**REST API -** [Meghana Prakash](https://medium.com/@meghanap1196?source=post_page-----698c230d2aa7--------------------------------) Feb 21, 2022

What is REST API?

**REST** stands for **“Representational State Transfer”**.

A picture containing text

Description automatically generated

* **Client** — Our application is requesting the data.
* **Dynamic Server/Cloud**—The server to which the client will be trying to access the data from.
* When a request is sent, the server, in turn, returns the data in either XML or JSON format.

**API** — This is an **Application Programming Interface** that helps to connect our application with the Dynamic Server/Cloud.

**Why is the data in XML or JSON format?**

It’s because the data must follow a particular structure.

Only the values/state of an object will be sent from the server-side to the client-side in JSON/XML format. [During this, there is transfer of state happening and hence called ‘State Transfer’.]

**Operations performed on websites -**

We can perform CRUD Operations and for this we will be using the HTTP methods.

Table

Description automatically generated

**example —**

<http://vidly.com/api/customers>

http/https — This can be either of them

vidly.com — Domain

API — convention to expose the RESTful services

customers — a collection of customers in the application (resource)

We can perform operations by sending an HTTP request to the endpoint.

Every HTTP request has something called a method that determines the type.

* **GET —**

GET /api/customers — This will return all the customer details

GET /api/customers/1 — This will return the customer details with id:1

* **PUT —**

PUT /api/customers/1

{ name: ‘’ }

We should also include the customer ID and also customer object in the body of the request to get it updated.

* **DELETE —**

DELETE /api/customers/1

To delete a customer.

* **POST —**

POST /api/customers

{ name: ‘’ }

We will be able to add a new customer, we have to include the customer object in the body of the request to create a new customer.

Note —

* Information can be stored by the client to prevent multiple calls.
* We can use REST when both the client and server operates on a web environment.

Example —

1. [LinkedIn API](https://developer.linkedin.com/)
2. [Twitter API](https://developer.twitter.com/en)