

[What is AJAX?]

- **A**ynchronous
- **J**avaScript
- **A**nd
- **X**MLHttpRequest

[AJAX is..]

- **A**ynchronous - get only the parts of a web page that needs to be updated from the web server.
- **J**avaScript - by using Javascript to make the calls
- **A**nd
- **X**MLHttpRequest - how we communicate with the web server

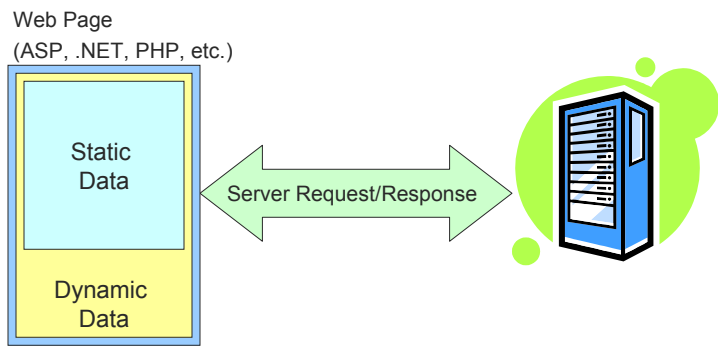
[Anatomy of an AJAX App]

- standards-based presentation using XHTML and CSS;
- dynamic display and interaction using the Document Object Model;
- asynchronous data retrieval using XMLHttpRequest;
- and JavaScript binding everything together

[Why Use AJAX?]

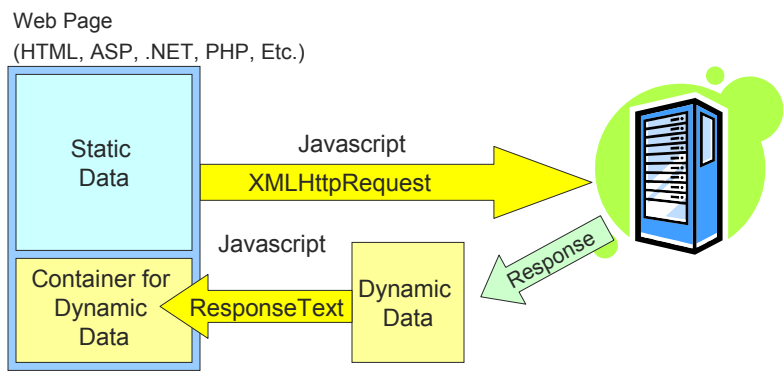
- It allows web apps to behave much like desktop apps.
- It can reduce web server traffic by eliminating the need to re-send static page elements..
- It's cross-platform.
- All the cool kids are using it.

[Traditional Model]



In the traditional model, a request sent to the server returns the static, unchanged data along with the dynamic data.

[AJAX model]



In the AJAX model, a request sent to the server returns only the dynamic data. Javascript is used to send the request, fetch the returned data and insert it into the web page.

[Demos]

- Phone List Search –
<http://www.adsa.dshs.wa.gov/AJAX/noAJAX.asp>
- Phone List Search –
<http://www.adsa.dshs.wa.gov/AJAX/demo1.htm>
- Phone List Search with Javascript event handling -
<http://www.adsa.dshs.wa.gov/AJAX/suggest.htm>

[The Code]

```
function loadurl(dest)
{
    try {
        //first, let's create an object to store our requested page
        // Moz supports XMLHttpRequest. IE uses ActiveX. create HTTP request object depending on browser
        // type
        xmlhttp = window.XMLHttpRequest ? new XMLHttpRequest(): new
        ActiveXObject("Microsoft.XMLHTTP");
    } catch (e) { // if the browser doesn't support AJAX, do something else.. }

    // open takes in the HTTP method and url. i.e. - get the URL of the page (dest) you want to open
    xmlhttp.open("GET", dest + "?lname=" + document.forms[0].lname.value);

    // the xmlhttp object triggers an event everytime the status changes
    //the event is stored in the onreadystatechange property
    // if the property has changed, handle it with our triggered function
    xmlhttp.onreadystatechange = triggered;

    // send the request.
    xmlhttp.send(null); //if using POST, we would enter the post variables here, otherwise, send NULL.
    return false;
}
```

[Code Cont'd..]

```
function triggered()
{
    // if the readyState code is 4 (Page Load Completed)
    // and http status is 200 (OK) we go ahead and get the responseText
    // ResponseText is basically a fancy way of saying get content of the page that was requested in text
    // format
    // other readyState codes:
    // 0=Uninitialised 1=Loading 2=Loaded 3=Interactive

    if ((xmlhttp.readyState == 4) && (xmlhttp.status == 200))
    {
        // xmlhttp.responseText object contains the response.

        //populate the div with the text that was fetched from the (dest) page
        document.getElementById("output").innerHTML = xmlhttp.responseText;
    }
}
```

[The HTML]

```
<form name="frmSort" method="get" action="phonesearch.asp">
  <p>Search Phone List</p>
  <p>Enter Partial Lastname: <input type="text" name="lname" size="20" maxlength="50" />
  <input name="submit" type="submit" value="Go!" onkeyup="return loadurl('phonesearch.asp');" />
</p>
</form>
<div id="output" style="background: #ffc; padding: 5px; border: 1px #f00 solid;">
<p><em>Type a name</em></p> <!--only this part of the page will be updated-->
</div>
```

[A Few Caveats]

- AJAX is Javascript, you must have fallback code that handles non-JS enabled browsers.
- Can break the “Back Button”, but workarounds are available.
- Clear user feedback should be considered in cases of extreme network latency issues. E.g. – “Content loading...please wait.”
 - This can be accomplished with the “readyState” and “status” values.