SoapUI – Service testing tool for webservices and REST APIs.

Client – UI and Front end

Server – Back end and data base.

API – is an interface and play key role to of communication protocol between client and a server to simply the building of client-side software. To simplify to design process.

We use API to communicate between front end and back end. It is also beneficial when it comes to programing language is used for front end is different than back end. Which acts as interface between client and Server, also known has middleware layer between client and server. API is independent of any language. Front-end send all the information to API using http request protocol. We usually use Json or XML format to send information. They don’t belong to any specific language families. API send request to back end, back response back to API and API response back to front end using http protocol.

All Webservices are API. All APIs are not web services. Like in our lectures, Place and Library are Webservices but they can also be called APIs. And also like in JAVA, we have APIs (e.g methods) but they cannot be called web services.

Rest API is easy to build and most of the big tech companies using Rest API.

End point/BASE URL: Address where API is hosted on the server. Hitting the endpoint.

4 commonly used http methods to communicate with REST Api.

**GET, POST, PUT, DELTE – CRUD operations** create, retrieve, update, delete.

**GET:** when you want to retrieve some existing information from server. While using GeT request, it should only extract data and should have o other effect on data. No payload/body required. We use query parameters to send input data in GET.

**POST:** A post request is used to send data to the server, for example: customer information, file upload, etc. using HTML forms.

**PUT:** A put request is used to update **existing** information on server.

**DELETE:** Removes all current representation of the target resources given by a URI.

**Resources:** resources represent API/Collections which accessed from the server. Resource names need to be included in base URL. Like map/search/image shown below.

**Base URL**: google.com/map or google.com/search or google.com/map where APIs hosted.

There are two types of parameters, path parameter and query parameter.

**Path Parameters:** variable part of URL path, they are typically used to point to specific resource within a collection, such a user identified by ID. **Amazon.com/orders/112** <- path parameters also known as **sub resource**.

**Query Parameters**: is used to sort/filer the resources. Using existing resource. Always identified by **?** **Amazon.com/orders?sort\_by=2/20/2020**

<https://www.google.com/search?source=hp&ei=Ix_RX5DaHcur5NoP47SdkAc&q=new+york>

once we get the response from API then front-end developers read the response and generate html page. Front end data is generated from API response we get.

**End point request URL can constructed as below**

**Base URL/resource/(Query/Path)Parameter:** server name where your request should go to, then resource to point out the specific resource. IF Api demand to have parameters.

**Headers/Cookies:** Headers represent the meta-data associated with the API request and response. In layman terms, we were sending Additional details to API to process our request. **Example: Authorization details.** We are sending additional detail to API. API should know what format you are sending request, you have to tell API what type of format should we use.

https://docs.google.com/document/d/18FC3jDnsOol9zn3\_KGSrjg35a4unpiSG/edit

<https://docs.google.com/document/d/1A3Q_HX8A_GtamXs5kpdZ_7jM8W-OajKS/edit>

You need to get the API contact, BASE URL, Resource, parameters, http method, what is the json body, what does it look, what is the response you are expecting. Your developers should share all details with you.

{

    "status": "OK",

    "place\_id": "0f4b56ef5c25e8aed371828ebb6ff889",

    "scope": "APP",

    "reference": "a5b375c3a11bdcb6ce3fddbdef7b61cca5b375c3a11bdcb6ce3fddbdef7b61cc",

    "id": "a5b375c3a11bdcb6ce3fddbdef7b61cc"

}

1. A UR**I** is an **identifier** of a specific resource. Like a page, or book, or a document.
2. A UR**L** is special type of identifier **that also tells you how to access it**, such as HTTPs, FTP, etc.—like https://www.google.com.
3. If the protocol (https, ftp, etc.) is either present or implied for a domain, **you should call it a URL**—even though it’s also a URI.

How to parse complex JSON. Json like Nested json or Json contains Nested Arrays. Very important to know how to parse complex JSON.

Arrays is collection of different elements. Arrays always inside the square bracket [“Hello”, “Yellow”]

Mock Response: You will get a contract first, Developer will start working on API, we shouldn’t wait until development is done, parallelly we should be working on automation using mock/dummy response.