

## What's AJAX ?

AJAX (Asynchronous JavaScript and XML) is a newly coined term for two powerful browser features that have been around for years, but were overlooked by many web developers until recently when applications such as Gmail, Google Suggest, and Google Maps hit the streets.

Asynchronous JavaScript and XML, or Ajax (pronounced "Aye-Jacks"), is a web development technique for creating interactive web applications using a combination of XHTML (or HTML) and CSS for marking up and styling information. (XML is commonly used, although any format will work, including preformatted HTML, plain text, JSON and even EBML).

The Document Object Model manipulated through JavaScript to dynamically display and interact with the information presented

The XMLHttpRequest object to exchange data asynchronously with the web server. In some Ajax frameworks and in some situations, an IFrame object is used instead of the XMLHttpRequest object to exchange data with the web server.

Like DHTML, LAMP, or SPA, Ajax is not a technology in itself, but a term that refers to the use of a group of technologies together. In fact, derivative/composite technologies based substantially upon Ajax, such as AFLAX, are already appearing.

Ajax applications are mostly executed on the user's computer; they can perform a number of tasks without their performance being limited by the network. This permits the development of interactive applications, in particular reactive and rich graphic user interfaces.

Ajax applications target a well-documented platform, implemented by all major browsers on most existing platforms. While it is uncertain that this compatibility will resist the advent of the next generations of browsers (in particular, Firefox), at the moment, Ajax applications are effectively cross-platform.

While the Ajax platform is more restricted than the Java platform, current Ajax applications effectively fill part of the one-time niche of Java applets: extending the browser with portable, lightweight mini-applications.

Ajax isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways. Ajax incorporates:

- \* standards-based presentation using XHTML and CSS;
- \* dynamic display and interaction using the Document Object Model;
- \* data interchange and manipulation using XML and XSLT; \* asynchronous data retrieval using XMLHttpRequest;
- \* and JavaScript binding everything together.

---

## Who's Using Ajax?

Google is making a huge investment in developing the Ajax approach. All of the major products Google has introduced over the last year — Orkut, Gmail, the latest beta version of Google Groups, Google Suggest, and Google Maps — are Ajax applications. (For more on the technical nuts and bolts of these Ajax implementations, check out these excellent analyses of Gmail, Google Suggest, and Google Maps.) Others are following suit: many of the features that people love in Flickr depend on Ajax, and Amazon's A9.com search engine applies similar techniques.

These projects demonstrate that Ajax is not only technically sound, but also practical for real-world applications. This isn't another technology that only works in a laboratory. And Ajax applications can be any size, from the very simple, single-function Google Suggest to the very complex and sophisticated Google Maps.

At Adaptive Path, we've been doing our own work with Ajax over the last several months, and we're realizing we've only scratched the surface of the rich interaction and responsiveness that Ajax applications can provide. Ajax is an important development for Web applications, and its importance is only going to grow. And because there are so many developers out there who already know how to use these technologies, we expect to see many more organizations following Google's lead in reaping the competitive advantage Ajax provides.

## Moving Forward

The biggest challenges in creating Ajax applications are not technical. The core Ajax technologies are mature, stable, and well understood. Instead, the challenges are for the designers of these applications: to forget what we think we know about the limitations of the Web, and begin to imagine a wider, richer range of possibilities

---

## Should I consider AJAX?

AJAX definitely has the buzz right now, but it might not be the right thing for you. AJAX is limited to the latest browsers, exposes browser compatibility issues, and requires new skill-sets for many. There is a good blog entry by Alex Bosworth on [AJAX Mistakes](#) which is a good read before you jump full force into AJAX.

On the other hand you can achieve highly interactive rich web applications that are responsive and appear really fast. While it is debatable as to whether an AJAX based application is really faster, the user feels a sense of immediacy because they are given active feedback while data is exchanged in the background. If you are an early adopter and can handle the browser compatibility issues, and are willing to learn some more skills, then AJAX is for you. It may be prudent to start off AJAX-ifying a small portion or component of your application first. We all love technology, but just remember the purpose of AJAX is to enhance your user's experience and not hinder it.

---

### [Does AJAX work with Java?](#)

Absolutely. Java is a great fit for AJAX! You can use Java Enterprise Edition servers to generate AJAX client pages and to serve incoming AJAX requests, manage server side state for AJAX clients, and connect AJAX clients to your enterprise resources. The JavaServer Faces component model is a great fit for defining and using AJAX components.

---

### [Won't my server-side framework provide me with AJAX?](#)

You may be benefiting from AJAX already. Many existing Java based frameworks already have some level of AJAX interactions and new frameworks and component libraries are being developed to provide better AJAX support. I won't list all the Java frameworks that use AJAX here, out of fear of missing someone, but you can find a good list at [www.ajaxpatterns.org/Java\\_Ajax\\_Frameworks](http://www.ajaxpatterns.org/Java_Ajax_Frameworks).

If you have not chosen a framework yet it is recommended you consider using JavaServer Faces or a JavaServer Faces based framework. JavaServer Faces components can be created and used to abstract many of the details of generating JavaScript, AJAX interactions, and DHTML processing and thus enable simple AJAX used by JSF application developer and as plug-ins in JSF compatible IDE's, such as Sun Java Studio Creator.

