Notes

Friday, April 17, 2020 10:29 AM

https://teamtreehouse.com/library/introducing-ajax-2

Resources

https://www.w3schools.com/js/js_ajax_intro.asp

https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX

Prerequisites

It's a good idea to have some knowledge of JavaScript and jQuery.

Introduction

- Asynchronous JavaScript and (XML or AJAX)
    - XML the X in AJAX isn't used as often anymore
    - JSON is now the most popular format and is much easier to use with JavaScript
- If AJAX wasn't asynchronous it would freeze your browser. When a request is sent it doesn’t stop everything it's doing and wait for a response. Sometimes a response isn't even returned.
- AJAX allows you to update a part of an HTML page without loading or reloading a webpage
- AJAX feels faster because it loads only certain parts of the page instead of the entire page
- AJAX works with server-side languages such as Ruby, PHP, Python and more…
- Example: Twitter is a page that never ends. More tweets are loaded as you scroll to the end of the page

1. **REQUEST:** AJAX asks for information from a server
2. **RESPONSE:** Web server returns data to the web browser AKA **client**
3. JavaScript process that data to selectively change parts of a page

- **XHR Object** is acronym for XMLHttpRequest Object

4 Steps of AJAX Programming Process

1. Create an XMLHttpRequest Object
2. Create a callback function AKA event handler
3. Open a request
4. Send the request

AJAX Security

- AJAX only works if your viewing your page thru a web server
- Same Origin Policy

    - Can't use AJAX to access other servers because it's not from same origin
    - Can't switch protocols even if on same server. Example if your on http:// you can't make request to https:// even if it's on the same server
    - Can't switch port numbers
    - Can't switch hosts e.g.) http://www.myserver.com can't make a requst to http://db.myserver.com
- You can still ebmed Google Maps, Tweets etc. even though these assets are on different domains. There are a few ways to circumvent the same origin policy.
    - **CREATE A WEB PROXY:**
        - A web proxy allows you to bypass a web browser's same origin policy by retrieving data from another server, while keeping AJAX requests within the same domain.
        - Web servers aren't limited by the same origin policy so they can request data from servers at other domains. Because of this you can setup a script in PHP for example that sits on your server and asks for information from another web server.
            - AJAX can then be used to talk to the script on your site, which talks to the other site and returns the data to your page.
            - This makes sure the AJAX part stays within the same webpage and obeys the same origin policy.
    - **JSONP (JSON with Padding)**
        - It's not traditional AJAX and relies on the ability to link to JavaScript files across domains.
        - Browsers allow many types of cross domain links.
            - E.g.) You can link to photos, CSS or JavaScript files on other sites.
            - Loading JS files across domains is a common technique when working with popular JavaScript libraries like jQuery and is how CDNs work.
            - It saves disk space and your server's processing power since the file doesn’t sit on your own server.
            - Instead of using AJAX to contact another web server, you load a JavaScript file from the other site.
                - This is perfectly okay with a browser. That JavaScript file contains the info you're after.
    - **CORS (Cross-Origin Resource Sharing)**
        - CORS is a W3C recommendation implemented in most current browsers.
        - It requires setup on the server's part
            - It allows a server to accept requests from other domains.
            - It even allows for more complex types of authentication that require the web browser to supply credentials before the web server will provide information.

Parsing JavaScript

- When we receive responseText it may look like HTML, XML, or JSON. but it's really nothing more than a string of plain text.
- This text that needs to be parsed to be understood by the web browser
    - E.g.) Convert a JSON data string and convert it into JavaScript

- Parsing is easy and can be done with a single command

Cheatsheet - Important Parts

Method	**Description**
var xhr = new XMLHttpRequest();	Creates new XHR Object
xhr.onreadystatechange	Uses a callback function (event handler) to listen for property state changes
xhr.readystate	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
xhr.status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" View more
xhr.responseText	Returns the response data as a string
JSON.parse(xhr.responseText)	Parses responseText to convert from a string to JavaScript
xhr.open(method, url, async)	Specifies the type of request method: the type of request: GET or POST url: the server (file) location async: true (asynchronous) or false (synchronous), true is default and not required if true
xhr.send()	Sends the request to the server (used for GET)