Getting Rid of Synchronous XHRs

By Eric Bidelman

Engineer @ Google working on web tooling: Headless Chrome, Puppeteer, Lighthouse

Heads up! The XMLHttpRequest2 spec was recently changed to prohibit sending a synchronous request when `xhr.responseType` is set. The idea behind the change is to help mitigate further usage of synchronous xhrs wherever possible.

For example, the following code will now throw an `INVALID_ACCESS_ERR` in developer channel builds of Chrome and FF:

```
var xhr = new XMLHttpRequest();
xhr.responseType = 'arraybuffer';
xhr.open('GET', '/', false); // sync request
xhr.send();
```

See WebKit Bug, Buzilla Bug

Note: the ability to parse HTML has also been added to `XMLHttpRequest` but the caveat is that you can't use it unless you're sending an asynchronous request! See HTML in XMLHttpRequest.

Synchronous XHRs are bad for a number of reasons, but MSDN's blog post, "Why You Should Use XMLHttpRequest Asynchronously" has a great explanation of the issues.

This is a generally a great change for the web, but it has the potential to break some existing apps that were relying on synchronous behavior. Please look over your XHR code and update it ASAP to use asynchronous requests.

Last updated July 2, 2018.