API-Application Programming Interface.

Interface : let’s say a YouTube video the play button works that happens due to Event Handler. That play button is an interface. This button is part of the API.

\*While UI is for the user of the Application. The API is made for the application Programmer to be used in the Application.

API – is a contract of sorts.

The term API is always used as a way to explain web-Based APIs but web based are just a subset of all types of APIs.

Frameworks ( Express.js/flask) provide an API that allow you to extend what has provided to make it your own.

Most web frameworks provide you with the ability to add a new route to your web application.

You don’t need to understand how it’s doing things. But you do need to know how to write an implementation to what the framework is expecting.

**What is an API ?**

Let’s say a remote connected to radio. The radio has an interface to it let’s say power button and volume button. If we increase the volume using the interface on remote. The remote requests the radio to increase volume and the radio responds by getting loud.

Remote API:

Let’s say Spotify it uses a Remote API to get all the songs from the Database and its’s surprisingly Quick.

REST – Representational state transfer.

The popularity of REST spread so rapidly that it nearly completely overtook the term API.

\*Just like we call xerox instead of photocopy. i.e. xerox is a brand . similarly for REST.

Deep dive into APIs that are written using the REST style.

When APIs embrace the style and constraints of REST, they are said to be RESTful !!

A web browser is a client and it connects to the server by the URL.

URL-Universal Resource Locator. Goes through HTTP – Hypertext transfer protocol.

A Protocol is also often Likened to a contract. It’s a lower level than an API. But its similar in concept.

The Browser creates a HTTP request for you, along with the URI and the HTTP also carries a verb such as GET,POST.

HTTP = URI + GET( To clarify that this request will only receive data.)

\*As we requested from the client to the server. the server sends response to the client.

The response consists of the body .the Hyper Text in the HTML takes care of the body.

RESTful Approach.

REST – Representational State Transfer.

When APIs embrace the style and constraints of REST, they are said to be RESTful !!.

1. Client-Server Architecture
2. Statelessness
3. Cachebility
4. Layered system.
5. Code on demand.
6. Uniform interface