

Rich Internet Application (RIA)

- Bridge gap between native application and normal Internet ones
- More interactive web applications
- Perform similar to native application
- Enabling technologies: JavaScript, DHTML (DOM), SVG, AJAX
- Shift from page-based interaction to event-driven programming where event result in updates of portions of pages.

1

AJAX

- Asynchronous JavaScript+XML
- 2nd communication path between browser and web server
 - Remove communication bottleneck bw user and web application
 - Talk to server from JavaScript
 - Skip page reload
- Changes the typical page flow
 - More often requests
 - Smaller responses of non-HTML data
- Defined by Jesse James Garret in Feb. of 2005
 - Underlying technology in place since 2001
 - 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

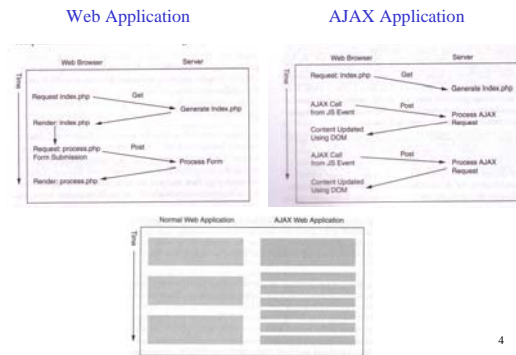
2

Normal Web vs. AJAX

- Traditional web
 - User initiated HTTP requests
 - Typing on navigation window, or clicking on form
 - Response from server overrides existing page
 - Low request rate, and random amount of time between request
- AJAX application
 - New type of request that does not trigger page reload
 - Not initiated by user
 - Requests are typically small, but more frequent

3

Request Flow



4

Taxonomy

- Core AJAX
 - Asynchronous communication (XMLHttpRequest)
 - Data encoding (XML)
- Closely related to AJAX
 - Interactive part of UI
 - Visual effects libraries (fades, swipes)
 - Drag and drop
 - Dynamic positioning
- Part of the rest of RIA
 - Scalable Vector Graphics (SVG)

5

XMLHttpRequest

- JavaScript HTTP client
 - Can load content other than XML
- Issue GET and POST requests
- Security limitation apply
 - Same point of origin; Can only connect to same domain as currently loaded page
 - Only load HTTP requests
 - File URLs don't work

6

XMLHttpRequest Methods

Method	Description
abort()	Cancels the current request.
getAllResponseHeaders()	Returns the complete set of HTTP headers as a string.
getResponseHeader(headerName)	Returns the value of the specified HTTP header.
open(method, URL)	Specifies the method, URL, and other optional attributes of a request.
open(method, URL, async)	The method parameter can have a value of "GET", "POST", "HEAD", "PUT", "DELETE", or a variety of other HTTP methods listed in the W3C specification [1].
open(method, URL, async, username)	The URL parameter may be either a relative or complete URL.
open(method, URL, async, username, password)	The "async" parameter specifies whether the request should be handled asynchronously or not - "true" means that script processing carries on after the send() method, without waiting for a response, and "false" means that the script waits for a response before continuing script processing.
send(content)	Sends the request.
setRequestHeader(label, value)	Adds a label/value pair to the HTTP header to be sent.

7

XMLHttpRequest Properties

Property	Description
onreadystatechange	An event handler for an event that fires at every state change.
readyState	Returns the state of the object as follows: <ul style="list-style-type: none"> 0 = uninitialized 1 = open 2 = sent 3 = receiving 4 = loaded
responseText	Returns the response as a string.
responseXML	Returns the response as XML. This property returns an XML document object, which can be examined and parsed using W3C DOM node tree methods and properties.
status	Returns the HTTP status code as a number (e.g. 404 for "Not Found" and 200 for "OK").
statusText	Returns the status as a string (e.g. "Not Found" or "OK").

