

CSCC09F

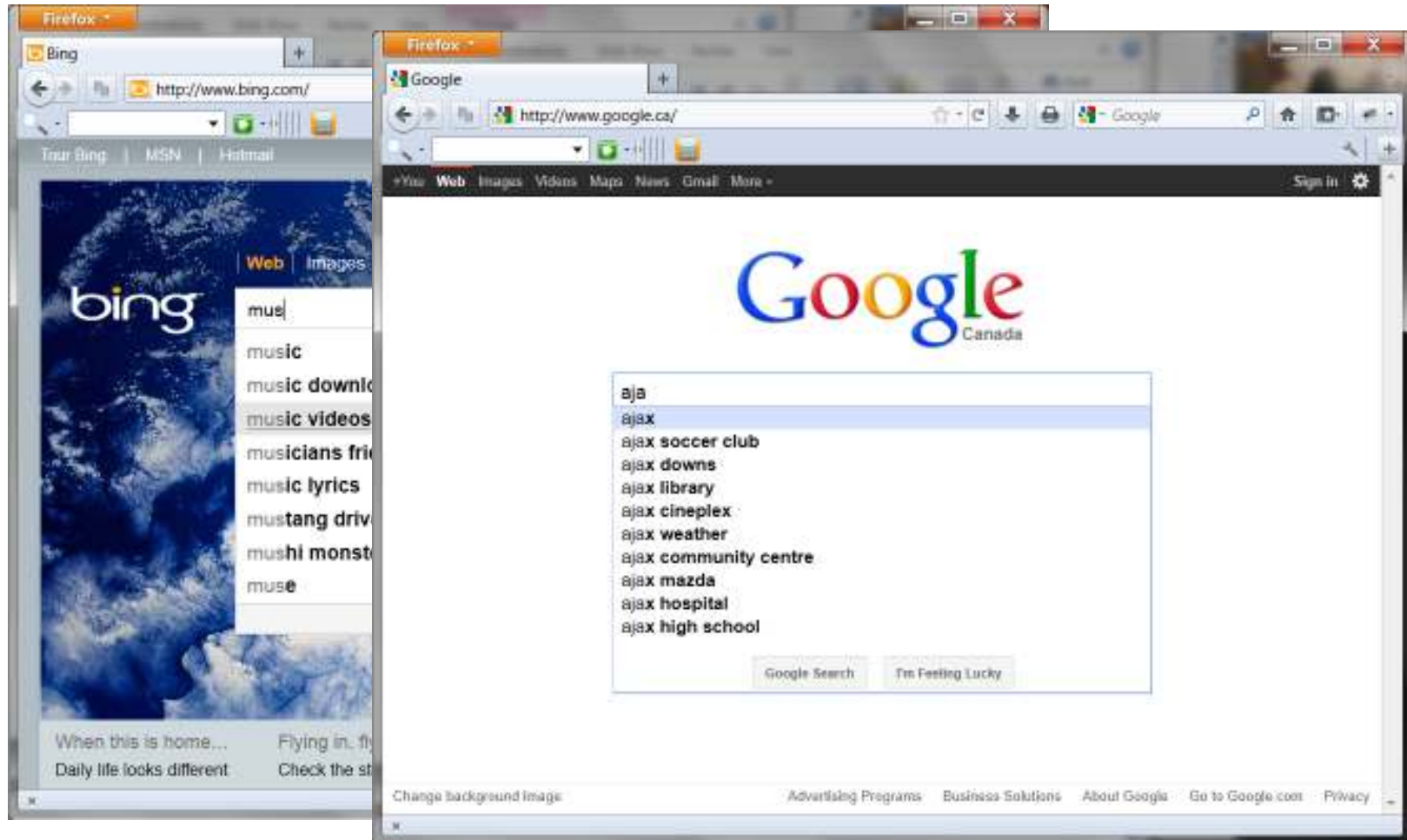
Programming on the Web



Ajax

Enabling RIA/SPA/Web-2.0 applications

Ajax example: autocomplete

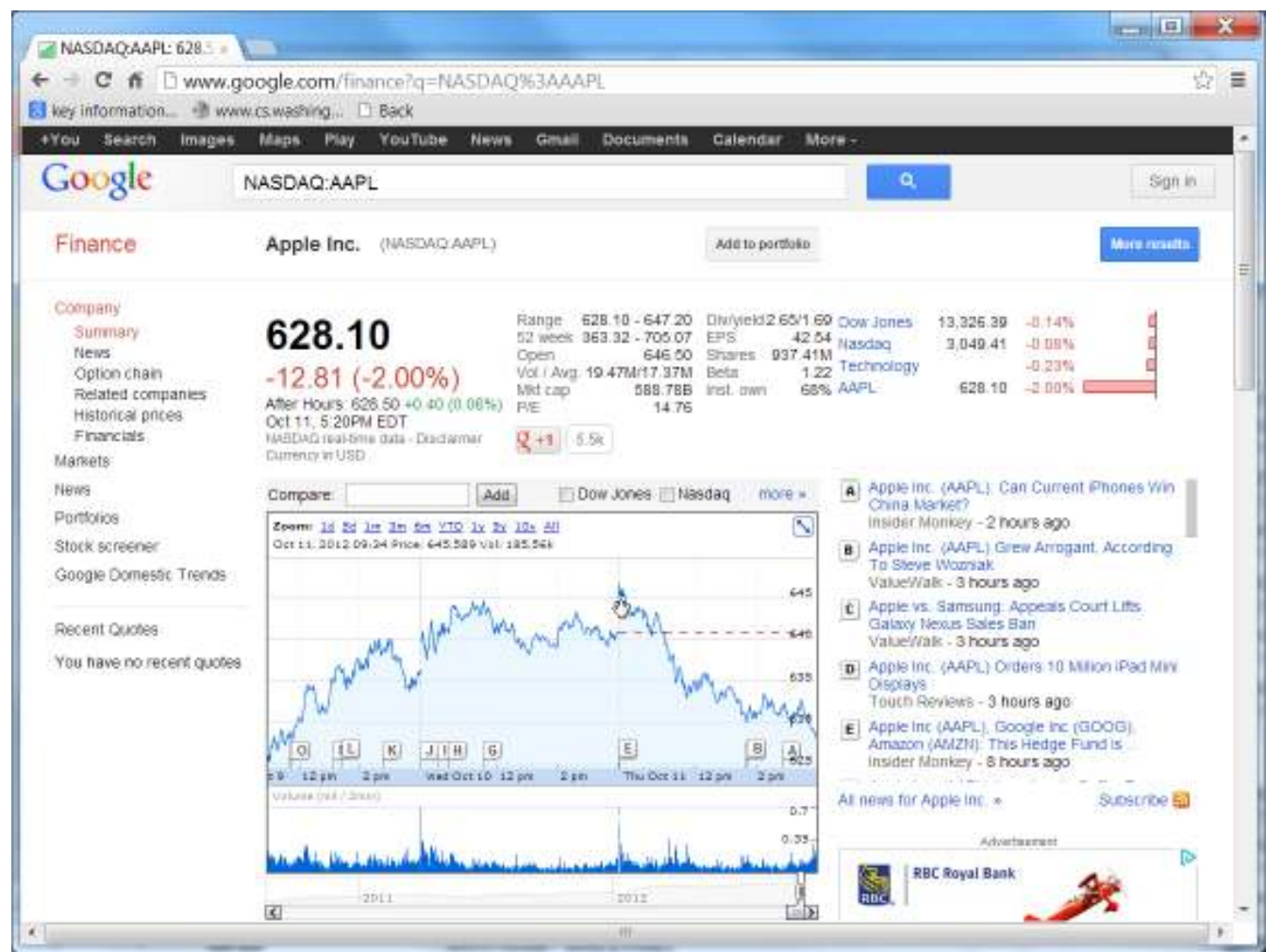


24 AJAX

CSCC09 Programming on the Web

2

Ajax example: live data updates

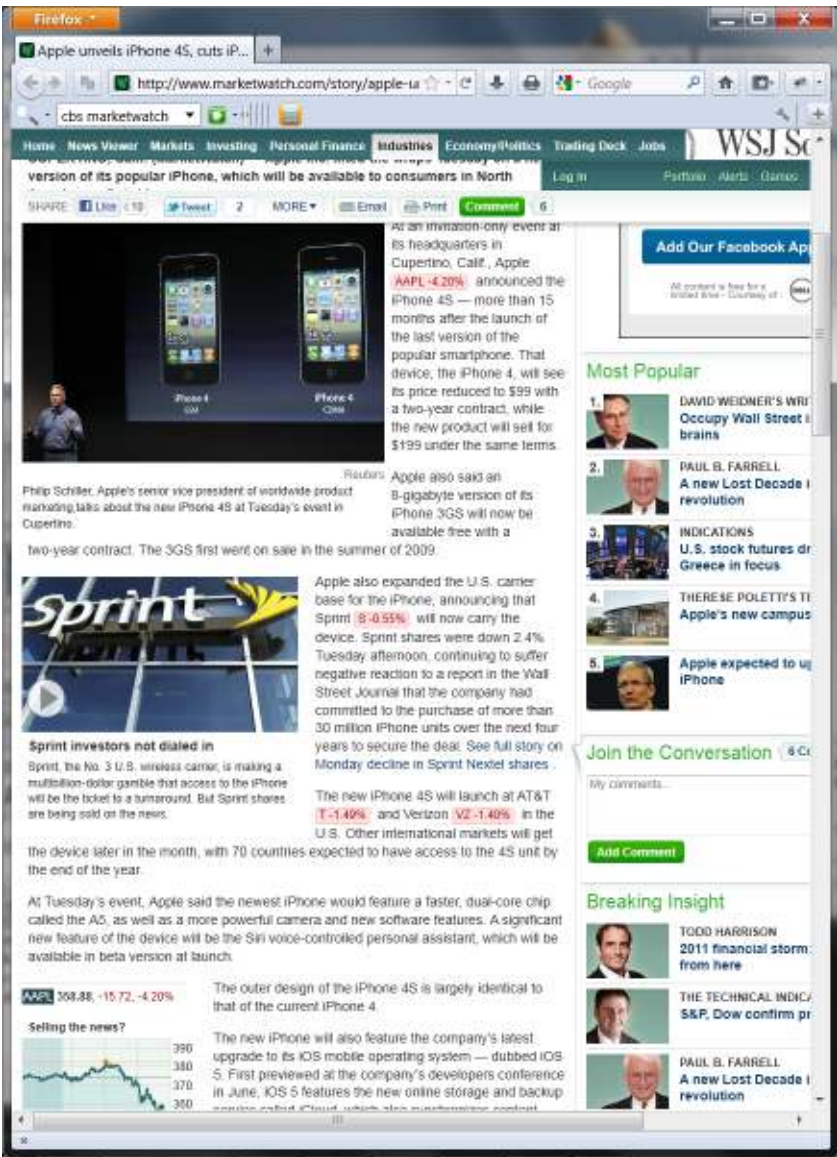


24 AJAX

CSCC09 Programming on the Web

3

Ajax example: live document updates



