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.domain.top-level-domain/path/query-string-parameters
```

- protocol
- subdomain, domain, top-level domain
- path
- query string parameters
  - `?` the start of the query string
  - `key=value` param pair
  - `&` 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
- `POST`, `PATCH`, `PUT`: Submit data for the first time or update existing data
- `DELETE`: Delete existing data

CRUD

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

- C: Create some data → `POST`
- R: Read some data → `GET`
- U: Update some data → `PATCH` or `PUT`
- D: Delete some data → `DELETE`

Status Codes

Whenever you make a request, the server will send you back a status code.

- `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

The following is a basic example of a POST request using `fetch`.

Notice that the arguments we pass to the `fetch` method include not only the url, but an object containing options that set 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(); // converts our data to JSON
  })
  .then(function(data) {
    console.log("You have been successfully subscribed", data);
  })
  .catch(function(error) {
    console.log("Something went wrong", error);
  });
```