---

### [Where should I start?](#)

Assuming the framework you are using does not suffice your use cases and you would like to develop your own AJAX components or functionality I suggest you start with the article [Asynchronous JavaScript Technology and XML \(AJAX\) With Java 2 Platform, Enterprise Edition](#).

If you would like to see a very basic example that includes source code you can check out the tech tip [Using AJAX with Java Technology](#). For a more complete list of AJAX resources the [Blueprints AJAX](#) home page.

Next, I would recommend spending some time investigating AJAX libraries and frameworks. If you choose to write your own AJAX clients-side script you are much better off not re-inventing the wheel.

[AJAX in Action](#) by Dave Crane and Eric Pascarello with Darren James is good resource. This book is helpful for the Java developer in that it contains an appendix for learning JavaScript for the Java developer.

---

### [Did Adaptive Path invent Ajax? Did Google? Did Adaptive Path help build Google's Ajax applications?](#)

Neither Adaptive Path nor Google invented Ajax. Google's recent products are simply the highest-profile examples of Ajax applications. Adaptive Path was not involved in the development of Google's Ajax applications, but we have been doing Ajax work for some of our other clients.

---

### Is it possible to set session variables from javascript?

It's not possible to set any session variables directly from javascript as it is purely a client side technology. You can use AJAX though to asynchronously.

Cannot parse XML generated by JSP I am generating an XML using JSP, when i run the JSP in IE it shows the XML as per DOM, but when i try to parse it using Javascript , the command `xmlDoc.documentElement`.

This is working code I am using, it might help you. `if (isIE) xmlDoc = req.responseXML; else { //IE does not take the responseXML as.`

---

### What do I need to know to create my own AJAX functionality?

If you plan not to reuse and existing AJAX component here are some of the things you will need to know.

Plan to learn Dynamic HTML (DHTML), the technology that is the foundation for AJAX. DHTML enables browser-base real time interaction between a user and a web page. DHTML is the combination of JavaScript, the Document Object Model (DOM) and Cascading Style Sheets (CSS).

\* **JavaScript** - JavaScript is a loosely typed object based scripting language supported by all major browsers and essential for AJAX interactions.

JavaScript in a page is called when an event in a page occurs such as a page load, a mouse click, or a key press in a form element.

\* **DOM** - An API for accessing and manipulating structured documents. In most cases DOM represent the structure of XML and HTML documents.

\* **CSS** - Allows you to define the presentation of a page such as fonts, colors, sizes, and positioning. CSS allow for a clear separation of the presentation from the content and may be changed programmatically by JavaScript.

Understanding the basic request/response nature of HTTP is also important. Many subtle bugs can result if you ignore the differences between the GET and OIst methods when configuring an XMLHttpRequest and HTTP response codes when processing callbacks.

JavaScript is the client-side glue, in a sense. JavaScript is used to create the XMLHttpRequest Object and trigger the asynchronous call. JavaScript is used to parse the returned content. JavaScript is used to analyze the returned data and process returned messages. JavaScript is used to inject the new content into the HTML using the DOM API and to modify the CSS.

---

### Do I really need to learn JavaScript?

Basically yes if you plan to develop new AJAX functionality for your web application.

On the other hand, JSF components and component libraries can abstract the details of JavaScript, DOM and CSS. These components can generate the necessary artifacts to make AJAX interactions possible. Visual tools such as Java Studio Creator may also use AJAX enabled JSF components to create applications, shielding the tool developer from many of the details of AJAX. If you plan to develop your own JSF components or wire the events of components together in a tool it is important that you have a basic understanding of JavaScript. There are client-side JavaScript libraries (discussed below) that you can call from your in page JavaScript that abstract browser differences. Object Hierarchy and Inheritance in JavaScript is a great resource for a Java developer to learn about JavaScript objects.

---

### Do Ajax applications always deliver a better experience than traditional web applications?

Not necessarily. Ajax gives interaction designers more flexibility. However, the more power we have, the more caution we must use in exercising it. We must be careful to use Ajax to enhance the user experience of our applications, not degrade it.

---

### What JavaScript libraries and frameworks are available?

There are many libraries/frameworks out there (and many more emerging) that will help abstract such things as all the nasty browser differences. Three good libraries are The Dojo Toolkit, Prototype, and DWR.

\* The Dojo Toolkit contains APIs and widgets to support the development of rich web applications. Dojo contains an intelligent packaging system, UI effects, drag and drop APIs, widget APIs, event abstraction, client storage APIs, and AJAX interaction APIs. Dojo solves common usability issues such as support for dealing with the navigation such as the ability to detect the browser back button, the ability to support changes to the URL in the URL bar for bookmarking, and the ability to gracefully degrade when AJAX/JavaScript is not fully supported on the client. Dojo is the Swiss Army Knife of JavaScript libraries. It provides the widest range of options in a single library and it does a very good job supporting new and older browsers.

\* Prototype focuses on AJAX interactions including a JavaScript AJAX object that contains a few objects to do basic tasks such as make a request, update a portion of a document, insert content into a document, and update a portion of a document periodically. Prototype JavaScript library contains a set of JavaScript objects for representing AJAX requests and contains utility functions for accessing in page components and DOM manipulations. Script.aculo.us and Rico are built on top of Prototype and provide UI effects, support for drag and drop, and include common JavaScript centric widgets. If you are just looking to support AJAX interactions and a few basic tasks Prototype is great. If you are looking for UI effects Rico and Script.aculo.us are good options.

