

12. How the Web Works

Part 2

Blase Ur and David Cash
February 8th, 2021
CMSC 23200 / 33250



THE UNIVERSITY OF
CHICAGO

So... Interactive Pages?

- Javascript!
 - The core idea: Let's run (somewhat) arbitrary code on the **client's** computer
- Math, variables, control structures
- Imperative, object-oriented, or functional
- Modify the DOM
- Request data (e.g., through AJAX)
- Can be multi-threaded (web workers)

Common Javascript Libraries

- JQuery (easier access to DOM)
 - `$(".test").hide()` hides all elements with `class="test"`
- JQueryUI
- Bootstrap
- Angular / React
- Google Analytics (*sigh*)

Importing Javascript Libraries

```
673
674         </ul>
675     </div>
676 </div>
677 </div>
678 <div class="row">
679     <div class="footer_copy">
680         <p>2021 <span class="url fn org">The University of Chicago</span></p>
681     </div>
682 </div>
683 </div>
684 <a id="back-to-top" href="#" class="back-to-top" role="button"></a>
685 </footer>
686
687 <script defer src="/js/libs/modernizr.js?updated=20191205080224"></script>
688 <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
689 <script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.4/jquery-ui.min.js"></script>
690 <script>window.jQuery || document.write('<script src="/js/libs/jquery/2.1.4/jquery.min.js"></script><script src="/js/libs
691 <script defer src="/js/core-min.js?updated=20191205080225"></script>
692
693 <!--[if lte IE 8]><script src="/js/libs/selectivizr.js"></script><![endif]-->
694 <!--[if lte IE 9]><script src="/js/ie_fixes/symbolset.js"></script><![endif]-->
695 <!--<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.lifestream/0.3.7/jquery.lifestream.min.js"></script> -->
696
697
698
699
700
701 <script async src="https://www.googletagmanager.com/gtag/js?id=UA-3572058-1"></script>
702 <script>window.dataLayer = window.dataLayer || [];function gtag(){dataLayer.push(arguments);}gtag('js', new Date());
703 gtag('config', 'UA-3572058-1');gtag('config', 'UA-187440939-1');</script>
704
705 </body>
706 </html>
707
```

Do You Have the Right .js File?

- Subresource integrity:
https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity
- `<script src=https://example.com/example-framework.js integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQIGYI1kPzQho1wx4JwY8wC" crossorigin="anonymous"></script>`
- `cat FILENAME.js | openssl dgst -sha384 -binary | openssl base64 -A`

Same-Origin Policy

- Prevent malicious DOM access
- Origin = URI scheme, host name, port
- Only if origin that loaded script matches can a script access the DOM
 - Not where the script ultimately comes from, but what origin **loads** the script
- Frames / iframes impact origin
- CORS (Cross-Origin Resource Sharing)

Same-Origin Policy (SOP)

https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy



Definition of an origin

Two URLs have the *same origin* if the protocol, port (if specified), and host are the same for both. You may see this referenced as the "scheme/host/port tuple", or just "tuple". (A "tuple" is a set of items that together comprise a whole — a generic form for double/triple/quadruple/quintuple/etc.)

The following table gives examples of origin comparisons with the URL `http://store.company.com/dir/page.html`:

URL	Outcome	Reason
<code>http://store.company.com/dir2/other.html</code>	Same origin	Only the path differs
<code>http://store.company.com/dir/inner/another.html</code>	Same origin	Only the path differs
<code>https://store.company.com/page.html</code>	Failure	Different protocol
<code>http://store.company.com:81/dir/page.html</code>	Failure	Different port (<code>http://</code> is port 80 by default)
<code>http://news.company.com/dir/page.html</code>	Failure	Different host

CORS (Relaxes SOP)

- Cross-Origin Resource Sharing
 - Specifies when specific other origins can make a request for data on a different origin
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

