

Home / Blog

How to use fetch in JavaScript to GET or POST data

Posted on **20th April 2019**. 2 mins reading time

The code examples below are written in ES6/ES2015 code, if you require slightly older JavaScript, it can be converted to ES5 using [es6console](#)

The `fetch` command is an asynchronous function which allows you to send and receive data in JavaScript - commonly used for getting data from an API or similar.

GET data

The default method for `fetch` is `GET`. This allows you to pass in a URL and get the data from the other end. In the example below we are retrieving a JSON API. Using promises, the data is parsed and then made available for use.

The thing to note with asynchronous functions (in laymen's terms) is, although they fire when executed, they do not stop the rendering of the page. If part of your code requires the `fetch` to be complete before firing, ensure it is triggered in the promise (the `then` function).

```
fetch(URL)
  .then(data => data.json())
  .then(data => {
    // data is your API data
  });
...

```

POST data

There may be instances when your JavaScript code or app may wish to POST data to your backend code. This

```
```js
let data = {
 key: "value"
};

fetch(URL, {
 method: 'post',
 body: JSON.stringify(data)
}).then(data => {
 // data is anything returned by your API/backend code
});

```

## Bonus: PHP example

This one took me a while to figure out, but capturing the POST variables from your `fetch` function in a PHP script is not as obvious as it initially seems!

The key is using `php://input` - which is a [readable stream](#) which contains the data as a JSON string (as sent with the fetch function). This can be decoded using the native PHP function.

```
<?php

$input = json_decode(file_get_contents('php://input'), true);
// $input['key'] would equal "value"

```

If you were to remove the `true` from the end of the `json_decode`, your data would be available as a PHP object instead, meaning you would access "value" using `$input->key`.

This website is currently having a full content audit - apologies if some of the code or content looks a bit funky!

[Web](#) | [Javascript](#) | [PHP](#)

[View this post on Gitlab](#)

## You might also enjoy...

### Build your own "share this" component

Posted on **3rd May 2019**.

Using a Share This plugin (or similar) or the native sharing widgets from social networks can really bloat your page and look ugly. This code is how we, at Liquid Light, implement social sharing widgets

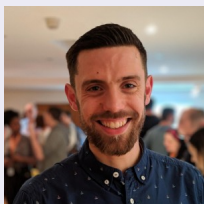
[Web](#) | [Javascript](#) | [jQuery](#)

### Developing websites locally with Docker

Posted on **14th February 2019**.

Docker allows you to host websites locally, which allows you can develop them within an environment which can be an exact replica of the live one. This blog post explores my first experience with Docker.

[Web](#) | [Command Line](#) | [Back-end Development](#)



#### Written by Mike Street

Mike is a front-end developer from Brighton, UK. He spends his time writing, cycling and coding. You can find Mike on [Twitter](#).

[Twitter](#) | [Instagram](#) | [Buy me a Coffee](#)

[View upcoming posts](#) | [The diary](#) | [Mike Street](#) | [RSS Feed](#)