AJAX (Asynchronous JavaScript And XML)

Nan Niu (nn@cs.toronto.edu)
CSC309 -- Fall 2008

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

AJAX
- 2nd communication path b/w browser and server
  - Remove communication bottleneck b/w user and Web application
  - Talk to server from JavaScript
  - Skip page reload
- Changes the typical page flow
  - More frequent 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

Traditional 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 requests
- AJAX application
  - New type of request that does not trigger page reload
  - Not initiated by user
  - Requests are typically small, but more frequent

Request Flow

Classic vs. AJAX

browser client
HTTP request
HTML+JS+CSS data
Ajax engine
HTTP request
http(s) transport
web server
objects, controls, processing, reply systems
server-side systems
classic web application model

Ajax
http(s) transport
web server
objects, controls, processing, reply systems
server-side systems
Ajax web application model
Synchronous vs. Asynchronous

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)

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

XMLHttpRequest Methods

onreadystatechange Property
- Defines a function to retrieve data returned by the server after a request is sent
- Must be set before sending request
- The following code defines a function for this purpose (with an empty body for now):

```javascript
var xmlhttp = new XMLHttpRequest();
xmlobject.onreadystatechange = function() {
  // code for receiving response data
}
```
readyState Property
• This property holds the status of the server's response
• Each time the readyState changes, the onreadystatechange function will be executed
• State description
  - 0 The request is not initialized
  - 1 The request has been set up
  - 2 The request has been sent
  - 3 The request is in process
  - 4 The request is complete

Update the Function (Event Handler)
```javascript
xmlhttp.onreadystatechange = function()
{
  if(xmlhttp.readyState==4)
  {
    // Get the data from the server's response
  }
}
```

responseText Property
• This property retrieves the response body returned by the server as a string
• Type: DOMString (read-only)
```javascript
xmlhttp.onreadystatechange = function()
{
  if(xmlhttp.readyState==4)
  {
    document.getElementById("formentry").value = xmlhttp.responseText;
  }
}
```

responseXML Property
• Retrieve document data returned by the server as an XML DOM object
• Type: Document (read-only)
```javascript
var xmldoc = xmlhttp.responseXML.documentElement;
• You can access it as a DOM document
```

XMLHttpRequest Methods
• Asking for data (send request)
  - open()
    - Two required arguments
      - method (GET, POST, PUT, DELETE, HEAD, OPTION)
      - server-side URI
  - send()
    - One argument
      - data to be sent (DOMString or Document)
      - null for GET
      - can be omitted

Methods (Verbs)
• GET
  - Retrieve a representation of a resource
• POST
  - Create a new resource
• HEAD
  - Request a resource without body
• PUT
  - Update a resource
• DELETE
  - Delete a resource
Execute the AJAX Function

- Want it to run "behind the scenes"

```javascript
function myajax()
{
    /* all of the code from before */
}
```
Add HTML to a Page

IE AJAX

- XMLHttpRequest is an ActiveX object
  ```javascript
  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

Cross-Browser XMLHttpRequest

```javascript
function XMLHttpRequest() {
  try {
    // Mozilla / Safari / Firefox
    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 && !success; i++) {
      try {
        this xmlhttp = new ActiveXObject(XMLHTTP_IDS[i]);
        success = true;
      } catch (e) {}
    }
    if (!success) {
      throw new Error('Unable to create XMLHttpRequest.');
    }
  }
  return xmlhttp;
}
```

Schedule

- This week
  - Tuesday: lecture
  - Thursday: A3 due (no tutorial)
  - A4 will be posted around Thursday

- Next week
  - Tuesday: no lecture
  - Thursday: A2 re-marking & A4 office hour, in BA 5170