**INT106 – Lab – AJAX**

\*Credit: [www.w3schools.com](http://www.w3schools.com)

**What is AJAX?**

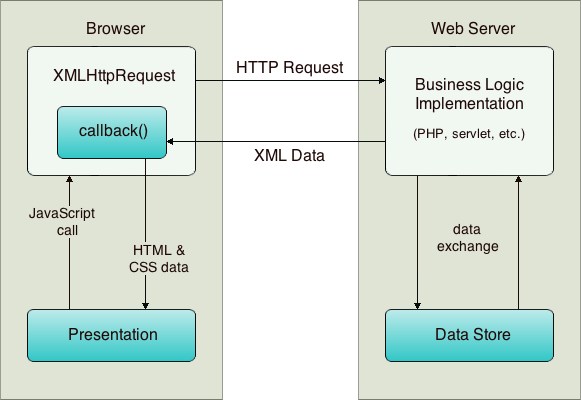
AJAX = Asynchronous JavaScript and XML.

AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.



(image from https://netbeans.org/kb/docs/web/ajax-quickstart.html)

**AJAX Example**

<http://www.w3schools.com/ajax/tryit.asp?filename=tryajax_first>

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script>  **function loadXMLDoc()**  **{**  **var xmlhttp;**  **if (window.XMLHttpRequest)**  **{// code for IE7+, Firefox, Chrome, Opera, Safari**  **xmlhttp=new XMLHttpRequest();**  **}**  **else**  **{// code for IE6, IE5**  **xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");**  **}**  **xmlhttp.onreadystatechange=function()**  **{**  **if (xmlhttp.readyState==4 && xmlhttp.status==200)**  **{**  **document.getElementById("myDiv").innerHTML=xmlhttp.responseText;**  **}**  **}**  **xmlhttp.open("GET","ajax\_info.txt",true);**  **xmlhttp.send();**  **}**  </script>  </head>  <body>  <div id="myDiv"><h2>Let AJAX change this text</h2></div>  <button type="button" onclick="loadXMLDoc()">Change Content</button>  </body>  </html> |

**AJAX Example Explained**

The AJAX application above contains one div section and one button.

The div section will be used to display information returned from a server. The button calls a function named loadXMLDoc(), if it is clicked:

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <div id="myDiv"><h2>Let AJAX change this text</h2></div>  <button type="button" onclick="loadXMLDoc()">Change Content</button>  </body>  </html> |

Next, add a <script> tag to the page's head section. The script section contains the loadXMLDoc() function:

|  |
| --- |
| <head>  <script>  function loadXMLDoc()  {  .... AJAX script goes here ...  }  </script>  </head> |

**AJAX - Create an XMLHttpRequest Object**

**The XMLHttpRequest Object**

All modern browsers support the XMLHttpRequest object (IE5 and IE6 use an ActiveXObject).

The XMLHttpRequest object is used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

**Create an XMLHttpRequest Object**

All modern browsers (IE7+, Firefox, Chrome, Safari, and Opera) have a built-in XMLHttpRequest object.

Syntax for creating an XMLHttpRequest object:

|  |
| --- |
| variable=new XMLHttpRequest(); |

Old versions of Internet Explorer (IE5 and IE6) uses an ActiveX Object:

|  |
| --- |
| variable=new ActiveXObject("Microsoft.XMLHTTP"); |

To handle all modern browsers, including IE5 and IE6, check if the browser supports the XMLHttpRequest object. If it does, create an XMLHttpRequest object, if not, create an ActiveXObject:

|  |
| --- |
| var xmlhttp;  if (window.XMLHttpRequest)  {// code for IE7+, Firefox, Chrome, Opera, Safari  xmlhttp=new XMLHttpRequest();  }  else  {// code for IE6, IE5  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  } |

**AJAX - Send a Request To a Server**

To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

|  |
| --- |
| xmlhttp.open("GET","ajax\_info.txt",true);  xmlhttp.send(); |

|  |  |
| --- | --- |
| **Method** | **Description** |
| open(*method,url,async*) | Specifies the type of request, the URL, and if the request should be handled asynchronously or not.  *method*: the type of request: GET or POST *url*: the location of the file on the server *async*: true (asynchronous) or false (synchronous) |
| send(*string*) | Sends the request off to the server.  *string*: Only used for POST requests |

**GET or POST?**

GET is simpler and faster than POST, and can be used in most cases.