Ajax



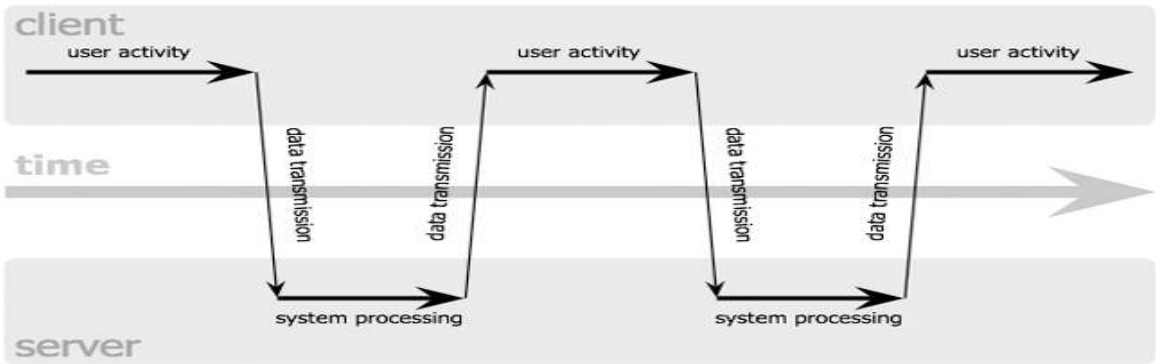
- ❑ Ajax coined as a term by Jesse James Garrett in Feb. 2005
 - based on MS protocol dating back to 2001
 - originally acronym for “Asynchronous JavaScript with XML”, now just a word (XML not the dominant use case any more)
- ❑ Early high-traffic app: Google releases Gmail in March of 2004
