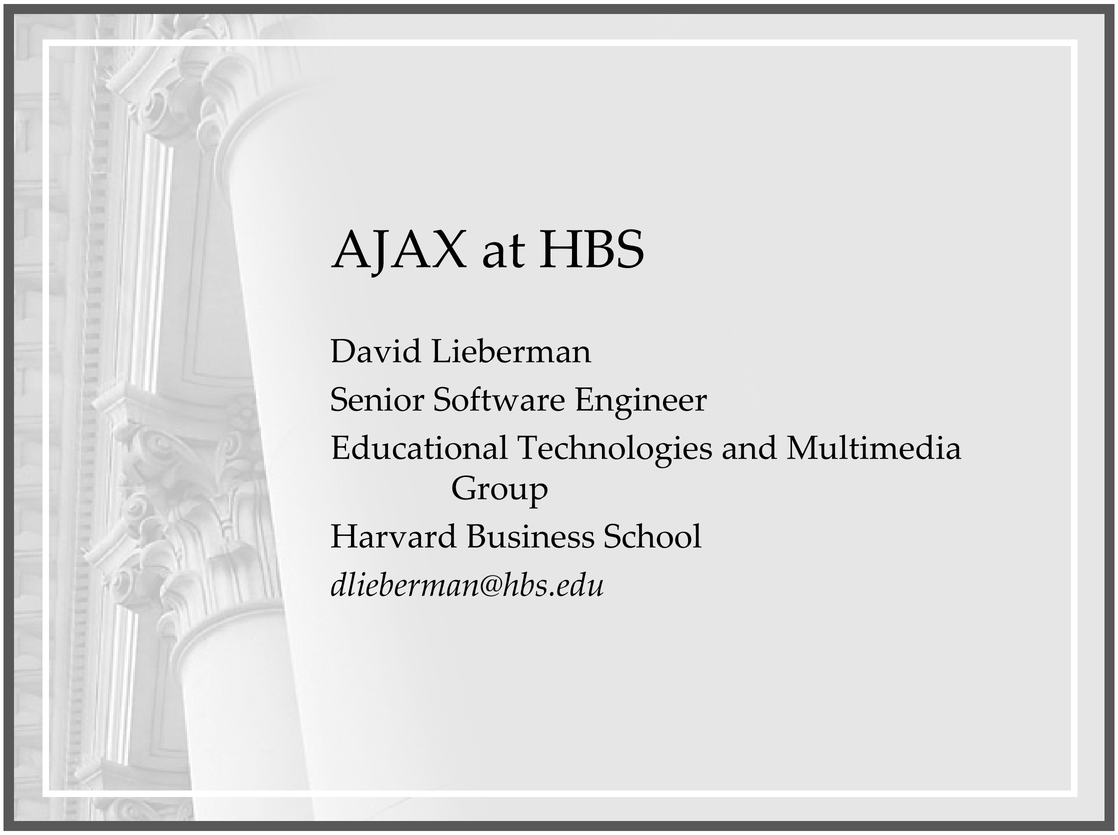

AJAX at HBS

David Lieberman
Senior Software Engineer
Educational Technologies and Multimedia Group
Harvard Business School
dlieberman@hbs.edu

Why bother?

- Performance gain?
- Simpler code?
- Better, more intuitive UI
- Rising user expectations!

My Goals for AJAX

- Take the protocol out of the equation

My Goals for AJAX

- Take the protocol out of the equation
  - Put data exchange in the foreground

My Goals for AJAX

- Take the protocol out of the equation
  - Put data exchange in the foreground
  - If I have to write
    `request.getParameter(“…”)`
    I’m not interested …

My Goals for AJAX

- Take the protocol out of the equation
  - Put data exchange in the foreground
  - If I have to write
    `request.getParameter(“…”)`
    I’m not interested …
- So: XML in, XML out

My Goals for AJAX

- Take the protocol out of the equation
  - Put data exchange in the foreground
  - If I have to write
    `request.getParameter("…")`
    I'm not interested …
- So: XML in, XML out
- Show me the framework!
  - Struts?

XMLHttpRequest properties

- responseText [String]

  Raw response text
- responseXML [XML Object]

  If response text is well-formed XML
- readyState [Integer]
- status [Integer = the HttpResponse code]

XMLHttpRequest methods

- open([String] method,
  [String] url,
  [boolean] asynchronous)

- send([String] arg)

XMLHttpRequest event

- onReadyStateChange

XMLHttpRequest

Set up method variables

```
function get(url, callback) {
   var ajaxrequest;
   var ajaxresponse;
   .  .  .
}
```

XMLHttpRequest

Bind the request callback method

```
function get(url, callback) {
   .  .  .
    function bindRequestChange() {
        if (ajaxrequest.readyState == 4) {
            if (ajaxrequest.status == 200) {
                ajaxresult = ajaxrequest.responseXML;
                eval(callback + "(ajaxresult)");
            } else {
                alert("There was a problem retrieving the XML
                 data: \n" + ajaxresult.statusText);
            }
        }
    }
}
```

XMLHttpRequest

Instantiate the XMLHttpRequest object

```
function get {
   .  .  .

 if (window.XMLHttpRequest) {
   ajaxrequest = new XMLHttpRequest();
 } else if (window.ActiveXObject) {
   ajaxrequest = new ActiveXObject("Microsoft.XMLHTTP");
 }

 if (ajaxrequest) {
   ajaxrequest.onreadystatechange = bindRequestChange;
   ajaxrequest.open("GET", url, true);
   ajaxrequest.send(null);
 }
}
```

Three Implementations at HBS

- Application Access Admin
- Tutorial Platform Authoring
- Videotools Portal Admin
- Controlled environments with small user communities

Application Access Admin

- Utility for controlling access to applications or application modules based on group membership
- AJAX feature:
  - Auto-populating roles selector

Tutorials Platform Authoring

- Tutorials Platform: exposes Flash content to end-users delivered via XML (AJAX competitor alert!)
- Authoring environment: assembling these tutorials element-by-element
- AJAX features:
  - Metadata
  - Tree sorting
  - Rich text editor

Gotchas

- Support!
- Google we are not.
- The Javascript DOM API: more like JAXP than JDOM (cf JDOM + JXPath)
- Safari!

Videotools Portal Admin

- Portal: a stylized, sorted collection of video assets targeted to a specific user community
- Admin requires: sorting, labeling, refreshing, creating RSS feeds.
- Out with XML, in with JSON!

JavaScript Libraries

- Scriptaculous
  http://script.aculo.us/
- Prototype
  http://prototype.conio.net/
- Yahoo! UI Library
  http://developer.yahoo.com/yui/
- Zimbra Toolkit
  http://www.zimbra.com/community/ajaxtk_download.html

Other Resources

- http://www.jdom.org
- http://jakarta.apache.org/commons/jxpath/
- http://www.json.org
- http://www.ajaxian.com