Sending Data to a Server

- GET request
 - Data at end of URL (following “?”)
- POST request
 - Typically used with forms
 - Data *not* in URL, but rather (in slightly encoded form) in the HTTP request body
- PUT request
 - Store an entity at a location

URL Parameters / Query String

- End of URL (GET request)
 - <https://www.cs.uchicago.edu/?test=foo&test2=bar>

The screenshot shows a web browser displaying the University of Chicago Department of Computer Science website. The address bar shows the URL <https://www.cs.uchicago.edu/?test=foo&test2=bar>. The website header includes the University of Chicago logo and the text "Department of Computer Science". Below the header is a navigation bar with links: ABOUT, PEOPLE, RESEARCH, UNDERGRADUATE, GRADUATE, and ADMISSION. The browser's developer tools are open, showing the Network tab. The network log displays a list of requests, with the first request (Status 200, Method GET, URL /?test=...) selected. The right-hand pane of the developer tools shows the "Params" tab for the selected request, displaying the query string parameters: test: foo and test2: bar.

THE UNIVERSITY OF CHICAGO

Department of Computer Science

Industry / Diversity / Apply

ABOUT PEOPLE RESEARCH UNDERGRADUATE GRADUATE ADMISSION

Inspector Console Debugger Style Editor Performance Memory Network Storage Accessibility New

Filter URLs

Status	Method	F...	Domain	Cause	Type	Transferred	Size
200	GET	/?test=...	www.cs.uchi...	document	html	6.76 KB	23.87 KB
302	GET	fonts.css	cloud.typogr...	stylesheet	css	154.58 KB	205.03 KB
200	GET	main.cs...	www.cs.uchi...	stylesheet	css	cached	189.57 KB
200	GET	moder...	www.cs.uchi...	script	js	cached	5.65 KB
200	GET	jquery....	ajax.googlea...	script	js	cached	0 B
200	GET	jquery-...	ajax.googlea...	script	js	cached	0 B

Headers Cookies Params Response Timings Stack Trace

Filter request parameters

Query string

test: foo

test2: bar

Processing Data on the Server

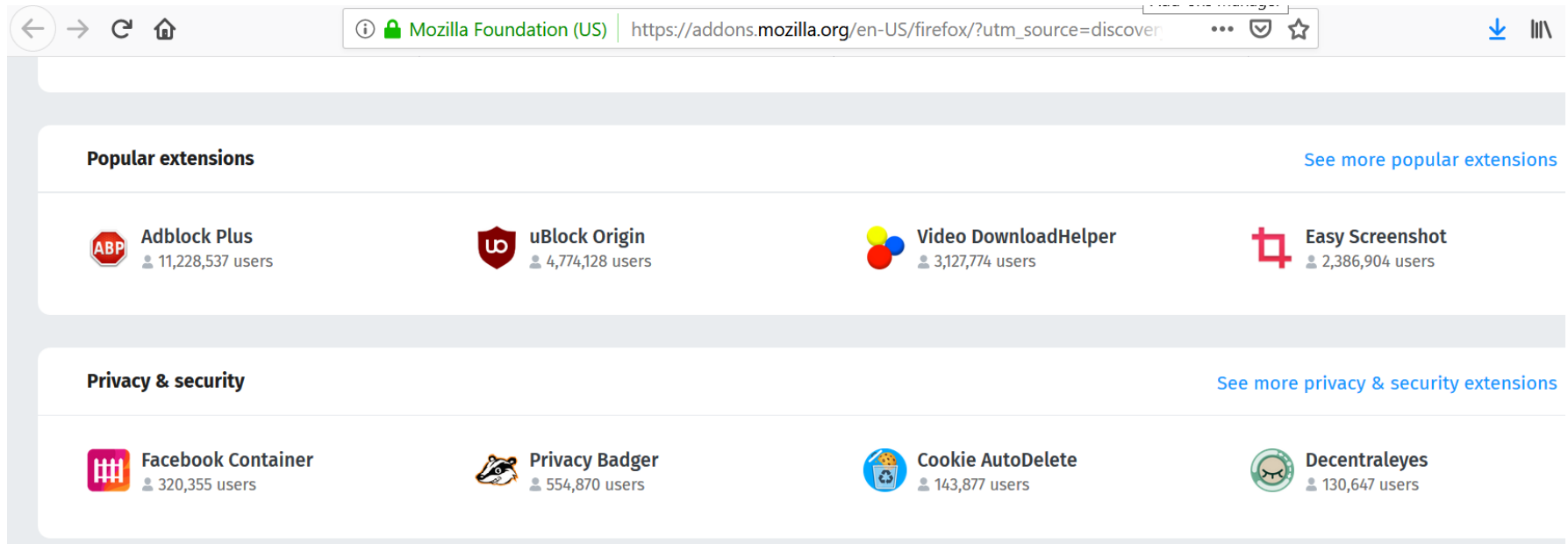
- Javascript is client-side
- Server-side you find Perl (CGI), PHP, Python (Django)
- Process data on the server
- What happens if this code crashes?

