

## Your First AJAX Application

To understand how AJAX works, we will create a small AJAX application.

First, we are going to create a standard HTML form with two text fields: username and time. The username field will be filled in by the user and the time field will be filled in using AJAX.

# Your First AJAX Application

## Your First AJAX Application

The HTML file will be named "testAjax.htm", and it looks like this (notice that the HTML form below has no submit button!):

```
<html>
<body>

<form name="myForm">
Name: <input type="text" name="username" />
Time: <input type="text" name="time" />
</form>

</body>
</html>
```

## AJAX - Browser Support

The keystone of AJAX is the XMLHttpRequest object.

Different browsers use different methods to create the XMLHttpRequest object.

Internet Explorer uses an **ActiveXObject**, while other browsers use the built-in JavaScript object called **XMLHttpRequest**.

To create this object, and deal with different browsers, we are going to use a "try and catch" statement. You can read more about the try and catch statement in our JavaScript tutorial.

## AJAX - Browser Support

Let's update our "testAjax.htm" file with the JavaScript that creates the XMLHttpRequest object:

```
<html>
<body>

<script type="text/javascript">
function ajaxFunction()
{
var xmlHttp;
try
{
// Firefox, Opera 8.0+, Safari
xmlHttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
try
{
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e)
{
alert("Your browser does not support AJAX!");
return false;
}
}
}
}
}
</script>

<form name="myForm">
Name: <input type="text" name="username" />
Time: <input type="text" name="time" />
</form>

</body>
</html>
```

## AJAX - Browser Support

Example explained: First create a variable `xmlHttp` to hold the `XMLHttpRequest` object.

Then try to create the object with `XMLHttp=new XMLHttpRequest()`. This is for the Firefox, Opera, and Safari browsers. If that fails, try `xmlHttp=new ActiveXObject("Msxml2.XMLHTTP")` which is for Internet Explorer 6.0+, if that also fails, try `xmlHttp=new ActiveXObject("Microsoft.XMLHTTP")` which is for Internet Explorer 5.5+

If none of the three methods work, the user has a very outdated browser, and he or she will get an alert stating that the browser doesn't support AJAX.

## AJAX - Browser Support

Note: The browser-specific code above is long and quite complex. However, this is the code you can use every time you need to create an XMLHttpRequest object, so you can just copy and paste it whenever you need it. The code above is compatible with all the popular browsers: Internet Explorer, Opera, Firefox, and Safari.

The next chapter shows how to use the XMLHttpRequest object to communicate with the server.

## AJAX - The XMLHttpRequest Object

Before sending data to the server, we have to explain three important properties of the XMLHttpRequest object.

### The onreadystatechange Property

After a request to the server, we need a function that can receive the data that is returned by the server.

The onreadystatechange property stores your function that will process the response from a server. This is not a method, the function is stored in the property to be called automatically. The following code sets the onreadystatechange property and stores an empty function inside it:

```
xmlHttp.onreadystatechange=function()  
{  
// We are going to write some code here  
}
```

## AJAX - The XMLHttpRequest Object

### The readyState Property

The readyState property holds the status of the server's response. Each time the readyState changes, the onreadystatechange function will be executed.

Here are the possible values for the readyState property:

State	Description
0	The request is not initialized
1	The request has been set up
2	The request has been sent
3	The request is in process
4	The request is complete

## AJAX - The XMLHttpRequest Object

### The readyState Property

We are going to add an If statement to the onreadystatechange function to test if our response is complete (this means that we can get our data):

```
xmlHttp.onreadystatechange=function()  
{  
if(xmlHttp.readyState==4)  
  {  
    // Get the data from the server's response  
  }  
}
```

## AJAX - The XMLHttpRequest Object

### The responseText Property

The data sent back from the server can be retrieved with the responseText property.

In our code, we will set the value of our "time" input field equal to responseText:

```
xmlHttp.onreadystatechange=function()  
{  
  if(xmlHttp.readyState==4)  
  {  
    document.myForm.time.value=xmlHttp.responseText;  
  }  
}
```

Now we can look at how to ask the server for some data!

## AJAX - Request a Server

### **Sending a Request to the Server**

To send off a request to the server, we use the `open()` method and the `send()` method.

The `open()` method takes three arguments. The first argument defines which method to use when sending the request (GET or POST). The second argument specifies the URL of the server-side script. The third argument specifies that the request should be handled asynchronously. The `send()` method sends the request off to the server. If we assume that the HTML and PHP file are in the same directory, the code would be:

```
xmlHttp.open("GET", "time.php", true); xmlHttp.send(null);
```

## AJAX - Request a Server

### Sending a Request to the Server

Now we must decide when the AJAX function should be executed. We will let the function run "behind the scenes" when the user types something in the username text field:

```
<form name="myForm">  
Name: <input type="text"  
onkeyup="ajaxFunction();" name="username" />  
Time: <input type="text" name="time" />  
</form>
```

# Your First AJAX Application

## AJAX - Request a Server

Our updated AJAX-ready "testAjax.htm" file now looks like this:

```
<html>
<body>

<script type="text/javascript">
function ajaxFunction()
{
var xmlhttp;
try
{
// Firefox, Opera 8.0+, Safari
xmlhttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
try
{
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e)
{
alert("Your browser does not support AJAX!");
return false;
}
}
}
xmlhttp.onreadystatechange=function()
{
if(xmlhttp.readyState==4)
{
document.myForm.time.value=xmlhttp.responseText;
}
}
xmlhttp.open("GET","time.asp",true);
xmlhttp.send(null);
}
</script>

<form name="myForm">
Name: <input type="text"
onkeyup="ajaxFunction();" name="username" />
Time: <input type="text" name="time" />
</form>

</body>
</html>
```

## AJAX - Request a Server

All that's left to do is to make sure time.php returns some data.

..and the TEST !