CHAPTER AJAX-2

* AJAX (Async JS + XML)

Asynchronous JS, making requests, getting responses, managing DB etc with the server through this.

* WEB API

Interface that occurs over an http request and renders the webpage. Connected through the exposed API end point. Returns a Json format file.

* JSON (AJAJ)

JavaScriptObjectNotation- way of sending/ retrieve data in a formatted manner.

Rules: 1. Double quotes for key-value pair.

* Extract data from json: JSON.parse(data) 🡨Returns a JS Object

Covert JS Object to JSON- JSON.stringify(data)

* POSTMAN

Tool that helps visualize the http request and different components of the request being made.

COMPONENTS:

Body- Content that URL renders. Can be html, json etc.

Status- Numeric codes to indicate from server to user about the response status (Starting with 2 🡪 Kinda ok; Start with 3 🡪 Redirecting; Start with 4 🡪 Client errors; Start with 5 🡪 Server errors)

* HEADERS

Kind of Meta data for the request/ response.

Date, Last modified- Current actions of the access

Content type- What type of content was returned.

Can be set in postman using Header key-value pair. [Eg. Key= Accept, Value= Json].

* URL

url/search?q=<search item> : q+str = Query string that’s being searched. The url specifically looks for the letter ‘q’ for Query String.

To attach further queries: a=<search item1>&number=<search item 2>; where a and n are different end points for the API to fetch. [Key and Value for URL] 🡨Can be done in Postpone using Params or manually in the API url.

* XML HTTP Request

Older way of making an API request. Bulky code and confusing syntax.

When the request is send to the API, it responds with the different params like header, body etc.

* FETCH API

Fetch function is used to request the API and a promise is sent as response.

.then and .catch to be chained with fetch to extract information.

* .JSON

In the chaining for fetch, since it doesn’t wait for the entire data to be parsed, .json extracts the complete data in the form a promise from the .then of Fetch API.

* AXIOS

Built on Fetch Api. Simpler code.

.get()🡪 Method of Axios that directly provides a promise with all the params.

* AXIOS HEADERS

The headers can be changed according to the API (read documentation) to get desired format of response.

Eg. const config = {header: {Accept: ‘application/json’} }

const res = await axios.get(‘url’, config)

* AXIOS PARAMS

Just like setting header, to set a query we can use params. Params are just query string that need to be requested with the URL. Any number can be added to the param list.

Const searchTerm = form.element.query.value;

Const config = { params: {q: searchTerm}}

These can be chained with other configurations like headers etc.