 - one of first mainstream apps to use Ajax
 - examples: read/tag/spell-check messages without a page reload, auto save drafts

Ajax Interaction

- ❑ How does Ajax change client-server interaction?
- ❑ Traditional document-model interaction:
 - user-initiated HTTP requests
 - ❑ typing on navigation window, or clicking on form or hyperlink
 - response from server replaces existing page
 - low request rate, and random amount of time between requests (user driven, not program driven)
- ❑ Ajax interaction:
 - new type of request that does not trigger page-reload
 - not initiated by user (at least not directly)
 - requests typically small, but more frequent

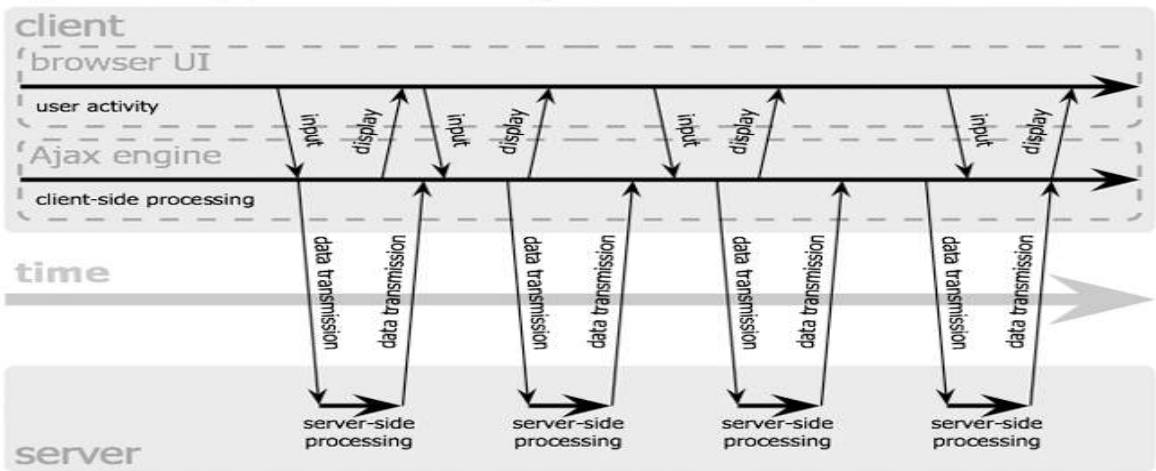
Ajax-Server Interaction

classic web application model (synchronous)