\* Yahoo UI Library is a utility library and set of widgets using the APIs to support rich clients. The utility library includes support for cross-browser AJAX interactions, animation, DOM scripting support, drag and drop, and cross browser event support. The Yahoo UI Library is well documented and contains many examples.

\* DWR (Dynamic Web Remoting) is a client-side and server-side framework that focuses on allowing a developer to do RPC calls from client-side JavaScript to plain old Java objects in a Java Enterprise Edition web container. On the server side DWR uses a Servlet to interact with the Java objects and returns object representations of the Java objects or XML documents. DWR will be easy to get up and running and plays well with other Java technologies. If you are looking for a client-side and server-side framework that integrates well use DWR.

\* Google Web Toolkit (GWT) is client/server framework provided by Google that allows a developer to write an AJAX application in pure Java. The GWT takes care of the details of generating all the client-side code using a Java-to-JavaScript compiler. One of the key benefits of the GWT Software Developer Kit (SDK) is that it allows you to debug your applications in what is known as GWT hosted mode using an embedded browser (IE on Windows and Mozilla/Gecko on Linux) that is tied to the toolkit. In GWT hosted mode you setup through the code and debug it as it is running on both the client and server. The GWT contains a default set of widgets and widget containers. An application is built by coding a set of widgets and containers together much like would be done in a Swing application. The GWT Software Developer Kit (SDK) is limited to Linux and Windows XP/2000 though the web applications it generates are compatible with the latest generation of the mainstream browsers.

There are many new and emerging libraries for JavaScript and this list only reviews some of the more common libraries. When making a choice choose the library which suites your needs the best. While it might be better to choose one, there is nothing stopping you from using more than one framework. For a more extensive list of client-side frameworks see: [Survey of AJAX/JavaScript Libraries](#).

---

### [What is the difference between proxied and proxyless calls?](#)

Proxied calls are made through stub objects that mimic your PHP classes on the JavaScript side. E.g., the helloworld class from the Hello World example.

Proxyless calls are made using utility javascript functions like `HTML_AJAX.replace()` and `HTML_AJAX.append()`.

---

### [Should I use XML or text, JavaScript, or HTML as a return type?](#)

It depends. Clearly the 'X' in AJAX stands for XML, but several AJAX proponents are quick to point out that nothing in AJAX, per se, precludes using other types of payload, such as, JavaScript, HTML, or plain text.

\* **XML** - Web Services and AJAX seem made for one another. You can use client-side API's for downloading and parsing the XML content from RESTful Web Services. (However be mindful with some SOAP based Web Services architectures the payloads can get quite large and complex, and therefore may be inappropriate with AJAX techniques.)

\* **Plain Text** - In this case server-generated text may be injected into a document or evaluated by client-side logic.

\* **JavaScript** - This is an extension to the plain text case with the exception that a server-side component passes a fragment of JavaScript including JavaScript object declarations. Using the JavaScript `eval()` function you can then create the objects on the client. JavaScript Object Notation (JSON), which is a JavaScript object based data exchange specification, relies on this technique.

\* **HTML** - Injecting server-generated HTML fragments directly into a document is generally a very effective AJAX technique. However, it can be complicated keeping the server-side component in sync with what is displayed on the client.

Mashup is a popular term for creating a completely new web application by combining the content from disparate Web Services and other online API's. A good example of a mashup is [housingmaps.com](http://housingmaps.com) which graphically combines housing want-ads from [craigslist.org](http://craigslist.org) and maps from [maps.google.com](http://maps.google.com).

---

### Are there Usability Issues with AJAX?

The nature of updating a page dynamically using data retrieved via AJAX interactions and DHTML may result in drastically changing the appearance and state of a page. A user might choose to use the browser's back or forward buttons, bookmark a page, copy the URL from the URL bar and share it with a friend via an email or chat client, or print a page at any given time. When designing an AJAX based application you need to consider what the expected behavior would be in the case of navigation, bookmarking, printing, and browser support as described below.

\* Navigation - What would be the expected behavior of the back, forward, refresh, and bookmark browser buttons in your application design. While you could implement history manipulation manually it may be easier to use a JavaScript frameworks such as Dojo that provides API's history manipulation and navigation control.

\* Bookmarking and URL sharing - Many users want to bookmark or cut and paste the URL from the browser bar. Dojo provides client-side for bookmarking and URL manipulation.

\* Printing - In some cases printing dynamically rendered pages can be problematic.

Other considerations as a developer when using AJAX are:

\* Browser Support - Not all AJAX/DHTML features are supported on all browsers or all versions of a browser. See [quirksmode.org](http://quirksmode.org) for a list of browser support and possible workarounds.

\* JavaScript disabled - You should also consider what happens if the user disables JavaScript. Additionally, there are several legitimate reasons why JavaScript and CSS support may be unavailable on a user's web browser.

\* Latency - Keep in mind latency in your design. A running application will be much more responsive than when it is deployed.

---

### Latency problems: myth or reality?

\* Accessibility - Guaranteeing your site is accessible to people with disabilities is not only a noble goal, it is also required by law in many markets. Some marvellous enabling technology is available to help people use the Web in spite of disabilities including visual, auditory, physical, speech, cognitive, and neurological disabilities. With a little forethought, and comprehension of some well documented best practices, you can assure that your application is compatible with that enabling technology.

