

Building a single-page application with JS

AJAX & the single-page app

Asynchronous JavaScript and XML

A way of making requests programatically from your JavaScript instead of having the browser sent the request via a link or a url in the address bar.

This way, you can get and send data to a server without having to reload the page or go to a different page.

URL anatomy

`http://www.google.com/maps/index.html?search=tacos&zip=27701`

- protocol
- subdomain, second-level-domain, top-level domain
- path
- query string parameters
 - ? the start of the query string
 - key=value param pair
 - & (ampersand character) used to add additional query params
- ACSII encoding: only characters in the ASCII char set
 - e.g. space represented as %20

HTTP Request Methods

- GET Retrieve data (e.g., HTML or JSON)
- POST: Submit data for the first time
- PATCH, PUT: Update existing data, either all or in part
- DELETE: Delete existing data

CRUD

Many web applications have a common core functionality we know by the charming acronym **CRUD**, which stands for the common actions most applications need to do:

Create some data → POST

Read some data → GET

Uppdate some data → PATCH or PUT

Delete some data → DELETE

Status Codes

An HTTP response message includes a status code that can indicate something about the state of the request. These are chosen by humans and aren't always extremely precise!

- 1xx: Informational
- 2xx: Success
- 3xx: Redirection
- 4xx: Client Error
- 5xx: Server Error

200 is the response we most often want, indicating a successful request.

POST requests with Fetch

Notice that we pass a second argument to the `fetch` method, in addition to the `url`: an object containing options that supply additional details for the request.

```
fetch(url, {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({ username: "pesopenguin", email: "peso@octonauts.org" })
})
  .then(function (response) {
    return response.json()
  })
  .then(function(data) {
    console.log("You have been successfully subscribed", data)
  })
  .catch(function(error) {
    console.log("Something went wrong", error)
  })
})
```