use autherization as Inherit from parrent option  3.check the response body for respositories | <https://api.github.com/repositories> | GET | NA | Status: 200 All repos & data should display | NO | 200 ok | Got success response |  |
| Create a repositary in [github.com](http://github.com/) in personal account | 1.click on new button in repository section 2,given a name to repo 3.select public access and create a repo | <https://github.com/dashboard> | NA | NA | NA | NA | NA | Got success response |  |
| To fetch all repos in the git hub account | 1.select end point for get operation for git in git end point documentation 2.paste it in postman and select GET method and use autherization as Inherit from parrent option 3.check the response body for respositories | <https://api.github.com/users/Mohanakasi/repos> | GET | NA | Status: 200 All repos & data should display |  | 200 ok | Got success response |  |
| Fetching the created respositary in github using postman tool | 1.select end point for get operation for git in git end point documentation 2.paste it in postman and select GET method and use autherization as Inherit from parrent option 3.check the response body for respositories | [https://api.github.com/repos/Mohanakasi/Kasi\_git\_practice\_repo](https://api.github.com/repos/Mohanakasi/REPO) | GET | NA | Status: 200 All repos & data should display |  | 200 ok | Got success response |  |
| Updating the created repository | 1.select end point for PATCH operation for git in git end point documentation 2.paste it in postman and select PATCH method and use autherization as Inherit from parrent option 3.check the response body for updated response code | [https://api.github.com/repos/Mohanakasi/Kasi\_git\_practice\_repo](https://api.github.com/repos/Mohanakasi/REPO) | PATCH | {  "name": "Mohana Kasi",  "description": "This is your first repository",  "homepage": "https://github.com",  "private": true,  "has\_issues": true,  "has\_projects": true,  "has\_wiki": true  } | Status: 200 or 201 created Updated messege should displsy |  | 200 ok or 201 created | Respoitory successfully edited |  |
| list all contributors for repository | 1.select end point for get operation for git in git end point documentation 2.paste it in postman and select GET method and use autherization as Inherit from parrent option 3.check the response body for contributors info | <https://api.github.com/repos/Mohanakasi/Kasi_git_practice_repo/contributors> | GET | NA | Status: 200  All Contributors data should display |  | 200 | got contributors list |  |
| list all languages | 1.select end point for get operation for git in git end point documentation 2.paste it in postman and select GET method and use autherization as Inherit from parrent option 3.check the response body for languages | [https://api.github.com/repos/Mohanakasi/Kasi\_git\_practice\_repo/languages](https://api.github.com/repos/OWNER/REPO/languages) | GET | NA | Status: 200 All languages data should display |  | 200 | No langauges showinh |  |
| Deleting the updated repository | 1.select end point for Delete operation for git in git end point documentation  2.paste it in postman and select PATCH method and use autherization as Inherit from parrent option  3.check the response body for updated response code | <https://api.github.com/repos/Mohanakasi/Kasi_git_practice_repo> | DELETE | NA | 204 |  | 204 | respository successfully deleted and got 204 ok messege |  |

 An API is essentially a contract between the client and the server or between two applications.

 API tests are fast, give high ROI, and simplify the validation of business logic, security, compliance, and other aspects of the application.

 While testing, verify correct HTTP status code**, v**erify response payload and  verify response headers.

 The major core **advantage** of **API testing** is that it provides access to application without users actually having to interact with a potentially disparate system. This helps the **tester** to detect and recognize the errors early, instead of them becoming larger issues during GUI **testing**.