Degradability is the term used to describe techniques used by web applications to adapt to the wide range of web browser capabilities. Many AJAX libraries have automatic degradability built in. But if you are coding your own custom AJAX functionality, simply taking some care to follow the best practices promoted by standards bodies like the World Wide Web Consortium (W3C), and grass root movements like the Web Standards community and many others, your application can run usefully on browsers that are incapable of AJAX behaviors. Granted, your application may lose some of the "wow factor" on these less capable browsers, but your application will still be usable.

Remember to not design with AJAX just for the sake of coolness. The reason you built your application is so people will use it. And people will not use your application if your application is not compatible with their web browser.

---

### Are there any frameworks available to help speedup development with AJAX?

There are several browser-side frameworks available, each with their own uniqueness...

---

#### [Is Adaptive Path selling Ajax components or trademarking the name? Where can I download it?](#)

Ajax isn't something you can download. It's an approach — a way of thinking about the architecture of web applications using certain technologies. Neither the Ajax name nor the approach are proprietary to Adaptive Path.

---

#### [Should I use an HTTP GET or POST for my AJAX calls?](#)

AJAX requests should use an HTTP GET request when retrieving data where the data will not change for a given request URL. An HTTP POST should be used when state is updated on the server. This is in line with HTTP idempotency recommendations and is highly recommended for a consistent web application architecture.

---

#### [How do I debug JavaScript?](#)

There are not that many tools out there that will support both client-side and server-side debugging. I am certain this will change as AJAX applications proliferate. I currently do my client-side and server-side debugging separately. Below is some information on the client-side debuggers on some of the commonly used browsers.

- \* Firefox/Mozilla/Netscape - Have a built in debugger Venkman which can be helpful but there is a Firefox add on known as FireBug which provides all the information and AJAX developer would ever need including the ability to inspect the browser DOM, console access to the JavaScript runtime in the browser, and the ability to see the HTTP requests and responses (including those made by an XMLHttpRequest). I tend to develop my applications initially on Firefox using Firebug then venture out to the other browsers.

- \* Safari - Has a debugger which needs to be enabled. See the Safari FAQ for details.

- \* Internet Explorer - There is MSDN Documentation on debugging JavaScript. A developer toolbar for Internet Explorer may also be helpful.

While debuggers help a common technique knowing as "Alert Debugging" may be used. In this case you place "alert()" function calls inline much like you would a System.out.println. While a little primitive it works for most basic cases. Some frameworks such as Dojo provide APIs for tracking debug statements.

---

#### [How do I provide internationalized AJAX interactions?](#)

Just because you are using XML does not mean you can properly send and receive localized content using AJAX requests. To provide internationalized AJAX components you need to do the following:

- \* Set the charset of the page to an encoding that is supported by your target languages. I tend to use UTF-8 because it covers the most languages.

The following meta declaration in a HTML/JSP page will set the content type:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

- \* In the page JavaScript make sure to encode any parameters sent to the server. JavaScript provides the escape() function which returns Unicode escape strings in which localized text will appear in hexadecimal format. For more details on JavaScript encoding see Comparing escape(), encodeURIComponent(), and encodeURIComponent().

- \* On the server-side component set the character encoding using the HttpServletRequest.setCharacterEncoding() method. Before you access the localized parameter using the HttpServletRequest.getParameter() call. In the case of UTF this would be request.setCharacterEncoding("UTF-8");. A server-side component returning AJAX responses needs to set the encoding of the response to the same encoding used in the page.  
response.setContentType("text/xml; charset=UTF-8");  
response.getWriter().write("<response>invalid </response>");

For more information on using AJAX with Java Enterprise Edition technologies see [AJAX and Internationalization](#) and for developing multi-lingual applications see [Developing Multilingual Web Applications Using JavaServer Pages Technology](#).

---

[Some of the Google examples you cite don't use XML at all. Do I have to use XML and/or XSLT in an Ajax application?](#)

No. XML is the most fully-developed means of getting data in and out of an Ajax client, but there's no reason you couldn't accomplish the same effects using a technology like JavaScript Object Notation or any similar means of structuring data for interchange.

---

[Are Ajax applications easier to develop than traditional web applications?](#)

Not necessarily. Ajax applications inevitably involve running complex JavaScript code on the client. Making that complex code efficient and bug-free is not a task to be taken lightly, and better development tools and frameworks will be needed to help us meet that challenge.

---

[When do I use a synchronous versus a asynchronous request?](#)

Good question. They don't call it AJAX for nothing! A synchronous request would block in page event processing and I don't see many use cases where a synchronous request is preferable.

