Advanced Javascript

Day 9- fetch

Recap - XHR

4 Steps of XHR

Create XMLHttpRequest:

var xhr = new XMLHttpRequest();

Step 2

Initialize it, usually right after new XMLHttpRequest:

xhr.open(method, URL, [async, user, password])

Step 3

Send it out.

xhr.send(body)

This method opens the connection and sends the request to server. The optional body parameter contains the request body.

Some request methods like GET do not have a body. And some of them like POST use body to send the data to the server. We'll see examples of that later.

4 Steps of XHR

Listen to xhr events for response.

These three events are the most widely used:

load – when the request is complete (even if HTTP status is like 400 or 500), and the response is fully downloaded.

error – when the request couldn't be made, e.g. network down or invalid URL.

progress – triggers periodically while the response is being downloaded, reports how much has been downloaded.

Key to Understanding XHR

1. Difference between GET, POST, PUT, DELETE.
2. 4 Steps Of XHR
3. Do I need to send a body or not?
4. Sending the data in JSON
5. Listening to events.

Remember, as we have new requirements, we invent new way to do things.

XHR has it’s drawbacks, so we found a new way to make network requests.

fetch

The Fetch API, has been standardized as a modern approach to asynchronous network requests

Why Fetch?

Clean and easy syntax.

Modern way of making network requests.

3 Steps of XHR

fetch(url)

3 Steps of XHR

.then(function() {

//What do you want to do if you get back response

})

3 Steps of XHR

```
catch(function() {

//What do you want to do if there's an error

});
```

To summarize, using the Fetch API will look like this:

```
    fetch(url)
    .then(function() {

    })
    .catch(function() {

});
```