Storing Data on the Server

- Run a database on the server
- MySQL, SQLite, MongoDB, Redis, etc.
- You probably don't want to allow access from anything other than *localhost*
- You definitely don't want human-memorable passwords for these

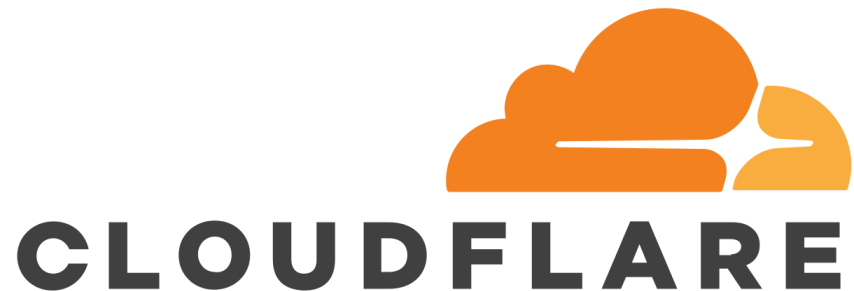
Browser Extensions

- Can access most of what the browser can
- Requires permissions system
- Malicious extensions!



What If You Get Lots of Traffic?

- CDNs (content delivery networks)



What If You Don't Want To Code?

- CMS (content management system)
 - WordPress (PHP + MySQL), Drupal

The screenshot displays the WordPress dashboard interface. At the top, the site name 'Restaurant World Tou...' is visible, along with navigation links for 'Upgrade to Pro', 'New Post', and a user profile for 'Dave'. The left sidebar contains a menu with options like 'Dashboard', 'Home', 'Comments I've Made', 'Site Stats', 'Akismet Stats', 'My Blogs', 'Blogs I Follow', 'Store', 'Posts', 'Media', 'Links', 'Pages', 'Comments', 'Feedbacks', 'Appearance', 'Users', 'Tools', 'Settings', and 'Collapse menu'. The main content area is titled 'Dashboard' and features several widgets:

- Right Now:** A summary of site statistics.

CONTENT	DISCUSSION
8 Posts	9 Comments
1 Page	9 Approved
5 Categories	0 Pending
52 Tags	0 Spam
- QuickPress:** A form for quickly creating a new post, including fields for title, content, tags, and buttons for 'Add Media', 'Save Draft', 'Reset', and 'Publish'.
- Recent Comments:** A list of recent comments, showing a comment from 'Dave' on 'Arctic Char #'.
- Storage Space:** A widget showing storage usage: '3,072MB Space Allowed' and '0.08MB (0%) Space Used'.
- Recent Drafts:** A section indicating 'There are no drafts at the moment'.
- Stats:** A section stating 'No stats are available for this time period.'