---

[How do I handle concurrent AJAX requests?](#)

With JavaScript you can have more than one AJAX request processing at a single time. In order to insure the proper post processing of code it is recommended that you use JavaScript Closures. The example below shows an XMLHttpRequest object abstracted by a JavaScript object called AJAXInteraction. As arguments you pass in the URL to call and the function to call when the processing is done.

```
function AJAXInteraction(url, callback) {
    var req = init();
    req.onreadystatechange = processRequest;

    function init() {
        if (window.XMLHttpRequest) {
            return new XMLHttpRequest();
        } else if (window.ActiveXObject) {
            return new ActiveXObject("Microsoft.XMLHTTP");
        }
    }

    function processRequest () {
        if (req.readyState == 4) {
            if (req.status == 200) {
                if (callback) callback(req.responseXML);
            }
        }
    }

    this.doGet = function() {
        req.open("GET", url, true);
```

```

req.send(null);
}

this.doPost = function(body) {
req.open("POST", url, true);
req.setRequestHeader("Content-Type", "
application/x-www-form-urlencoded");
req.send(body);
}
}

function makeRequest() {
var ai = new AJAXInteraction("processme",
function() { alert("Doing Post Process");});
ai.doGet();
}

```

The function makeRequest() in the example above creates an AJAXInteraction with a URL to of "processme" and an inline function that will show an alert dialog with the message "Doing Post Process". When ai.doGet() is called the AJAX interaction is initiated and when server-side component mapped to the URL "processme" returns a document which is passed to the callback function that was specified when the AJAXInteraction was created.

Using this closures insures that the proper callback function associated with a specific AJAX interaction is called. Caution should still be taken when creating multiple closure objects in that make XmlHttpRequests as to there is a limited number of sockets that are used to make requests at any given time. Because there are limited number of requests that can be made concurrently. Internet Explorer for example only allows for two concurrent AJAX requests at any given time. Other browsers may allow more but it is generally between three and five requests. You may choose to use pool of AJAXInteraction objects.

One thing to note when making multiple AJAX calls from the client is that the calls are not guaranteed to return in any given order. Having closures within the callback of a closure object can be used to ensure dependencies are processed correctly.

There is a discussion titled Ajaxian Fire and Forget Pattern that is helpful.

---

### What do I do on the server to interact with an AJAX client?

The "Content-Type" header needs to be set to "text/xml". In servlets this may be done using the HttpServletResponse.setContentType() should be set to "text/xml" when the return type is XML. Many XMLHttpRequest implementations will result in an error if the "Content-Type" header is set. The code below shows how to set the "Content-Type".

```

response.setContentType("text/xml");
response.getWriter().write("<response>invalid</response>");

```

You may also want to set whether or not to set the caches header for cases such as autocomplete where you may want to notify proxy servers and browsers not to cache the results.

```

response.setContentType("text/xml");
response.setHeader("Cache-Control", "no-cache");
response.getWriter().write("<response>invalid</response>");

```

Note to the developer: Internet Explorer will automatically use a cached result of any AJAX response from a HTTP GET if this header is not set which can make things difficult for a developer. During development mode you may want to set this header. Where do I store state with an AJAX client

As with other browser based web applications you have a few options which include:

- \* On the client in cookies - The size is limited (generally around 4KB X 20 cookies per domain so a total of 80KB) and the content may not be secure unless encrypted which is difficult but not impossible using JavaScript.



- \* On the client in the page - This can be done securely but can be problematic and difficult to work with. See my blog entry on Storing State on the Client for more details on this topic.
  - \* On the client file system - This can be done if the client grants access to the browser to write to the local file system. Depending on your use cases this may be necessary but caution is advised.
  - \* On the Server - This is closer to the traditional model where the client view is of the state on the server. Keeping the data in sync can be a bit problematic and thus we have a solution Refreshing Data on this. As more information processing and control moves to the client where state is stored will need to be re-evaluated.
- 

### What's with the -alpha in the install instructions?

HTML\_AJAX hasn't had a stable release yet and the pear installer doesn't install non stable packages by default unless you specify a version.

---

### How do I submit a form or a part of a form without a page refresh?

When creating a form make sure that the "form" element "onSubmit" attribute is set to a JavaScript function that returns false.

```
<form onSubmit="doAJAXSubmit();return false;" >
<input type="text" id="tf1" />
<input type="submit" id="submit1" value="Update"/>
</>
```

*You can also submit data by associating a function with a form button in a similar way.*

```
<form onSubmit="doAJAXSubmit();return false;" >
<input type="text" id="tf1" />
<input type="button" id="button1" onClick="doAJAXSubmit()" value="Update"/>
</>
```

Note that the form "onSubmit" attribute is still set. If the user hits the enter key in the text field the form will be submitted so you still need to handle that case.

When updating the page it is recommended you wait to make sure that the AJAX update of the form data was successful before updating the data in the page. Otherwise, the data may not properly update and the user may not know. I like to provide an informative message when doing a partial update and upon a successful AJAX interaction I will then update the page.

---

### How do I test my AJAX code?

There is a port of JUnit for client-side JavaScript called JsUnit

---

### What exactly is the W3C DOM?

The W3C Document Object Model (DOM) is defined by the W3C as the following: The Document Object Model is a platform- and language-neutral interface...

---

### When will HTML\_AJAX have a stable release?

Once all the major features are complete and the API has been tested, the roadmap gives an idea of whats left to be done.

---

### What parts of the HTML\_AJAX API are stable?

We don't have a list right now, but most of the API is stable as of 0.3.0. There should be no major changes at this point, though there will be lots of new additions.

---

### What Browsers does HTML\_AJAX work with?

As of 0.3.0, all the examples that ship with HTML\_AJAX have been verified to work with

- \* Firefox 1.0+
- \* Internet Explorer 5.5+ (5.0 should work but it hasn't been tested)

Most things work with

- \* Safari 2+
  - \* Opera 8.5+
- 

### Is the server or the client in control?

It depends. With AJAX the answer is more in between. Control can be more centralized in a server-side component or as a mix of client-side and server-side controllers.