traditional
Web app

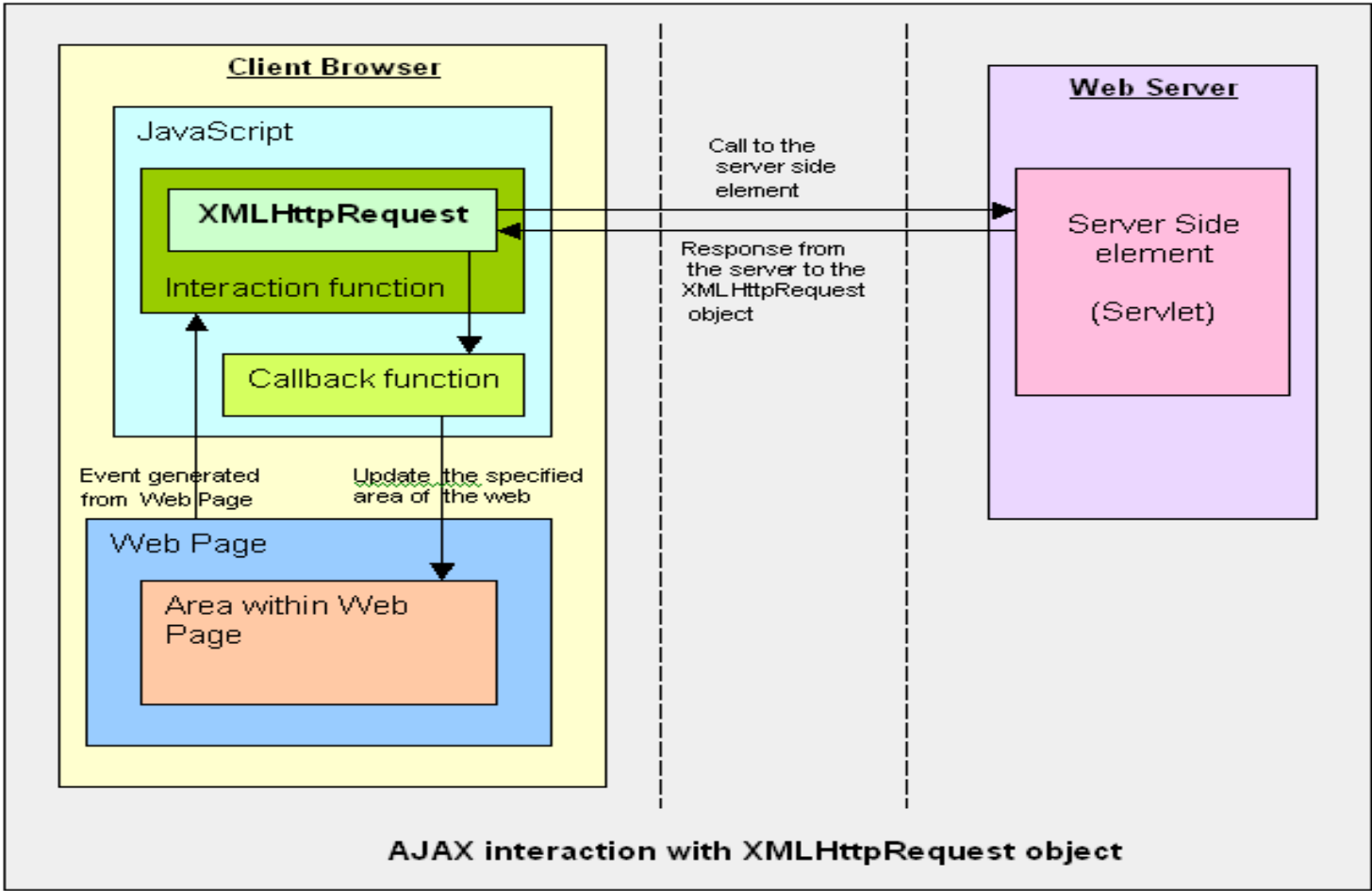
Ajax web application model (asynchronous)



