

# AJAX101

Asynchronous JavaScript and XML





# About me

- CTO for iChameleon Group
- Editor-in-Chief of the Ajax Developers Journal
- Advanced ColdFusion Certified
- Frequent contributor of the CFDJ
- Author of ajaxCFC
- Speaking in a number of seminars nationwide



# What is AJAX?

- Asynchronous JavaScript + XML
- Interacting with the server without refreshing the page
- Rich Internet Applications with JavaScript
- A geek marketing term
- Venture Capital Magnet



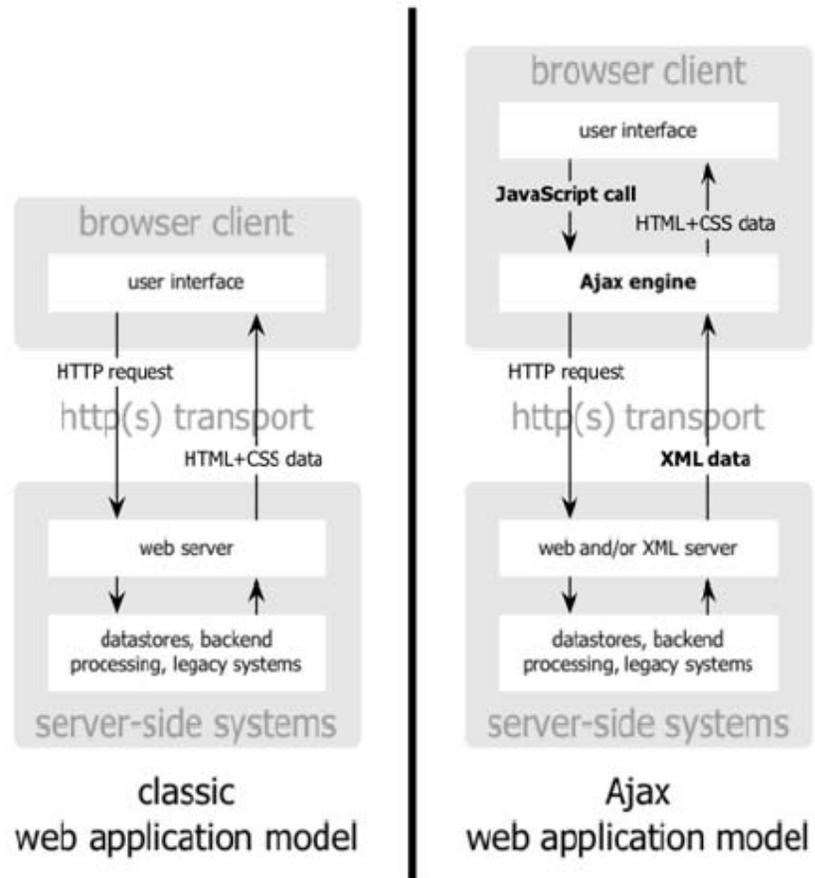
# What is AJAX? (tech view)

- Asynchronous data retrieval using **XMLHttpRequest**
- Data interchange and manipulation using *\_XML\_* (or not)
- Dynamic display and interaction using Document Object Model (DOM)
- JavaScript binding everything together



# Classic Vs. Ajax Web

No more single request / response restrictions





# AJAX Sites

- Google Mail, Maps, Reader, Suggest, Personalized Homepage
- 37signal's Basecamp / Backpack
- Backbase
- Zimbra Collaboration Suite
- Bindows
- Writely



# History – Ajax is not new

- Hidden Frames since mid-90's
- XHR and ActiveX since IE5+, Mozilla 1.0+, Safari 1.2+, and Opera 7.6+
- February 18, 2005: Jesse James Garret wrote the article that coined the AJAX acronym.



# Why is it popular?

- Google helped popularize, and legitimize it in Gmail and Google Maps
- Increase Usability of Web Applications
- Rich Internet Applications without Flash
- Save Bandwidth
- Download only data you need
- Faster interfaces and better user experience



# Potential Problems

- Breaks back button support
- URL's don't change as state changes
- SEO / Search Engine Indexing
- Cross Browser Issues can be a pain
- Cross-domain requests (SOA, WS)
- JavaScript disabled or methods not available on some old browsers



# Flash Vs. AJAX

Feature	Flash	Ajax
Audio	Native Support	Needs plug-ins
Browser Integration	Plug-in required	Native Support
Compatibility Issues	Minor variations between versions of Flash.	Major compatibility differences between browser versions.
CSS	Limited support.	Full Support (depending on browser).
Dynamic Content	Difficult. SWF is a pre-compiled closed format.	HTML can be written out using just about any kind of Server technology.
Programming Model	ActionScript 2.0 provides robust, java-like framework.	JavaScript 2.0 not yet supported by <i>any</i> major browser. JS 1.5 not recommended for large OOP applications.