- \* Centralized server-side controller - When having a more centralized controller the key is to make sure the data in client-side page is in sync with that of the server. Some applications may keep all the state on the server and push all updates to client DOM via a simple JavaScript controller.
- \* Client and server-side controllers - This architecture would use JavaScript to do all presentation related control, event processing, page manipulation, and rendering of model data on the client. The server-side would be responsible for things such as business logic and pushing updated model data to the client. In this case the server would not have intimate knowledge of the presentation short of the initial page that would be sent to the client page request.

There are some use cases where an entire AJAX application can be written in a single page. Keep in mind if you choose this type of architecture that navigation and bookmarking should be considered.

Both methods are viable depending on what you are trying to accomplish. I tend to prefer spreading the control across the client and server.

---

### Is Ajax just another name for XMLHttpRequest?

No. XMLHttpRequest is only part of the Ajax equation. XMLHttpRequest is the technical component that makes the asynchronous server communication possible; Ajax is our name for the overall approach described in the article, which relies not only on XMLHttpRequest, but on CSS, DOM, and other technologies.

---

### How do I abort the current XMLHttpRequest?

Just call the abort() method on the request.

---

### What is the minimum version of PHP that needs to be running in order to use HTML\_AJAX?

The oldest PHP version i've fully tested HTML\_AJAX is 4.3.11, but it should run on 4.2.0 without any problems. (Testing reports from PHP versions older than 4.3.11 would be appreciated.)

---

### Why does HTML\_AJAX hang on some server installs

If you run into an HTML\_AJAX problem only on some servers, chances are your running into a problem with output compression. If the output compression is handled in the PHP config we detect that and do the right thing, but if it's done from an apache extension we have no way of knowing its going to compress the body. Sometimes setting HTML\_AJAX::sendContentLength to false fixes the problem, but in other cases you'll need to disabled the extension for the AJAX pages.

I've also seen problems caused by debugging extensions like XDebug, disabling the extension on the server page usually fixes that. Questions dealing with Using HTML\_AJAX, and general JavaScript development

---

### How do I get the XMLHttpRequest object?

Depending upon the browser... if (window.ActiveXObject) { // Internet Explorer http\_request = new ActiveXObject("Microsoft.XMLHTTP"); } else if...

---

### Are there any security issues with AJAX?

JavaScript is in plain view to the user with by selecting view source of the page. JavaScript can not access the local file system without the user's permission. An AJAX interaction can only be made with the servers-side component from which the page was loaded. A proxy pattern could be used for AJAX interactions with external services.

You need to be careful not to expose your application model in such as way that your server-side components are at risk if a nefarious user to reverse engineer your application. As with any other web application, consider using HTTPS to secure the connection when confidential information is being exchanged.

---

### What about applets and plugins?

Don't be too quick to dump your plugin or applet based portions of your application. While AJAX and DHTML can do drag and drop and other advanced user interfaces there still limitations especially when it comes to browser support. Plugins and applets have been around for a while and have been able to make AJAX like requests for years. Applets provide a great set of UI components and APIs that provide developers literally anything.

Many people disregard applets or plugins because there is a startup time to initialize the plugin and there is no guarantee that the needed version of a plugin of JVM is installed. Plugins and applets may not be as capable of manipulating the page DOM. If you are in a uniform environment or can depend on a specific JVM or plugin version being available (such as in a corporate environment) a plugin or applet solution is great.

One thing to consider is a mix of AJAX and applets or plugins. Flickr uses a combination of AJAX interactions/DHTML for labeling pictures and user interaction and a plugin for manipulating photos and photo sets to provide a great user experience. If you design your server-side components well they can talk to both types of clients.

---

Why did you feel the need to give this a name?

I needed something shorter than “Asynchronous JavaScript+CSS+DOM+XMLHttpRequest” to use when discussing this approach with clients.

---

Is AJAX code cross browser compatible?

Not totally. Most browsers offer a native XMLHttpRequest JavaScript object, while another one (Internet Explorer) require you to get it as an ActiveX object..

---

Techniques for asynchronous server communication have been around for years. What makes Ajax a “new” approach?

What’s new is the prominent use of these techniques in real-world applications to change the fundamental interaction model of the Web. Ajax is taking hold now because these technologies and the industry’s understanding of how to deploy them most effectively have taken time to develop.

---

Is Ajax a technology platform or is it an architectural style?

It’s both. Ajax is a set of technologies being used together in a particular way.

---

How do I handle the back and forward buttons?

While you could go out and create a custom solution that tracks the current state on your application I recommend you leave this to the experts. Dojo addresses the navigation in a browser neutral way as can be seen in the JavaScript example below.

```
function updateOnServer(oldId, oldValue,
    itemId, itemValue) {
    var bindArgs = {
        url: "faces/ajax-dlabel-update",
        method: "post",
        content: {"component-id": itemId, "component-value":
            itemValue},
        mimetype: "text/xml",
        load: function(type, data) {
            processUpdateResponse(data);
        },
        backButton: function() {
            alert("old itemid was " + oldId);
        },
        forwardButton: function(){
            alert("forward we must go!");
        }
    };
    dojo.io.bind(bindArgs);
```

```
}
```

The example above will update a value on the server using `dojo.io.bind()` with a function as a property that is responsible for dealing with the browser back button event. As a developer you are capable of restoring the value to the `oldValue` or taking any other action that you see fit. The underlying details of how the browser button event are detected are hidden from the developer by Dojo.

