**Basic understanding of API**

When you enter some data in the front-end of the application. The data is sent to backend of the application through **http request** with a format i.e the data is either sent in JSON format or XML. The API acts as a middle tier in the application which is independent of any language and the API sends the data to the backend and gets the response from the backend to the frontend of the application with **a http response.**

**What is an API?**

An **Application Programming Interface** is a set of rules and protocols that allows one piece of software to interact with another. It defines the methods and data formats that applications can use to communicate with each other, enabling them to exchange information and perform specific tasks.

**Terminology to work with APIs**

**Endpoint / Base URL:** Address where API is hosted on the server.

HTTP methods commonly used to communicate with Rest APIs

GET, POST, PUT and DELETE.—These are also CRUD (Create,Retreive,Update,Delete) operations.

**GET:** This method is used to retrieve information from the server.

**POST:** This method is used to create some information in the server.

**PUT:** This method is used to update some information in the server.

**Delete:** This method is used to delete some information from the server.

**Resources:** Resources represent API /Collection which can be accessed from the server.

**Path Paramaters:** are variable parts of a URL Path. They are typically used to point to a specific resource within a collection, such as user identifier by ID.

Path parameters are identified by “/” slash.

**Query Parameters:** is used to sort or filter the resources. They are identified with ? mark.

The endpoint URL can be constructed as follows

BaseURL/resource/(Query/Path)Parameters

**Headers/Cookies:** represent the meta-data associated with the API request and response. In other words, we are sending additional details to API to process our request. For example, authorization details.

Why **Postman**?

Postman is the best API tool to develop and test API/webservices tests with very minimal code. It offers many readymade framework features like variables, environments, workflows, Data driven components and PM Objects which quickly helps to setup automation lab for testing.

Postman supports various types of services like Rest API´s, SOAP Web services, GraphQL Testing etc.

**Collection**: is a group of related requests, making them easier to access and run.

What is environment in Postman?

It refers to a setup in which the API is deployed and tested. This includes the combination of hardware, software, network configurations and other components necessary to create the conditions under which the API operates.

What are variables in Postman?

Variables allow you to store and reuse values in your requests and scripts. By storing a value in a variable, you can reference it throughout your collections, environments, and requests—and if you need to update the value, you only have to change it in one place. Using variables increases your ability to work efficiently and minimizes the likelihood of error.

Postman supports the following variable scopes:

* Global
* Collection
* Environment
* Data
* Local

If a variable with the same name is declared in two different scopes, the value stored in the variable with narrowest scope will be used. For example, if there is a global variable named “username” and a local variable named username, the local value will be used when the request runs.

Scripting in Postman

Postman contains a powerful runtime based on Node.js that allows you to add dynamic behavior to requests and collections. This allows you to write test suites, build requests that can contain dynamic parameters, pass data between requests,

Diagram

Description automatically generated

**Assertions:** An assertion is a check or a verification point that ensures the API behaves as expected. It involves comparing the actual response from the API to the expected result to validate the correctness and reliability of the API.