**HTTP Requests**

1. **HTTP**

HTTP, which stands for **Hypertext Transfer Protocol**, is used to structure requests and responses over the internet 🡪 requires data to be transferred from one point to another over the network.

There are many types of HTTP requests. The four most commonly used types of HTTP requests are GET, POST, PUT, and DELETE.

With a GET request, we’re retrieving, or *getting*, information from some source.

For a POST request, we’re *posting* information to a source that will process the information and send it back.

Chart

Description automatically generatedThe server will **ALWAYS** send a status code indicating the success or failure of the request.

e.g., when we use GET method from an URL (<https://google.com>), we will receive HTML for that website. Our browser is able to get these data, and convert them into the website that we are able to look at and interact with.

**HTTP is the command language that devices on both sides of the connection follow to communicate**.

1. **TCP**

**Chanel of communication between client & server is managed by TCP (Transmission Control Protocol).**

* You type in URL into the browser (<http://www.google.com>)
* Browser extract *http* part, then take the *domain name* (google.com) to ask the internet Domain Name Server to return an **Internet Protocol (IP) address**.
* Now that the client knows the IP address, it opens a TCP connection to connect with the server at that address.
* Client will initiate a GET request to the server which contains the IP address of the host and optionally a data payload:

GET / HTTP/1.1

Host: [www.codecademy.com](http://www.codecademy.com)

* The server responds this request:

HTTP/1.1 200 OK or HTTP/1.1 404 NOT FOUND

Content-Type: text/html

* After the server has sent the response, the client **closes the TCP connection**. This will be reopened anytime the browser requests new things from the server.

**APIs and API Endpoints**

Diagram

Description automatically generatedAPIs, which stands for **Application Programming Interface**, is a **set of rules (stating the requirements)** to how code components inside a software to connect and interact together, privately or public.

🡪 **A software can make use of other software’s functionalities**, which is accessible through their APIs.

**APIs can be online/offline**

*So what role does HTTP Request play with APIs? (access via HTTP -> Web APIs)*

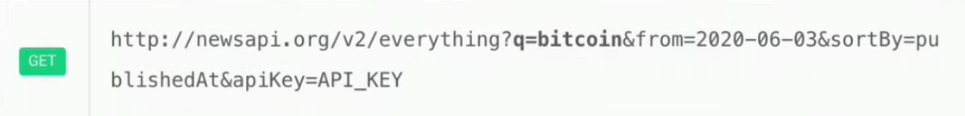
* All web services can be accessed using HTTP Requests 🡪 Some web services have APIs, allow us to connect and use it.
* Let’s say we want to connect to a website’s API 🡪 Check the Documentations on that website, to see instructions to how to connect to them (e.g., **which parameters to provide**)

A picture containing chart

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated- To use each web service, we need access to URL to retrieves particular functions. These specific functions are specified using **API Endpoints**.

Some API requires authentication (**API Key**), which sometimes cost monthly fees.