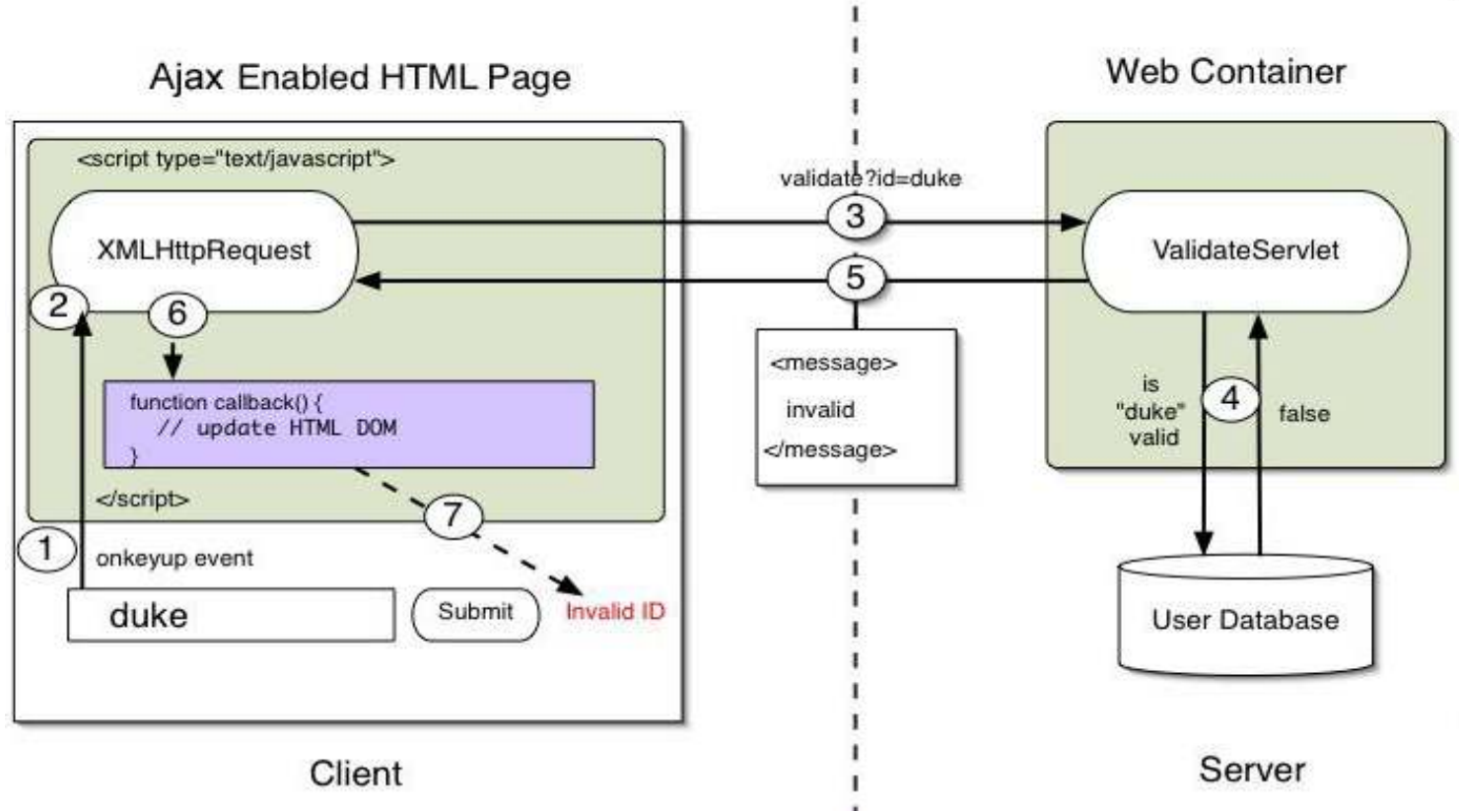
Ajax-based
Web app

Jesse James Garrett / adaptivepath.com

Ajax Control Flow



Ajax Example: Input Validation



Ajax is Asynchronous

□ Ajax requests are
“asynchronous” i.e.
non-blocking

- don't block browser (JS/DOM evaluation) if 3rd param to request.open() is true
- browser can continue running JavaScript
- user can continue interaction with page, generating events
- can mask long-latency, low-bandwidth, slow-server response time

