



AJAX

Presenter:
Nnamani Nkiruka



What is Ajax?

Definition

AJAX – Asynchronous Javascript and XML

AJAX is not a programming language, but a new way to use existing standards.

It is the act of exchanging data with a server, and updating parts of a web page – without reloading

The XMLHttpRequest Object

In order to make a request to the server through Ajax, an object must be created that can be used for different forms of functionality. We have to note that XMLHttpRequest object is both instantiated and handled differently by different browser.

For instance, Internet Explorer creates it as activeXObject while browsers such as Firefox and Safari creates it as XMLHttpRequest.

The XMLHttpRequest Methods

Once an instance of the XMLHttpRequest object has been created, there are a number of methods available to the user. Depending on how you want to use the object, different methods may become more important than others.

Let's take a closer look on some of these methods available for use.

The XMLHttpRequest Methods

abort()

The abort method is really quite simple—it stops the request in its tracks. This function can be handy if you are concerned about the length of the connection. If you only want a request to fire for a certain length of time, you can call the abort method to stop the request prematurely.

The XML HttpRequest Methods

`getResponseHeader("headername")`

You can use this method to obtain the content of a particular piece of the header. This method can be useful to retrieve one part of the generally large string obtained from a set of headers. For example, to retrieve the size of the document requested, you could simply call `getResponseHeader ("Content-Length")`.

The XML HttpRequest Methods

`getAllResponseHeaders()`

You can use this method to obtain the full information on all HTTP headers that are being passed.

The XML HttpRequest Methods

```
open ("method","URL","async","username","pswd")
```

This is the method you use to open a connection to a particular file on the server. It is where you pass in the method to open a file (GET or POST), as well as define how the file is to be opened. Keep in mind that not all of the arguments in this function are required and can be customized depending on the situation.

The XML HttpRequest Methods

`setRequestHeader("label","value")`

With this method, you can give a header a label of sorts by passing in a string representing both the label and the value of said label. An important note is that this method may only be invoked after the `open()` method has been used, and must be used before the `send` function is called.

The XML HttpRequest Methods

`send("content")`

This is the method that actually sends the request to the server. If the request was sent asynchronously, the response will come back immediately; if not, it will come back after the response is received. You can optionally specify an input string as an argument, which is helpful for processing forms, as it allows you to pass the values of form elements.

The XMLHttpRequest Properties

onreadystatechange

The onreadystatechange property is an event handler that allows you to trigger certain blocks of code, or functions, when the state (referring to exactly where the process is at any given time) changes.

For example, if you have a function that handles some form of initialization, you could get the main set of functionality you want to fire as soon as the state changes to the complete state.

The XMLHttpRequest Properties

readyState

The readyState property gives you an in-depth description of the part of the process that the current request is at.

This is a highly useful property for exception handling, and can be important when deciding when to perform certain actions.

You can use this property to create individual actions based upon how far along the request is. For example, you could have a set of code execute when readyState is loading, or stop executing when readyState is complete.

The XMLHttpRequest Properties

responseText

The responseText property will be returned once a request has gone through. If you are firing a request to a script of some sort, the output of the script will be returned through this property.

With that in mind, most scripts will make use of this property by dumping it into an innerHTML property of an element, thereby asynchronously loading a script or document into a page element.

The XML HttpRequest Properties

`responseXML`

This works similarly to `responseText`, but is ideal if you know for a fact that the response will be returned in XML format.

The XML HttpRequest Properties

Status

This property dictates the response code that was returned from the request. For instance, if the file requested could not be found, the status will be set to 404 because the file could not be found.

Sending a Requests to the Server

The first thing you have to do is instantiate the XMLHttpRequest object.

Example

```
If(window.XMLHttpRequest)
    xmlhttp = new XMLHttpRequest;
else if(window.ActiveXObject);
    xmlhttp = new ActiveObject;
else
    alert("your browser is obsolete");
```

Requests with Get Method

```
function sendData()  
{  
    if(xmlHttp.readyState==4 || xmlHttp.readyState==0)  
    {  
        xmlHttp.open("GET","page.pgp",true);  
        xmlHttp.onreadystatechange = handleSendData;  
        xmlHttp.send(null);  
    }  
}
```

Requests with Get Method

```
function handleSendData()
{
    if(xmlHttp.readyState==4)
    {
        xmlDoc = xmlHttp.responseXML;
    }
}
```

Requests with Post Method

```
function sendData()  
{  
    if(xmlHttp.readyState==4 || xmlHttp.readyState==0)  
    {  
        // create pairs index=value with data that must be sent to server  
        var data = 'test='+document.getElementById('txt2').innerHTML;  
        xmlHttp.open("GET","page.php",true);  
        xmlHttp.onreadystatechange = handleSendData;  
        xmlHttp.setRequestHeader("Content-type", "application/x-www-  
form-urlencoded");  
        xmlHttp.send(data);  
    }  
}
```

Setting of Headers

```
header("Expires: Tue, 11 Aug 1998 04:00:00 GMT");
```

```
header("Last_Modified: " . gmdate("D, d M Y, H:i:s") . "GMT");
```

```
header("Cache-control: no-cache, must-revalidate");
```

```
header("Content-Type: text/xml; charset=utf-8");
```

```
header("Pragma: no-cache");
```

Uploading images to the server with Ajax

It is not possible to process a file upload through the XMLHttpRequest object, because JavaScript has no access to your computer's file system. But, there are still ways to perform Ajax-like functionality for this task without making use of the XMLHttpRequest object.

This can be done via a hidden <iframe>, using the following method, which has four main steps:

Uploading images to the server with Ajax

❖ Create a form (with a file element) and a hidden iframe.

The iframe must have a name (name attribute).

The <form> tag must have a "target" attribute whose value is the name of <iframe> and an onsubmit() event (with return false;) that sends form data to a JavaScript function for the page is not refreshed when you press the Submit button.

Uploading images to the server with Ajax

- ❖ The JavaScript function uses the submit() event to send form data to the iframe (where the target attribute indicates). The iframe transmits these data to a PHP script whose address is added in its "src" attribute.
- ❖ The PHP script uploads the image file on the server.

Uploading images to the server with Ajax

- ❖ The JavaScript function uses the submit() event to send form data to the iframe (where the target attribute indicates). The iframe transmits these data to a PHP script whose address is added in its "src" attribute.
- ❖ The PHP script uploads the image file on the server.



Questions