Introduction to Communicating with the Server

(https://github.com/learn-co-curriculum/phase-1-intro-to-communicating-with-the-server) (https://github.com/learn-co-curriculum/phase-1-intro-to-communicating-with-the-server/issues/new)

Learning Goals

- Recall our Three Pillars of Web Programming
- Describe the process of communicating with the server
- Define AJAX

Introduction

We're two-thirds of the way through our exploration of using JavaScript with the DOM. We know how to navigate and change the DOM and we know how to make and recognize events. Along the way, we picked up necessary bits of JavaScript. Now we're getting ready to pull it all together.

Recall Our Three Pillars of Web Programming

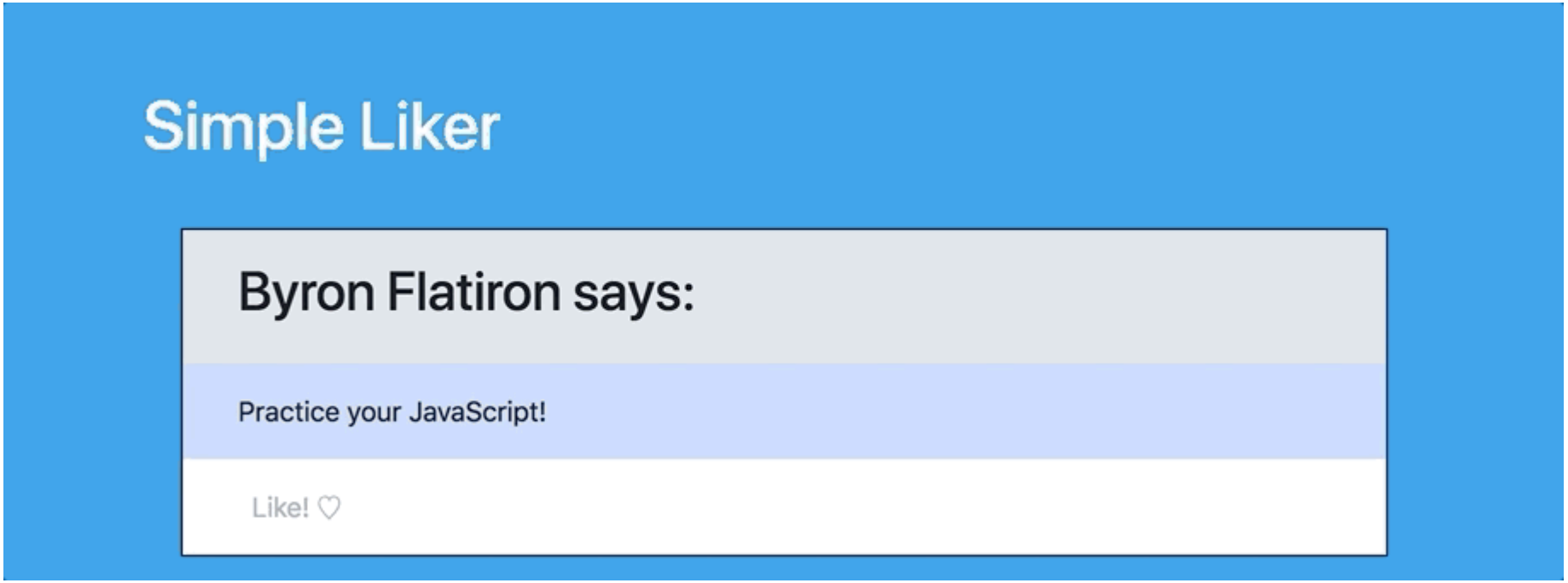
Remember our pillars? By this point, they should feel familiar, but it's worthwhile to refer back to them as we move along to keep our journey in perspective. So, in that spirit, our three pillars of web programming are:

- Manipulating the DOM
- Creating events
- Communicating with the server

The last piece, how we send and retrieve information from the server, is what we need to complete our "favoriting" app.

Describe the Process of Communicating With the Server

In our Simple Liker app, “favoriting” is a click event on a heart icon that updates the user’s DOM to show a full heart.



The click event kicks off a sequence of actions to notify the server that the post has received a like. The server updates the post in the backend then passes a message back to the browser indicating that the update was made successfully. When that success message is received, we then update the DOM to reflect the change.

> **NOTE:** The update to the DOM is not *necessarily* dependent on a success message from the server. We could just update the DOM in response to the click event itself. However, this is ***not*** the proper procedure. We only want to update the DOM once we know that the server successfully *persisted* the change in the backend.

The user doesn't see this entire process happening. Ideally, the process moves quickly enough that the user barely even notices that it took place. All they know is that the little heart icon is now reflecting their clicked appreciation. To keep the user experience fast and smooth, we use something called the *AJAX technique*.

Define AJAX

AJAX is short for "asynchronous JavaScript and XML," and it's the process used to make requests to the server and update the DOM without reloading the web page. There are a few different ways to implement this; a bit later in this section we'll take a look at one of the most efficient

ways: `fetch()`.

The name "asynchronous JavaScript and XML" arises from the fact that, in the past, the data sent back to the browser from the server was encoded as XML. Now, however, it's most often sent back in a format known as JSON ("Jay-Sawn"). JavaScript Object Notation (JSON) is a `String` that JavaScript knows how to turn into an `Object`. Using JavaScript, we can access the JSON returned by the server and use it to update the DOM.

Conclusion

The last skill we need to be effective JavaScript web programmers is communication with the server. Once we've mastered this final step, we will be able to listen for an event, persist the change to the backend, and manipulate the DOM to reflect the updated information. With the AJAX technique, we'll learn how to send and receive data quickly so that we keep our users' experience a positive one.