Objectives

- Review Security
- Ajax

Review: Security

- Why has the Web become such a huge target?
- What are some security vulnerabilities for Web applications?
  - How can you protect against them?

Ajax: Asynchronous JavaScript + XML

- Not a programming language
- A way of using JavaScript
- Way to provide more responsive Web pages
  - Get data from a server without reloading your page
- Allows dynamically displaying data or updating the page without disturbing the user experience
- Aids in the creation of rich, user-friendly web sites
  - Examples: Google maps, Facebook, Flickr, A9

Ajax Difference

- In normal request/response HTTP cycles, the browser locks waiting for the response and an entire page must be displayed
- With Ajax, asynchronous requests may be made and responses are used to update part of a page
  - User can continue to interact with a page while the request is in progress
  - Less data needs to be transmitted
  - Page update is quicker because only a part of a page is modified

Core Ajax Concepts

- JavaScript's XMLHttpRequest object can fetch files from a web server
  - Supported in IE5+, Safari, Firefox, Opera (with minor incompatibilities)
- JavaScript can execute XMLHttpRequest asynchronously
  - In the background, transparent to user
- Contents of fetched file can be put into current web page using DOM
  - Reminder: Document Object Model
- Result: user's web page updates dynamically without a page reload

Ajax Request Overview

Could be a Servlet, JSP, PHP, ASP, ...

HTML Page

Replace This

URL

Server

Piece: Identified using DOM
Typical Ajax Request

1. User clicks, invokes event handler
2. Handler's JS code creates an XMLHttpRequest object
3. XMLHttpRequest object requests a document from a web server

4. Server retrieves appropriate data, sends it back
5. XMLHttpRequest fires event when data arrives
   - Called a callback
6. Can attach a handler to be notified when data arrives to parse data, update web page

• Data can be any text format: HTML, XML, Text, ...

XMLHttpRequest Object

• Methods
  - abort, getAllResponseHeaders, getResponseHeader, open, send, setRequestHeader

• Properties
  - onreadystatechange, readyState, responseText, responseXML, status, statusText

Using XMLHttpRequest

• Attach an event handler to the onreadystatechange event
  - Handler will be called when request state changes, e.g., finishes
  - function contains code to run when request is complete
• Replace url with the file you want to download
• Send the request

In an onscreen control's event handler:

```javascript
var ajax = new XMLHttpRequest();
ajax.onreadystatechange = function;
ajax.open("GET", url, true);
ajax.send(null);
```
 XMLHttpRequest’s readyState Property

- Holds the status of the XMLHttpRequest
- Changes value from 0 to 4 during a request cycle:
  - 0: not initialized
  - 1: connection established
  - 2: request sent
  - 3: processing
  - 4: finished and response is ready
- readyState changes ➔ onreadystatechange handler (callback function) runs
- Usually we are only interested in readyState of 4

callback Function Example

```javascript
function processChange() {
  // 4 means the response has been returned, ready to be processed
  if (obj.readyState == 4) {
    // 200 means "OK"
    if (obj.status == 200) {
      // process whatever has been sent back here
    } else {
      alert("There was a problem in the returned data:");
    }
  }
}
```

Browser Compatibility

```javascript
function ProcessXML(url) {
  if (typeof XMLHttpRequest != "undefined") {
    // obtain new object
    obj = new XMLHttpRequest();
    // set the callback function
    obj.onreadystatechange = processChange;
    // as a GET with the url "true" for async
    obj.open("GET", url, true);
    // null for GET with native object
    obj.send(null);
    // IE/Windows ActiveX object
  } else if (window.ActiveXObject) {
    obj = new ActiveXObject("Microsoft.XMLHTTP");
    if (obj) {
      obj.onreadystatechange = processChange;
      obj.open("GET", url, true);
      // don’t send null for ActiveX
      obj.send();
    } else {
      // else browser does not support Ajax
    }
    return obj;
  }
}
```

Ajax XMLHttpRequest template

- Most Ajax code uses an anonymous function as the event handler
  ➔ A function declared inside another and not given a name
  ➔ Useful because it can access the surrounding local variable

```javascript
var ajax = new XMLHttpRequest();
ajax.onreadystatechange = function() {
  if (ajax.readyState == 4) {
    do something with ajax.responseText;
  }
};
ajax.open("GET", url, true);
ajax.send(null);
```

Pet Survey Example

- What happens if a student’s session times out?
  ➔ Will they be able to restart their quiz?
- Ready to apply AuthenticationFilter?
  ➔ Need well-known URLs, Roles

Project Questions
TODO
• Project
  ➢ Implementation deadline Friday
  ➢ Documentation, Demo on Monday
• Evaluations