API(Application Programming Interface)

API helps the user to connect their own program with someone else program.

DOM API (.getElementById)

Array Methods API(.map etc)

Client/ Server :-

Client: -

Any device that connects to the internet to get data from somewhere (”make a request”).

Example :- Laptop, phone. tablet.

Server: -

Accepts requests from a client asking for something then responds to the client with that thing

Example :- HTML page, an image file etc.

Request: -

When a device asks for a resource (data, image ,HTML page etc.)

Response: -

It replies to the request and could contain the resource (HTML,JSON etc.) asked by the client.

URL, Rest & BlogSpace

Components of a request: -

1) Path(URL)

2) Method

    like GET,POST,DELETE

    others like PATCH,OPTIONS

3) Body

4) Headers

Path: -

Address where your desired resource “lives”

BaseURL vs Endpoint

    Base URL: https://apis.scrimba.com/jsonlaceholder

    Endpoint: /posts

    Full URL:-https://apis.scrimba.com/jsonlaceholder/posts

Methods: -

GET: - Getting data

POST: - Adding a new data

Example: Like blog POST or facebook POST

PUT: - Updating existing data

DELETE: - Removing data

If you wan to get the GET the data or POST along with the fetch than you can use the following syntax: -

fetch(”https://apis.scrimba.com/jsonplaceholder/todos”, {method: “GET”})

Blog Space Project Notes: -

Body: -

The data we want to send to the servers.

Only makes sense with POST and PUT requests.

Needs to be turned into JSON first.

REST(REpresentational State Transfer): -

REST is a design pattern to provide a standard way for clients and servers to communicate.

Principles of REST: -

Client and server separation : -

Suppose client is navigating a weather app in browser and server end back that build HTML page with data and send it back.

Now what happens in REST the client request (GET) weather data in JSON and the server response weather data in JSON .

Statelessness(Session state): -

Accessing Resources: -

Interacting with Resources: -

Bicycle shop API: -

- /bikes
    - ▼ GET
    - ▼ POST(to add new)
- /bikes/: id (eg /bikes/123)
    - GET
    - PUT(to update)
    - DELETE(to delete data).
- /bikes/:id/reviews(eg /bikes/123/reviews) — > nested resource it will return the reviews of a specific bike
- /reviews —> it will return all the reviews of bike

Question: What URL use might use to GET the review with an ID of 5 on the bike with the ID of 123?

—> /bikes/123/reviews/5

Query Strings: -

Bicycle Shop API

- /bikes
- /bikes?type=mountain(question mark is the sign for query strings) (It will return the property in the form of objects for example in this case, {type: “mountain”} that’s the return type.
- /bikes?type=road&brand=trek (and for this case where multiple properties are present so it will return the object with multiple properties like {type:”road”,brand:”trek”}
- /bikes?type=hybrid&brand=trek&color=gray{ it will return like {type:”hybrid”,brand:””trek,color:”gray”}