# Flash Vs. AJAX

Feature	Flash	Ajax
(Bitmap) Graphics	Bitmap manipulation.	Load static images dynamically.
Regular Expressions	Not supported natively by AS 2.0	Full support.
Server Integration	Web Services, XML, Flash Remoting	IFRAME trick or XMLHttpRequest
Text	Text API mimics some HTML functionality.	Powerful layout capabilities.
Vector Graphics	Full Support.	None.
Video	Dynamically load FLV video files or playback embedded videos.	Supported only through external plug-ins
XML	Full support.	Limited Support



# XMLHttpRequest (XHR)

- A JavaScript Class that lets you make asynchronous HTTP requests from JavaScript
- Allows to kick off HTTP requests in the background
- A call back JavaScript function is invoked at each state of the HTTP request and response



# XMLHttpRequest Methods

- **open** – setup a request
- **send** – sends the request
- **abort** – aborts any request
- **getAllResponseHeaders** – Returns a Map of headers
- **getResponseHeader** – Returns a header value
- **setRequestHeader** – Set header



# XMLHttpRequest Properties

- **onreadystatechange** - call back function for state changes
- **readyState** - the current state of the HTTP call
  - 0 = uninitialized, 1 = loading, 2 = loaded, 3 = interactive, 4 = complete
- **responseText** - the text result of the request
- **responseXML** - XML DOM of response
- **status** - HTTP status code of the response
- **statusText** - HTTP status text



# Cross-browser: IE

- `new XMLHttpRequest("Microsoft.XMLHTTP");`
- You can't totally blame them because they invented it
- Native XMLHttpRequest support should be in IE7



# Cross Browser AJAX 1/3

```
var req;
function loadXMLDoc(url) {
    req = false;
    // branch for native XMLHttpRequest object

    if(window.XMLHttpRequest) {
        try {
            req = new XMLHttpRequest();
        } catch(e) {
            req = false;
        }
    }
}
```



# Cross Browser AJAX 2/3

```
// branch for IE/Windows ActiveX version
} else if(window.ActiveXObject) {
  try {
    req = new ActiveXObject("Msxml2.XMLHTTP");
  } catch(e) {
    try {
      req = new ActiveXObject("Microsoft.XMLHTTP");
    } catch(e) {
      req = false;
    }
  }
}
```



# Cross Browser AJAX 3/3

```
if(req) {  
    req.onreadystatechange = processReqChange;  
    req.open("GET", url, true);  
    req.send("");  
}  
}
```



# AJAX Libraries

- Many people opt for AJAX libraries.
- Provides many advantages
  - Sync / Async
  - Multithread
  - Error catching
  - Logging
- Disadvantages
  - Weight
  - Loads more than you need



# JavaScript AJAX Libraries

- Prototype (the most popular)
- Backbase
- Dojo
- DWR



# JavaScript UI Libraries

- Widgets
  - ActiveWidgets
  - Backbase
  - SmartClients
  - Dojo
- Low Level Effects
  - Scriptaculous (req. prototype)
  - Rico (req. prototype)



# ColdFusion AJAX Libraries

- ajaxCFC
  - OOP, CF extends objects
  - Built-in error trapping, security, debugging
- CFAjax
  - First Ajax CF Framework
- JSMX
  - Client side only
- Simple Remote Scripting (SRS)
  - Uses iframes



# Do's and Don'ts

- Do's
  - Use Ajax!
  - Evade unneeded page refreshes
    - Checking if a user exists
    - Checking password
    - Address from zip code
    - Taxes / Shipping estimation
  - Use DHTML for presentation layer and update server with Ajax. i.e. drag and drop shopping carts or list resorting
  - Provide progress indicator. The user is used to page refreshes
  - Pre-fetch data sets to improve speed



# Do's and Don'ts

- Don't's
  - Populate main site's menu.
  - Fetch important content that has to be indexed by search engines
  - Perform some server action without a proper visual representation
  - Delegate business logic to client side (hard to grasp)
  - Overuse XML



# ajaxCFC simple echo

```
<script type='text/javascript'>
  _ajaxConfig = {_cfscriptLocation:'echoTest.cfc', '_jsscriptFolder':'../js'};
</script>
<script type='text/javascript' src='../js/ajax.js'></script>
<script type="text/javascript">
  function doEcho() {
    DWREngine._execute(_ajaxConfig._cfscriptLocation, null, 'echo', $('echoInput'), doEchoResult);
  }

  function doEchoResult (r) {
    $('echoScreen').innerHTML = $('echoScreen').innerHTML + '<BR>' + r;
  }
</script>
<input type="Text" id="echoInput">
<input type="submit" value="enter" onClick='doEcho(); '>
<div id="echoScreen" style="height: 100px;overflow: scroll;"></div>
```

Annotations:

- setup (points to the `<script type='text/javascript' src='../js/ajax.js'></script>` line)
- Ajax call (points to the `function doEcho() {` line)
- Callback function (points to the `function doEchoResult (r) {` line)
- Input field (points to the `<input type="Text" id="echoInput">` line)
- Echo \_screen\_ (points to the `<div id="echoScreen" style="height: 100px;overflow: scroll;"></div>` line)



# ajaxCFC Server Side

```
<cfcomponent extends="ajax">  
  <cffunction name="echo" output="no" access="private">  
    <cfargument name="args" required="Yes" type="array">  
    <cfreturn arguments.args[1] />  
  </cffunction>  
</cfcomponent>
```



# Contact Info

- Blog: <http://www.robgonda.com>
- Email: [rob@robgonda.com](mailto:rob@robgonda.com)