**Ajax**

* Ajax is a technique that allows web pages to communicate asynchronously with a server, and it dynamically updates web pages without reloading.
* Ajax allows JavaScript to request resources from a server on behalf of the client. The resources requested are usually JSON data or small fragments of text or HTML rather than a whole web page.
* Ajax was a neat acronym that referred to the different parts of the process being used: Asynchronous JavaScript and XML

**XML**

* When the term Ajax was originally coined, XML documents were often used to return data. Many different types of data can be sent, but by far the most commonly used in Ajax nowadays is JSON, which is more lightweight and easier to parse than XML.

**APIs**

* An application programming interface (API) is a collection of methods that allows external access to another program or service

**The Fetch API**

**Basic Usage**

* The Fetch API provides a global fetch() method that only has one mandatory argument, which is the URL of the resource you wish to fetch. A very basic example would look something like the following piece of code:

Text

Description automatically generated

**Response Interface**

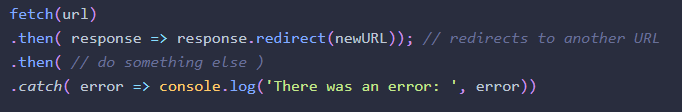
* Ok = Successful
* 200 = Response Successful
* 201 = Resource was created
* 204 = Request Successful but not content is returned

Some other properties of the Response object are:

* **headers** – A Headers object (see later section) containing any headers associated with the response
* **url** – A string containing the URL of response
* **redirected** – A boolean value that specifies if the response is the result of a redirect
* **type** – A string value of 'basic', 'cors', 'error' or 'opaque'. A value of 'basic' is used for a response from the same domain. A value of 'cors' means the data was received from a valid cross-origin request from a different domain.

**Redirects**

* The redirect() method can be used to redirect to another URL. It creates a new promise that resolves to the response from the redirected URL.



**JSON Responses**

* JSON is probably the most common format for AJAX responses. The json() method is used to deal with these by transforming a stream of JSON data into a promise that resolves to a JavaScript object.

**Creating Response Objects**

* Although most of the time you will be dealing with a response object that is returned from a request you make, you can also create your own response objects using a constructor function:

Graphical user interface, text

Description automatically generated

A constructor function is used to create a new Request object. An example is shown below:

Graphical user interface, text

Description automatically generated

**Putting It All Together**

* We can use the Headers, Request and Response objects to put together a typical example that sets up the URL, Request and Headers before calling the fetch() method:

Text

Description automatically generated

**FormData**

* The Fetch API includes the FormData interface, which makes it much easier to submit information in forms using Ajax.

A **FormData** instance is created using a constructor function:



* It’s also possible to add data to the form data instance as key-value pairs using the append() method:

