

<script>



Advanced JavaScript

AJAX and XMLHttpRequest

What is AJAX?



- ❑ **AJAX** stands for **A**synchronous **J**avaScript **A**nd **X**ML.
- ❑ AJAX is a technique for creating “better, faster, more responsive web applications”.
- ❑ 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.

What is AJAX? (Cont.)



- ❑ It is Not a stand-alone language or technology, but a new way to use existing standards.
- ❑ AJAX applications are browser- and platform-independent.
- ❑ It is a client side technology.
- ❑ The first use of the term in public was by Jesse James Garrett in February 2005.
- ❑ Web applications with Ajax are supposed to replace all our traditional desktop applications.
- ❑ These changes are so sweeping that the Ajax-enabled web is sometimes know as “Web 2.0”.

How AJAX Works?



□ **AJAX is based on internet standards, and uses a combination of:**

- XMLHttpRequest object (to exchange data asynchronously with a server)
- JavaScript/DOM (to display/interact with the information)
- CSS (to style the data)
- XML / JSON (often used as the format for transferring data)

AJAX Frameworks and Implementation



❑ Mainly, it is implemented in Javascript code.

❑ There are application ‘frameworks’ that support Ajax within a server/application building environment. For example:

- ASP.NET AJAX (previously Microsoft Atlas)
- JSF (Java).
- Sajax (PHP).

The XMLHttpRequest object



- ❑ The XMLHttpRequest object is used to exchange data with a server.
- ❑ It is an object (a constructor function) that allows you to send HTTP requests from JavaScript.
- ❑ supported by virtually all modern browsers, including IE 5+ , Firefox and Opera, and it is supported on a wide range of platforms, including Microsoft Windows, UNIX/Linux, and Mac OS X.
- ❑ Used in Ajax, and to get files and data from the server.

The XMLHttpRequest object(Cont.)



❑ Some of the object's properties and methods:

○ Properties:

| Property | Description |
|---------------------------|---|
| readyState | Integer reporting the status of the request |
| onreadystatechange | Determines which event handler will be called when the object's readyState property changes |
| responseXML | Data returned by the server expressed as a document object |
| responseText | Data returned by the server in text string form |
| status | Returns the status as a number (e.g. 404 for "Not Found" or 200 for "OK") |

The XMLHttpRequest object(Cont.)



□ Ready State Values:

| 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 |

The XMLHttpRequest object(Cont.)



□ HTTP error types (Status):

- 1xx Informational
- 2xx Success
- 3xx Redirection
- 4xx Client Error
- 5xx Server Error

□ Status & Status Text:

404: Not found

The server found nothing matching the URI given.

- 200: OK

The server successfully returned the page

- 400: Bad Request

Server didn't understand the request due to malformed syntax.

- 401: Unauthorized

The request requires user authentication.

- 500: Internal Server Error

The server encountered an unexpected error and couldn't fulfill the request.

- 503: Service Unavailable

The server is currently unable to handle the request due to temporary overloading or maintenance.

The XMLHttpRequest object(Cont.)



○Methods:

| Method | Description |
|---------------------------------|--|
| abort() | Stops the current request. |
| open('method','URL','a') | 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() | Sends the request, optionally take post data as a parameter. |

The XMLHttpRequest object(Cont.)



□ Steps for communication with the server using XMLHttpRequest:

- Creating XMLHttpRequest (XHR) object.
- Initialize the object (using open() method).
- Send the request (Using send() method).
- Monitoring the state of the request.
- Dealing with the server response.

Create XMLHttpRequest object



❑ Create a XMLHttpRequest object:

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

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

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

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



❑ Sending the Server Request (get):

```
xmlhttp.open("get","json_info.txt",true);  
xmlhttp.send();
```

❑ Sending the Server Request (difference between get and post):

```
xmlhttp.open("get","/checkEmail.aspx?Email=abc@test.com",true);  
xmlhttp.send();
```

```
xmlhttp.open("post","/checkEmail.aspx ",true);  
xmlhttp.send("abc@test.com");
```

Monitoring Request Status



□Monitoring RequestStatus:

- 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.

| 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 |

Monitoring Request Status (Cont.)



❑ Monitoring Request Status & Dealing with the Response:

```
xmlhttp.onreadystatechange=function()
{
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        var str=xmlhttp.responseText;
    }
}

xmlhttp.open("GET","json_info.txt",true);
xmlhttp.send();
```

Dealing with the server response



- ❑ Ajax allows for this information to be returned in a number of formats, including ASCII text and XML data.
- ❑ We can process the returned information with the help of two XMLHttpRequest object's properties:
 - responseText.
 - responseXML.

Dealing with the server response(Cont..)



❑ The responseText Property:

- Readonly Property.
- Returned Text can be manipulated using any of JavaScript's methods relating to strings(charAt(), indexOf(), substring()....).
- We can use it like that:

```
var myText = xmlhttp.responseText;
```

Dealing with the server response(Cont..)



□ The responseXML Property:

- Readonly Property.
- Returned XML document can be manipulated using JavaScript's DOM methods and properties.

Dealing with the server response(Cont..)



- ❑ JavaScript's DOM methods and properties.

| | |
|---------------------------------------|--|
| getElementsByTagName("Family") | Returns an array of all the elements having the "Family" tagname |
| childNodes[x] | Returns the node number (x) of a parent node in the document |
| nodeValue | Return the value of the current node |

```
var nodeArray = xmlhttp.responseXML.getElementsByTagName("Family");  
  
var famNode = nodeArray[0];  
  
var famText = famNode.childNodes[0].nodeValue;  
  
alert("Family Name: " + famText);
```

Putting it all together



❑ Complete code:

```
var xmlhttp;  
    xmlhttp=new XMLHttpRequest();  
xmlhttp.onreadystatechange=function()  
{  
    if (xmlhttp.readyState==4 && xmlhttp.status==200)  
    {  
        var str=xmlhttp.responseText;  
    }  
}  
xmlhttp.open("GET","json_info.txt",true);  
xmlhttp.send();
```

References ...



❑ For further information :

- Teach Yourself AJAX in 10 Minutes.

- A press - Beginning XML with DOM and Ajax - From Novice to Professional.

- Manning.Ajax.In.Action.Oct.2005.HmG.

Self Study...



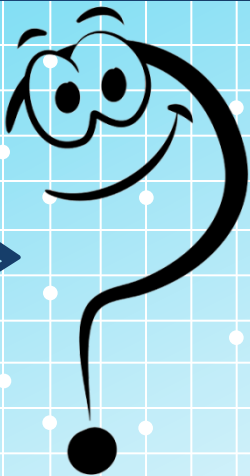
- ☐ HTTP protocol overview.
- ☐ HTTP headers.
- ☐ HTTP response codes.
- ☐ HTTP get & Post methods.

<script>



JavaScript

</script>

<SCRIPT>  </SCRIPT>

```
<script>document.writeln("Thank  
You!")</script>
```