24 AJAX

```
var req=XMLHttpRequest();
req.onreadystatechange = function() {
    if (req.readyState == 4) {
        if(req.status == 200) {
            alert(req.responseText);
        }
        else {
            alert('Loading Error: [' +
                req.status + ']' +
                req.statusText);
        }
    }
}
req.open('GET','ajaxserver',true);
req.send(null);    // request body
```

CSCC09 Programming on the Web

13

Ajax Request readyState

```
if (req.readyState == 4) { ... }
```

What do the readyState values mean?

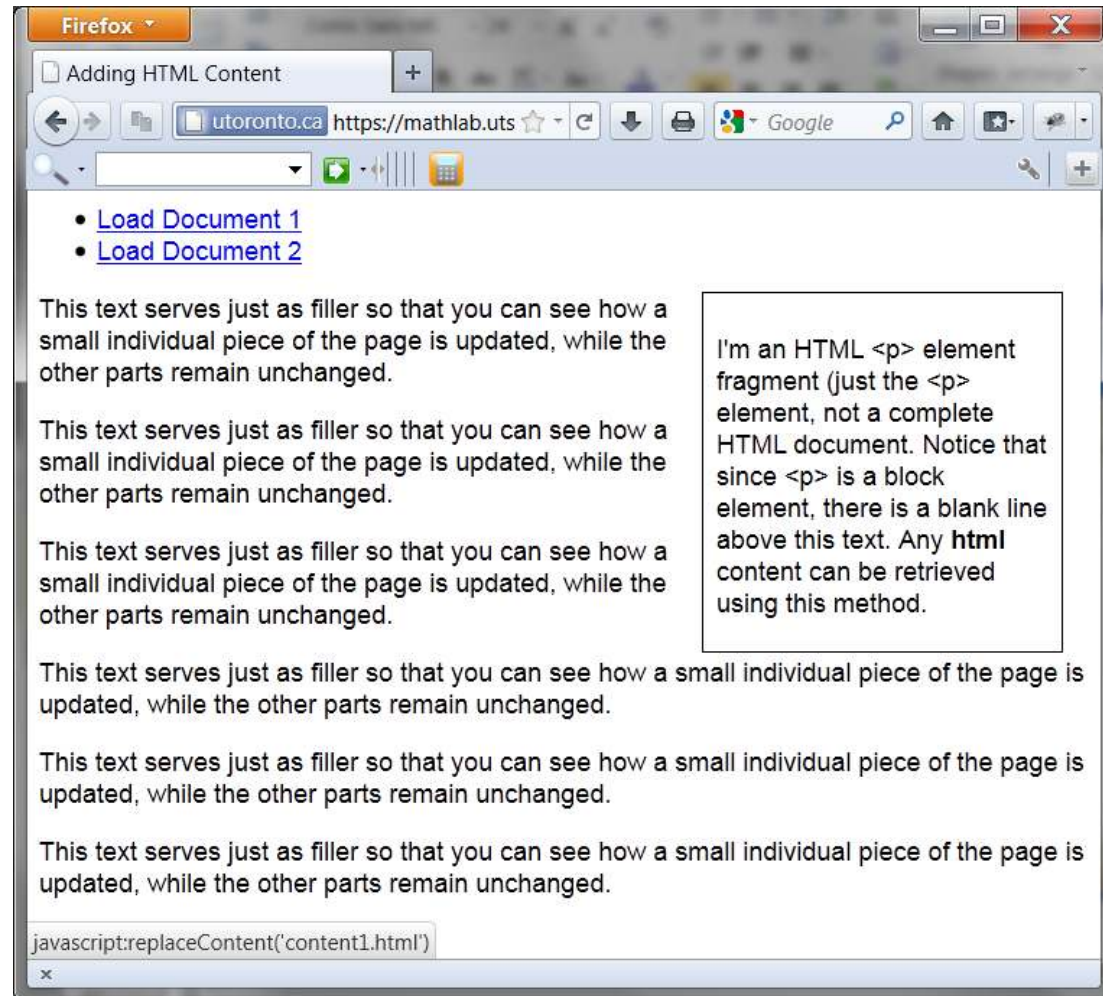
0. The request is uninitialized (before you've called `open()`).
1. The request is set up, but not sent (after `open()` but before you've called `send()`).
2. The request was sent and is in process (you can usually get content headers from the response at this point).
3. The request is in process; often some partial data is available from the response, but the server isn't finished with its response.
4. The response is complete; you can get the server's response and use it (usually use this one)

