**REST API**

**Fundamentals**

(Representational State Transfer Application Programming Interface) is a set of rules for building & interacting with web services.

It uses standard HTTP methods like **GET**, HEAD, **POST**, **PUT**. **DELETE**, CONNECT, OPTIONS, TRACE & **PATCH** in order to access & manipulate resources.

**We will require the following 3, to view HTTP**

[1] A browser in developer mode: Control Shift I

[2] Registered Account with Postman API (An platform which allows developers to design, build, test and document, API’s

[3] cURL command line via CMD (Stands for client URL, developers use this command line to transfer data to and from a server)

**Contents Page**

**[1] Foundational Technologies**

[1.1] HTTP

[1.2] JSON

**[2] REST APIs**

[2.1] REST Concepts

[2.2] REST Implementation

**[1] Foundational Technologies**

[1.1] HTTP

**- HTTP; an application layer protocol in the internet protocol suite(group of protocols, which together design different ways things can communicate over the internet )**

**Originally HTTP was used for transferring hypertext documents (documents which contain hyperlinks to other resources) (e.g., HTML)**

**Currently as of now HTTP 1.1 is widely been used the most, to this date.**

**All in all, HTTP is a way in applications can communicate with one another, across the internet ( e.g., web browser communicating with a web server) (e.g., mobile app communicating with a service to get data). HTTP response &request is in the form of a text (cycle). Communication is stake less, so server forgets about request once a response is given (aka cycle complete).**

- cURL; an command line tool that developers use to transfer data to & from a server. It allows you to talk to a server via specifying an location (URL) & the data you need to send.

E.g.,

A black and white screen with white text

AI-generated content may be incorrect.

We get more text generated, its important to note our some of the symbols:

[1] \* Asterik Lines; These are messages generated by curl itself. They provide info on the connection process , DNS resolution and other internal operations; WHICH ARE NOT part of the HTTP request/response.

[2] > Greater than symbol; This represents HTTP request headers that curl sends to a server. (Client/your machine talking to a server)

[3] Smaller than symbol; This represents the HTTP response headers that the server is sending back to curl. (server talking to the client/your machine).

[4] HTML content via <!doctype html><html …; HTTP response body which shows HTML content on a page.

[5] Headers: e.g., Host: [www.google.com](http://www.google.com); these headers are used to help process the request correctly. E.g., Accept: \*/\*, means we are happy to accept anything back from the servers.

A black text on a white background

AI-generated content may be incorrect.A white background with black text

AI-generated content may be incorrect.- HTTP Status/Responce Codes (<https://http.cat/>);

- Browser vs Curl;

Web browsers typically perform more "GET" requests than curl because browsers parse and render web pages, which often include resources like images, stylesheets, and scripts that are linked within the HTML. curl, on the other hand, primarily fetches the content specified in its command and doesn't automatically handle embedded resources or page rendering.

- HTTP Methods/Requests:

* **GET: Retrieves a resource.**
* **POST: Sends data to the server to create a new resource or update an existing one.**
* **PUT: Replaces a resource with the provided data.**
* **DELETE: Removes a resource.**
* **PATCH: Partially modifies a resource.**
* HEAD: Similar to GET, but only retrieves the headers of a resource.
* OPTIONS: Retrieves the communication options for a resource.
* CONNECT: Establishes a tunnel to the server.
* TRACE: Performs a message loop-back test.