asynchronous calls

Daniel Jackson
some history

in the 1990s
› most web sites issued a whole page at a time
› clunky for users, excessive bandwidth

idea
› update page incrementally
› do it asynchronously, so browser doesn’t freeze

in 1999, XMLHttpRequest arrives
› Microsoft invents XHR idea for ActiveX in IE5
› adopted by other browsers for Javascript

in 2005, called “Ajax”
› asynchronous JavaScript and XML
what you can do with Ajax

push updates to page
› status, stock quotes, weather, time

interact with user over input
› drill downs (eg, state>city), autocomplete

persist user data on server
› Google docs, stickies

execute server-side commands on same page
› vote up/down on posts, email portal
event model

› single thread
› event loop: run timed event if expired or event at front of Q
› events added by: user actions, server responses, events
particulars

two key facts
› each event runs to completion before next event
› response to Ajax request is not always next

good news
› no need to worry about mutual exclusion

bad news
› long-running event freezes the UI
› timeouts may run late
› when Ajax response comes, context may have changed
JQuery’s Ajax API

$.getScript(url, callback)
› get script at url, run it, then execute callback

e.load(url)
› get content at url, and insert into element e

$.getJSON(url, data, callback)
› pass data to server at url
› server sends JSON back; parse and pass to callback

$.get(url, data, callback, type)
› type determines preprocessing of response
› $.post similar, but does HTTP post instead

$.ajax(...)
› most flexible, lowest level method
using network inspector

<table>
<thead>
<tr>
<th>Name Path</th>
<th>Method</th>
<th>Status Text</th>
<th>Type</th>
<th>Size Transf</th>
<th>Time Latenc</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Users/dnj/Filestore/ajax-examples.htm</td>
<td>GET</td>
<td>(Pen...</td>
<td>text...</td>
<td>872B</td>
<td>9ms</td>
<td>5ms</td>
</tr>
<tr>
<td>code.jquery.com</td>
<td>GET</td>
<td>304 Not Mc</td>
<td>appl...</td>
<td>232.56kB</td>
<td>108ms</td>
<td>108ms</td>
</tr>
<tr>
<td>127.0.0.1/static</td>
<td>GET</td>
<td>304 NOT M</td>
<td>text...</td>
<td>32.60kB</td>
<td>85ms</td>
<td>85ms</td>
</tr>
<tr>
<td>127.0.0.1/static</td>
<td>GET</td>
<td>304 NOT M</td>
<td>text...</td>
<td>205.32kB</td>
<td>83ms</td>
<td>83ms</td>
</tr>
<tr>
<td>/Users/dnj/Filestore/ajax-examples.js</td>
<td>GET</td>
<td>(Pen...</td>
<td>text...</td>
<td>3.01KB</td>
<td>112ms</td>
<td>107ms</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>GET</td>
<td>200 OK</td>
<td>appl...</td>
<td>388B</td>
<td>39ms</td>
<td>37ms</td>
</tr>
</tbody>
</table>

6 requests | 4.69KB transferred

› example from Safari
using network inspector

› can see here that request was get
encoding data for transit

XML
› parsing built into browser (XHR)
› comes back as DOM: not convenient

JSON
› Javascript object literals
› JQuery uses parser, not eval (why?)

examples from http://en.wikipedia.org/wiki/JSON