Ajax Example (static content)

Examples:

[ajax_static.html](#)

[ajax_dynamic.html](#)



Can send() a payload

- ❑ Makes connection to the URL specified in `open()`
- ❑ Sends cookies and other default headers
- ❑ POST request:
 - send parameter is sent as body of HTTP message
 - need to set Content-type header
- ❑ Example: mimic sending a form (ajax form.html):

```
var req = new XMLHttpRequest();  
req.open('POST','serverSideApp', false);  
req.setRequestHeader('Content-type',  
    'application/x-www-form-urlencoded');  
req.send('param1=firstVal&param2=secondVal');  
if (req.status==200) alert(req.responseText);
```

Security Restrictions

- ❑ ‘Same Origin’ policy: an app can only up/load data to/from the domain of the currently-loaded page.
Why?
- ❑ Creates barriers to building “mashups” from parts hosted in different domains. Boooo!
- ❑ Workaround (kludge/hack) for Ajax: jsonp wraps data as a script and delivers it to a callback function
- ❑ CORS standard mitigates Same-Origin issue, by providing HTTP response-header servers can use to grant permission for access from other domains, e.g.:
Access-Control-Allow-Origin: *

Mitigating Ajax Issues

- ❑ The XMLHttpRequest object is still inconsistently implemented by different browsers; so you have to write code that checks which browser it is running on – ugly
- ❑ Fairly low-level interface: write more code, create more bugs
- ❑ To some extent, these issues can be mitigated through use of a JavaScript framework like jQuery.
- ❑ jQuery hides browser differences, and significantly simplifies the code required to write Ajax-enabled applications.
 - compare the [jquery_ajax.html](#) example code with the plain JavaScript in the other examples. Code is smaller, easier to understand, more maintainable, not browser dependent.

jQuery Ajax example

```

<body>
  <input type="button" class="getcds" value="Show CDs"/>
  Selected CD Titles: <br/>
  <div id="cdcat"></div>
  <script type="text/javascript" src="https://code.jquery.com/jquery.min.js">
  </script>
  <script type="text/javascript">
    $(document).ready(function(){
      $("body").on("click", ".getcds", function() {
        $("#cdcat").html('');
        $.getJSON('https://mathlab.../rosselet/lec/40cgi_ex/cdcat.cgi',
          { 'myparam': 'testing' }, // you can pass extra param values
          function(data) {
            var items = []; // array to contain HTML <li>'s
            $.each(data, function(key, val) {
              items.push('<li id="' +key+ '">' +val.title+ '</li>');
            });
            $('<ul/>', { html: items.join('') }).replaceAll('#cdcat');
          });
      });
    });
  </script>
</body>

```

❑ See [jquery_ajax.html](#) in 22jquery lecture examples for complete code

Ajax and Backbone

- ❑ Backbone uses Ajax behind the scenes to perform tasks such as sync'ing models with persistent data stores, when you invoke methods like `model.save()`, `collection.create()`, and `collection.fetch()`
- ❑ In later assignments, we'll be using jQuery methods to send Ajax requests for tasks such as:
 - change users' authentication status (login/logout)
 - upload image files to the server
 - translate street addresses into GPS coordinates