However, always use POST requests when:

* A cached file is not an option (update a file or database on the server)
* Sending a large amount of data to the server (POST has no size limitations)
* Sending user input (which can contain unknown characters), POST is more robust and secure than GET

**GET Requests**

A simple GET request:

|  |
| --- |
| xmlhttp.open(**"GET"**,"demo\_get.asp",true);  xmlhttp.send(); |

In the example above, you may get a cached result.

To avoid this, add a unique ID to the URL:

|  |
| --- |
| xmlhttp.open("GET","demo\_get.asp?t=" + **Math.random()**,true);  xmlhttp.send(); |

If you want to send information with the GET method, add the information to the URL:

|  |
| --- |
| xmlhttp.open("GET","demo\_get2.asp?**fname=Henry&lname=Ford"**,true);  xmlhttp.send(); |

**POST Requests**

A simple POST request:

|  |
| --- |
| xmlhttp.open(**"POST"**,"demo\_post.asp",true);  xmlhttp.send(); |

To POST data like an HTML form, add an HTTP header with setRequestHeader(). Specify the data you want to send in the send() method:

|  |
| --- |
| xmlhttp.open("POST","ajax\_test.asp",true);  **xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");**  xmlhttp.send("fname=Henry&lname=Ford"); |

|  |  |
| --- | --- |
| **Method** | **Description** |
| setRequestHeader(*header,value*) | Adds HTTP headers to the request.  *header*: specifies the header name *value*: specifies the header value |

**The url - A File On a Server**

The url parameter of the open() method, is an address to a file on a server:

|  |
| --- |
| xmlhttp.open("GET",**"ajax\_test.asp"**,true); |

The file can be any kind of file, like .txt and .xml, or server scripting files like .asp and .php (which can perform actions on the server before sending the response back).

**Asynchronous - True or False?**

AJAX stands for Asynchronous JavaScript and XML, and for the XMLHttpRequest object to behave as AJAX, the async parameter of the open() method has to be set to true:

|  |
| --- |
| xmlhttp.open("GET","ajax\_test.asp",**true**); |

Sending asynchronous requests is a huge improvement for web developers. Many of the tasks performed on the server are very time consuming. Before AJAX, this operation could cause the application to hang or stop.

With AJAX, the JavaScript does not have to wait for the server response, but can instead:

execute other scripts while waiting for server response deal with the response when the response ready

**Async=true**

When using async=true, specify a function to execute when the response is ready in the onreadystatechange event:

|  |
| --- |
| **xmlhttp.onreadystatechange=function()**  {  if (xmlhttp.readyState==4 && xmlhttp.status==200)  {  document.getElementById("myDiv").innerHTML=xmlhttp.responseText;  }  }  xmlhttp.open("GET","ajax\_info.txt",true);  xmlhttp.send(); |

**Async=false**

To use async=false, change the third parameter in the open() method to false:

|  |
| --- |
| xmlhttp.open("GET","ajax\_info.txt",false); |

Using async=false **is not recommended**, but for a few small requests this can be ok.

Remember that the JavaScript will NOT continue to execute, until the server response is ready. If the server is busy or slow, the application will hang or stop.

**Note:** When you use async=false, **do NOT write an onreadystatechange function** - just put the code after the send() statement:

Example

|  |
| --- |
| xmlhttp.open("GET","ajax\_info.txt",false);  xmlhttp.send();  document.getElementById("myDiv").innerHTML=xmlhttp.responseText; |

**AJAX - Server Response**

To get the response from a server, use the responseText or responseXML property of the XMLHttpRequest object.

|  |  |
| --- | --- |
| **Property** | **Description** |
| responseText | get the response data as a string |
| responseXML | get the response data as XML data |

**The responseText Property**

If the response from the server is not XML, use the responseText property.

The responseText property returns the response as a string, and you can use it accordingly:

|  |
| --- |
| document.getElementById("myDiv").innerHTML=xmlhttp.responseText; |

**The responseXML Property**

If the response from the server is XML, and you want to parse it as an XML object, use the responseXML property:

|  |
| --- |
| xmlDoc=xmlhttp.responseXML;  txt="";  x=xmlDoc.getElementsByTagName("ARTIST");  for (i=0;i<x.length;i++)  {  txt=txt + x[i].childNodes[0].nodeValue + "<br>";  }  document.getElementById("myDiv").innerHTML=txt; |

Full Code:

<http://www.w3schools.com/ajax/tryit.asp?filename=tryajax_responsexml>

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script>  function loadXMLDoc()  {  var xmlhttp;  var txt,x,i;  if (window.XMLHttpRequest)  {// code for IE7+, Firefox, Chrome, Opera, Safari  xmlhttp=new XMLHttpRequest();  }  else  {// code for IE6, IE5  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  }  xmlhttp.onreadystatechange=function()  {  if (xmlhttp.readyState==4 && xmlhttp.status==200)  {  xmlDoc=xmlhttp.responseXML;  txt="";  x=xmlDoc.getElementsByTagName("ARTIST");  for (i=0;i<x.length;i++)  {  txt=txt + x[i].childNodes[0].nodeValue + "<br>";  }  document.getElementById("myDiv").innerHTML=txt;  }  }  xmlhttp.open("GET","cd\_catalog.xml",true);  xmlhttp.send();  }  </script>  </head>  <body>  <h2>My CD Collection:</h2>  <div id="myDiv"></div>  <button type="button" onclick="loadXMLDoc()">Get my CD collection</button>    </body>  </html> |

**AJAX - The onreadystatechange Event**

When a request to a server is sent, we want to perform some actions based on the response.

The onreadystatechange event is triggered every time the readyState changes.

The readyState property holds the status of the XMLHttpRequest.

Three important properties of the XMLHttpRequest object:

|  |  |
| --- | --- |
| **Property** | **Description** |
| onreadystatechange | Stores a function (or the name of a function) to be called automatically each time the readyState property changes |
| readyState | Holds the status of the XMLHttpRequest. Changes from 0 to 4:  0: request not initialized  1: server connection established 2: request received  3: processing request  4: request finished and response is ready |
| status | 200: "OK" 404: Page not found |

In the onreadystatechange event, we specify what will happen when the server response is ready to be processed.

When readyState is 4 and status is 200, the response is ready:

<http://www.w3schools.com/ajax/tryit.asp?filename=tryajax_first>

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script>  function loadXMLDoc()  {  var xmlhttp;  if (window.XMLHttpRequest)  {// code for IE7+, Firefox, Chrome, Opera, Safari  xmlhttp=new XMLHttpRequest();  }  else  {// code for IE6, IE5  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  }  **xmlhttp.onreadystatechange=function()**  **{**  **if (xmlhttp.readyState==4 && xmlhttp.status==200)**  **{**  **document.getElementById("myDiv").innerHTML=xmlhttp.responseText;**  **}**  **}**  xmlhttp.open("GET","ajax\_info.txt",true);  xmlhttp.send();  }  </script>  </head>  <body>  <div id="myDiv"><h2>Let AJAX change this text</h2></div>  <button type="button" onclick="loadXMLDoc()">Change Content</button>  </body>  </html> |

**Note:** The onreadystatechange event is triggered five times (0-4), one time for each change in readyState.

**Using a Callback Function**

A callback function is a function passed as a parameter to another function.

If you have more than one AJAX task on your website, you should create ONE standard function for creating the XMLHttpRequest object, and call this for each AJAX task.

The function call should contain the URL and what to do on onreadystatechange (which is probably different for each call):

<http://www.w3schools.com/ajax/tryit.asp?filename=tryajax_callback>

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script>  var xmlhttp;  function loadXMLDoc(url,cfunc)  {  if (window.XMLHttpRequest)  {// code for IE7+, Firefox, Chrome, Opera, Safari  xmlhttp=new XMLHttpRequest();  }  else  {// code for IE6, IE5  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  }  xmlhttp.onreadystatechange=cfunc;  xmlhttp.open("GET",url,true);  xmlhttp.send();  }  **function myFunction()**  **{**  **loadXMLDoc("ajax\_info.txt",function()**  **{**  **if (xmlhttp.readyState==4 && xmlhttp.status==200)**  **{**  **document.getElementById("myDiv").innerHTML=xmlhttp.responseText;**  **}**  **});**  **}**  </script>  </head>  <body>  <div id="myDiv"><h2>Let AJAX change this text</h2></div>  <button type="button" onclick="myFunction()">Change Content</button>  </body>  </html> |

Practice #1

|  |
| --- |
| Modifies file cdinfo2.html by following conditions:  1. Write function showCDList by  2. Sent request to <http://www.108lab.com/getcdlist.php?q=req>  3. Make drop-down list have all cd title in http://www.108lab.com/cd\_catalog.xml |

Practice #2

|  |
| --- |
| Create webpage by following conditions:  1. Write any form you want and submit form using action="http://www.108lab.com/testinput.php"  2. See what happen to http://www.108lab.com/getinputlist.php  3. Write webpage by using AJAX to receive result from <http://www.108lab.com/getinputlist.php>  4. Make that webpage to (or alike) real time updates. |