AJAX: How to Handle Bookmarks and Back Buttons details this problem and provides a JavaScript library Really Simple History framework (RSH) that focuses just on the back and forward issue.

---

#### [How does HTML\\_AJAX compare with the XAJAX project at Sourceforge?](#)

XAJAX uses XML as a transport for data between the webpage and server, and you don't write your own javascript data handlers to manipulate the data received from the server. Instead you use a php class and built in javascript methods, a combination that works very similar to the `HTML_AJAX_Action` class and `haSerializer` combo. XAJAX is designed for simplicity and ease of use.

`HTML_AJAX` allows for multiple transmission types for your ajax data - such as urlencoding, json, phpserialized, plain text, with others planned, and has a system you can use to write your own serializers to meet your specific needs. `HTML_AJAX` has a class to help generate javascript (`HTML_AJAX_Helper`) similar to ruby on rail's javascript helper (although it isn't complete), and an action system similar to XAJAX's "action pump" that allows you to avoid writing javascript data handlers if you desire.

But it also has the ability to write your own data handling routines, automatically register classes and methods using a server "proxy" script, do different types of callbacks including grabbing remote urls, choose between sync and async requests, has iframe xmlhttprequest emulation fallback capabilities for users with old browsers or disabled activeX, and is in active development with more features planned (see the Road Map for details)

`HTML_AJAX` has additional features such as client pooling and priority queues for more advanced users, and even a javascript utility class.

Although you can use `HTML_AJAX` the same way you use XAJAX, the additional features make it more robust, extensible and flexible. And it is a pear package, you can use the pear installer to both install and keep it up to date.

If you're asking which is "better" - as with most php scripts it's a matter of taste and need. Do you need a quick, simple ajax solution? Or do you want something that's flexible, extensible, and looking to incorporate even more great features? It depends on the project, you as a writer, and your future plans.

---

#### [What browsers support AJAX?](#)

Internet Explorer 5.0 and up, Opera 7.6 and up, Netscape 7.1 and up, Firefox 1.0 and up, Safari 1.2 and up, among others.

---

#### [How do I send an image using AJAX?](#)

While it may appear that images are being sent when using AJAX with an application like Google Maps what is really happening is that the URLs of images are being sent as the response of an AJAX request and those URLs are being set using DHTML.

In this example an XML document is returned from an AJAX interaction and the category bar is populated.

```
<categories>
<category>
<cat-id>1</cat-id>
<name>Books</name>
<description>Fun to read</description>
<image-url>books_icon.gif</image-url>
</category>
<category>
<cat-id>2</cat-id>
```

```

<name>Electronics</name>
<description>Must have gadgets</description>
<image-url>electronics.gif</image-url>
</category>
</categories>

```

Notice that the image-url element contains the location of the URL for the image representing a category. The callback method of an AJAX interaction will parse the response XML document and call the addCategory function for each category included in the response XML document. The addCategory function looks up a table row element "categoryTable" in body of the page and adds a row to the element which contains the image.

```

<scrip type="text/javascript" >
...
function addCategory(id, name, imageSrc) {
var categoryTable = document.getElementById("categoryTable");
var row = document.createElement("tr");
var catCell = document.createElement("td");
var img = document.createElement("img");
img.src = ("images\\" + imageSrc);
var link = document.createElement("a");
link.className = "category";
link.appendChild(document.createTextNode(name));
link.setAttribute("onclick", "catalog?command=category&catid=" + id);
catCell.appendChild(img);
catCell.appendChild(link);
row.appendChild(catCell);
categoryTable.appendChild(row);
}
</script>
...
<table>
<tr>
<td width="300" bgcolor="lightGray">
<table id="categoryTable" border="0" cellpadding="0"></table>
</td>
<td id="body" width="100%">Body Here</td>
</tr>
</table>

```

Note that the source of the image is set to the image source. The image is loaded by a subsequent HTTP request for the image at the URL "images/books\_icon.gif" or "images/electronic\_icon.gif" that occurs when the img element is added to the categoryTable.

---

[Will HTML\\_AJAX integrate with other Javascript AJAX libraries such as scriptaculous ? How would this integration look like?](#)

HTML\_AJAX doesn't have specific plans to integrate with other JavaScript libraries. Part of this is because external dependencies make for a more complicated installation process. It might make sense to offer some optional dependencies on a library like scriptaculous automatically using its visual effects for the loading box or something, but there isn't a lot to gain from making default visuals like that flashier since they are designed to be easily replaceable.

Most integration would take place in higher level components. Its unclear whether higher level components like that should be part of HTML\_AJAX delivered through PEAR or if they should just be supported by HTML\_AJAX and made available from <http://htmlajax.org> or some other site. If your interested in building widgets or components based on HTML\_AJAX please let me know.

HTML\_AJAX does however offer the ability to use its library loading mechanism with any JavaScript library. I use scriptaculous in conjunction with HTML\_AJAX and I load both libraries through the server.

