

# XMLHttpRequest Object<sup>1</sup>

Jonathan Geisler

April 14, 2008



---

<sup>1</sup>Material heavily borrowed from Apple

Microsoft had the foresight to create a way for a web page to contact a server without having to refresh the page. Initially, it was an ActiveX object. Other browser developers (Mozilla, Safari, Opera, Konqueror, etc.) noticed and copied with a cross platform solution. And all the world rejoiced!

# Creating the Object

For native support:

```
var reqObj = new XMLHttpRequest();
```

For ActiveX support:

```
var reqObj  
    = new ActiveXObject("Microsoft.XMLHTTP");
```



# Creating the Object portably

Use a portability framework (e.g., Sarissa), or:

```
var reqObj;
```

```
if (window.XMLHttpRequest) {
    reqObj = new XMLHttpRequest();
} else if (window.ActiveXObject) {
    try {
        reqObj
            = new ActiveXObject("Msxml2.XMLHTTP")
    } catch (err) {
        reqObj
            = new ActiveXObject("Microsoft.XMLHTT
    }
}
```

The object contains the following methods:

- abort()
- getAllResponseHeaders()
- getResponseHeader()
- open()
- send()
- setRequestHeader()

The object contains the following properties:

- onreadystatechange
- **readyState**
- responseText
- responseXML
- status
- statusText

This function takes up to 5 parameters:

- 1 method
- 2 URL
- 3 asyncFlag = true
- 4 userName
- 5 password

The state can have the following values:

- 0 = uninitialized
- 1 = loading
- 2 = loaded
- 3 = interactive
- 4 = complete

To keep people from writing effective phishing attacks where they have a web site that borrows the look from one web site, but sends submitted data to a malicious site, most browsers will limit the object from contacting any site other than the site the containing page came from. This severely limits how you can tie a page together with services from a bunch of various sources. Oh well . . .