**API**

**1) What is API ?**

API is the abbreviation of the term **Application Programming Interface.** It is the software responsible for the connection for the communication and information exchange between two apps.

It is the piece of code that helps for communication between two Software or applications.

**Understanding API as an Example :**

The concept can be explained better with the help of an example. When you go to a restaurant and order food, the food comes from the kitchen. Hence the kitchen is the system. Your table is the place where the response is delivered. However, to deliver food on the table, there must be a channel of communication between the table and the kitchen. The waiter or API acts as a messenger between the table and the kitchen. It takes requests from you and tells the kitchen, which is a system, what has to be done. Then, the waiter delivers the response to the user, which is food in this case.

**API KEY :**

An **application programming interface** (API) key is a code used to identify and authenticate an application or user.

**API ENDPOINT :**

When an API interacts with another system, the touchpoints of this communication are considered endpoints. For APIs, an endpoint can include a URL of a server or service.

https://api.example.com/v1/users?role=admin&status=active

\\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\\_\_\_\_/ \\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

server URL endpoint query parameters

path

**SDK (Software Development Kit) :**

SDK is **a development kit that facilitates usages of an API**.

**## Programmable Web :**

It is the largest API directory where we can find thousands of API.

**API RESPONSE :**

The API supports 3 response types:

**JSON (Recommended)**

**XML**.

**NVP** (Deprecated)

**REST API**

**1) What is REST API ?**

A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services.

Basically it’s a standardized software architecture style which is well known.

## Understanding REST API Design

***Stateless***

REST APIs are stateless, meaning that calls can be made independently of one another, and each call contains all of the data necessary to complete itself successfully.

***Performance/caching***

As Service gets more complex the performance is high. Because a stateless API can increase request overhead by handling large loads of incoming and outbound calls, a REST API should be designed to encourage the storage of cacheable data.

**CRUD stands for Create, Read, Update, and Delete**, which are four primitive database operations. At first glance, these operations map well to the HTTP verbs most frequently used in REST:

Create (SQL INSERT) : POST - Used to support the creation of a child resource, but can also modify the underlying state of a system.

Read (SQL SELECT) : GET - Retrieve a representation of a resource, but with additional semantics available.

Update (SQL UPDATE) : PUT - Update a resource using a full representation. Can also be used to create a resource. The full representation requirement is a large caveat, see the following.

Update (again) : PATCH - Update a resource using a partial representation.

Delete (SQL DELETE) : DELETE - Delete a resource. This is the best matched mapping

**REST request structure**

Any REST request includes four essential parts: an HTTP method, an endpoint, headers, and a body.

An **HTTP method** describes what is to be done with a *resource*. There are four basic methods also named CRUD operations:

* POST to Create a resource,
* GET to Retrieve a resource,
* PUT to Update a resource, and
* DELETE to Delete a resource.

An **endpoint** contains a *Uniform Resource Identifier (URI)*indicating where and how to find the resource on the Internet. The most common type of URI is a *Unique Resource Location*(URL), serving as a complete web address.

**Headers** store information relevant to both the client and server. Mainly, headers provide authentication data — such as an API key, the name or IP address of the computer where the server is installed, and the information about the response format.

A**body**is used to convey additional information to the server. For instance, it may be a piece of data you want to add or replace.