8

XMLHttpRequest Events

Firefox Specific

Attribute	Type	Description
onload	onreadystatechange	Called when the HTTP document has been completely loaded. The event sent to the receiver has its target attribute set to the XMLHttpRequest that finished loading.
onerror	onreadystatechange	Called when an error occurs while attempting to retrieve the HTTP document. Check the status attribute of the XMLHttpRequest specified by the target attribute in order to determine which HTTP error occurred.
onprogress	onreadystatechange	Called periodically while the document is being loaded. The event sent to the receiver has a progress attribute indicating how far through the document the process is, as well as a totalSize attribute indicating the total size of the document. In addition, it has its target attribute set to the XMLHttpRequest corresponding to the progress notification.
onreadystatechange	onreadystatechange	Called when the readyState changes. What does this mean?

- Firefox resets handlers after request completes

9

send(payload)

- Makes connection to the URL specified in *open*
- Blocks for synchronous requests, and returns immediately otherwise.
- Sends cookies and other default headers
- POST request:
 - *payload* is sent as body of HTTP message
 - Need to set *Content-type* header
- Example: Mimic sending a form

```
var req = new XMLHttpRequest();
req.open('POST', 'index.php', false);
req.setRequestHeader('Content-type',
    'application/x-www-form-urlencoded');
req.send('one=firstArg&two=secondArg');
if (req.status==200)
    alert(req.responseText);
```

10

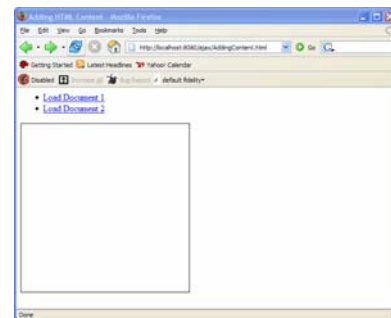
Asynchronous Request

- Happens in the background
 - Does not block browser
 - Browser can continue running JavaScript code
 - User can continue interaction with page
 - Can mask long latency/low bandwidth connections

```
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','/csc309/servlet/ajaxserver',true);
req.send(null);
```

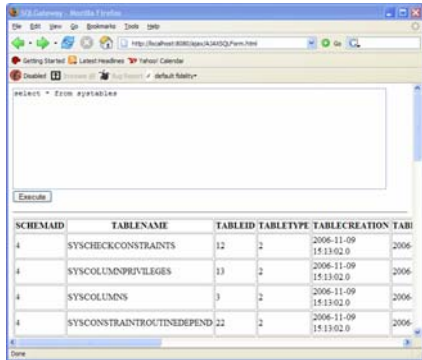
11

Adding HTML to a Page



12

Adding Dynamically Generated HTML to a Page



13

Consuming XML

- Write a XML application for keeping track of student academic history
 - For each student track name and courses taken
 - For each course taken track year and grade
 - For each course store id, name and pre-requisites
- Write web page that:
 - Prints a list of students and their academic history

14

IE AJAX

- XMLHttpRequest is an ActiveX object
 - var xmlhttp = new ActiveXObject('MSXML2.XMLHTTP.5.0');
- Multiple versions
 - Microsoft.XMLHTTP
 - MSXML2.XMLHTTP
 - MSXML2.XMLHTTP.3.0
 - MSXML2.XMLHTTP.4.0
 - MSXML2.XMLHTTP.5.0

15

Cross-Browser XMLHttpRequest

```
function XMLHttpRequest () {
  try {
    // Mozilla / Safari
    xmlhttp = new XMLHttpRequest();
  } catch (e) {
    // IE
    var XMLHTTP_IDS = new Array('MSXML2.XMLHTTP.7.0',
      'MSXML2.XMLHTTP.6.0',
      'MSXML2.XMLHTTP.5.0',
      'MSXML2.XMLHTTP.4.0',
      'MSXML2.XMLHTTP.3.0',
      'MSXML2.XMLHTTP',
      'Microsoft.XMLHTTP');

    var success = false;
    for (var i=0; i < XMLHTTP_IDS.length; i++) {
      try {
        this.xmlhttp = new ActiveXObject(XMLHTTP_IDS[i]);
        success = true;
      } catch (e) {}
    }
    if (!success) {
      throw new Error('Unable to create XMLHttpRequest');
    }
  }
  return xmlhttp;
}
```

16