To do this you just need to register the library with your server and load add its flag to your include line.

```
<?php
$this->server->registerJSLibrary('scriptaculous',
array('prototype.js','scriptaculous.js','builder.js','effects.js','dragdrop.js','controls.js','slider.js'), '/path/to/scriptaculous/');?>
<script type="text/javascript" src="server.php?client=scriptaculous"></script>
```

---

### When should I use an Java applet instead of AJAX?

Applets provide a rich experience on the client side and there are many things they can do that an AJAX application cannot do, such as custom data streaming, graphic manipulation, threading, and advanced GUIs. While DHTML with the use of AJAX has been able to push the boundaries on what you can do on the client, there are some things that it just cannot do. The reason AJAX is so popular is that it only requires functionality built into the browser (namely DHTML and AJAX capabilities). The user does not need to download and/or configure plugins. It is easy to incrementally update functionality and know that that functionality will readily available, and there are not any complicated deployment issues. That said, AJAX-based functionality does need to take browser differences into consideration. This is why we recommend using a JavaScript library such as Dojo which abstracts browser differences. So the "bottom line" is: If you are creating advanced UIs where you need more advanced features on the client where you want UI accuracy down to the pixel, to do complex computations on the client, use specialized networking techniques, and where you know that the applet plugin is available for your target audience, applets are the way to go. AJAX/DHTML works well for applications where you know the users are using the latest generation of browsers, where DHTML/AJAX "good enough" for you, and where your developers have JavaScript/DHTML/AJAX skills. Many amazing things can be done with AJAX/DHTML but there are limitations. AJAX and applets can be used together in the same UIs with AJAX providing the basic structure and applets providing more advanced functionality. The Java can communicate to JavaScript using the Live-Connect APIs. The question should not be should framed as do I use AJAX or applets, but rather which technology makes the best sense for what you are doing. AJAX and applets do not have to be mutually exclusive.

---

### What kinds of applications is Ajax best suited for?

We don't know yet. Because this is a relatively new approach, our understanding of where Ajax can best be applied is still in its infancy. Sometimes the traditional web application model is the most appropriate solution to a problem.

---

### Does this mean Adaptive Path is anti-Flash?

Not at all. Macromedia is an Adaptive Path client, and we've long been supporters of Flash technology. As Ajax matures, we expect that sometimes Ajax will be the better solution to a particular problem, and sometimes Flash will be the better solution. We're also interested in exploring ways the technologies can be mixed (as in the case of Flickr, which uses both).

---

### Where can I find examples of AJAX?

While components of AJAX have been around for some time (for instance, 1999 for XMLHttpRequest), it really didn't become that popular until Google took.

---

## What is the XMLHttpRequest object?

It offers a non-blocking way for JavaScript to communicate back to the web server to update only part of the web page.

---

Does Ajax have significant accessibility or browser compatibility limitations? Do Ajax applications break the back button? Is Ajax compatible with REST? Are there security considerations with Ajax development? Can Ajax applications be made to work for users who have JavaScript turned off?

The answer to all of these questions is “maybe”. Many developers are already working on ways to address these concerns. We think there’s more work to be done to determine all the limitations of Ajax, and we expect the Ajax development community to uncover more issues like these along the way.

---

## How do I access data from other domains to create a mashup with Java?

From your JavaScript clients you can access data in other domains if the return data is provide in JSON format. In essence you can create a JavaScript client that runs operates using data from a different server. This technique is know as JSON with Padding or JSONP. There are questions as to whether this method is secure as you are retrieving data from outside your domain and allowing it to be excuted in the context of your domain. Not all data from third parties is accessible as JSON and in some cases you may want an extra level of protection. With Java you can provide a proxy to third party services using a web component such as a servlet. This proxy can manage the communication with a third party service and provide the data to your clients in a format of your choosing. You can also cache data at your proxy and reduce trips to service. For more on using a Java proxy to create mashups see The XmlHttpProxy Client for Java.

---

## Does Java have support for Comet style server-side push?

Current AJAX applications use polling to communicate changes data between the server and client. Some applications, such as chat applications, stock tickers, or score boards require more immediate notifications of updates to the client. Comet is an event based low latency server side push for AJAX applications. Comet communication keeps one of the two connections available to the browser open to continously communicate events from the server to the client. A Java based solution for Comet is being developed for Glassfish on top of the Grizzly HTTP connector. See Enabling Grizzly by Jean-Francois Arcand for more details.

---

## How do I create a thread to do AJAX polling?

JavaScript does not have threads. JavaScript functions are called when an event happens in a page such as the page is loaded, a mouse click, or a form element gains focus. You can create a timer using the setTimeout which takes a function name and time in milliseconds as arguments. You can then loop by calling the same function as can be seen in the JavaScript example below.

```
function checkForMessage() {  
    // start AJAX interaction with processCallback as the callback function  
}  
  
// callback for the request  
function processCallback() {  
  
    // do post processing  
    setTimeout("checkForMessage()", 10000);  
}
```



```
}
```

Notice that the `checkForMessage` will continue to loop indefinitely. You may want to vary the increment the interval based on activity in the page or your use cases. You may also choose to have logic that would break out of the loop based on some AJAX response processing condition.

---

Is the `XMLHttpRequest` object part of a W3C standard?

No. Or not yet. It is part of the DOM Level 3 Load and Save Specification proposal.

---