

[Back To List of Tutorials](#)

# A Simple AJAX Tutorial

## What is AJAX?

AJAX stands for Asynchronous Java and XML. It is mainly a mix of several technologies such as HTML, CSS, JavaScript, The Document Object Model and the XMLHttpRequest object (an object that lets you make requests to a server). AJAX lets you make web based user interfaces more responsive by letting you update sections of the page without refreshing the entire page.

## What do I need to know to understand this article?

This article helps you get started with AJAX. You are expected to know some web related technologies and terms such as

- Basic HTML Concepts (GET requests/URL parameters)
- Basic Javascript Concepts
- The <div> tag

## What is the sample example about?

We have 2 pages - index.jsp & details.jsp

index.jsp: This one just lists certain terms on the page and the user can click on a link (there's a link for each term) to fetch the definition of that term.

definition.jsp: This one fetches the definition for different words from a database. This page takes in the word id as a URL parameter and does the necessary lookups to get the definition of the word. I shall not go into the source code for this page. Just consider it to be a simple server side page that returns dynamic content based on URL parameters. (This page can be a simple static HTML page as well. But it makes it easier to see the benefits of a AJAX (and I mean callbacks to the server to get more information) when the content is something that is not static and cannot be loaded with the index.jsp itself.

## So is the AJAX code complex? Whats does the source code of the index.jsp look like?

Source code for index.jsp

```
<html>
<body>

<script language="Javascript" type="text/javascript">
<!--

function createRequestObject() {
    var tmpXmlHttpRequest;

    //depending on what the browser supports, use the right way to create the XMLHttpRequest object
    if (window.XMLHttpRequest) {
        // Mozilla, Safari would use this method ...
        tmpXmlHttpRequest = new XMLHttpRequest();

    } else if (window.ActiveXObject) {
        // IE would use this method ...
        tmpXmlHttpRequest = new ActiveXObject("Microsoft.XMLHTTP");
    }

    return tmpXmlHttpRequest;
}

//call the above function to create the XMLHttpRequest object
var http = createRequestObject();

function makeGetRequest(wordId) {
    //make a connection to the server ... specifying that you intend to make a GET request
    //to the server. Specify the page name and the URL parameters to send
    http.open('get', 'definition.jsp?id=' + wordId);

    //assign a handler for the response
    http.onreadystatechange = processResponse;
```

```

    //actually send the request to the server
    http.send(null);
}

function processResponse() {
    //check if the response has been received from the server
    if(http.readyState == 4){

        //read and assign the response from the server
        var response = http.responseText;

        //do additional parsing of the response, if needed

        //in this case simply assign the response to the contents of the <div> on the page.
        document.getElementById('description').innerHTML = response;

        //If the server returned an error message like a 404 error, that message would be shown within the div tag!
        //So it may be worth doing some basic error before setting the contents of the <div>
    }
}

-->
</script>

<h1>Have you heard these terms before?</h1>
<p>
Ceraunophobia <a href="javascript:makeGetRequest(1)">More about Ceraunophobia</a><br>
Astraphobia <a href="javascript:makeGetRequest(2)">More about Astraphobia</a><br>
Ophidiophobia <a href="javascript:makeGetRequest(3)">More about Ophidiophobia</a><br>
</p>

<div id="description"></div>

</body>
</html>

```

The code should be pretty simple to understand. The part within `<script language="Javascript">` block is the main Javascript part that does all the AJAX calls to the server. I shall explain these later. The part after the `</script>` tag (the ending part of the code) contains simple html links for each of the different words (different types of fears that ppl suffer from.. couldn't come up with something better :-). These links basically call one of the Javascript functions named `makeGetRequest()` and pass in the id for the word. This id is then passed on to `details.jsp` (used by that page to fetch the correct data). There is also a `<div>` tag that is empty right now. If you do not know what the `<div>` tag is - you can think of it as a simple placeholder within which you can have stuff like text. Just remember that this placeholder has been assigned an id ('description' in this case) and you can refer to this placeholder by that id anytime you need to.

## Completed 12th?

Get an IT career along with college Join Aptech Computer Education



### What do the Javascript functions actually do?

There are 3 Javascript functions listed and one extra line outside the function blocks.

The `var http = createRequestObject()` uses the `createRequestObject()` function to create the XMLHttpRequest object (depending on whether the user is using Internet Explorer or some other browser). This object is used to make all the AJAX requests to other pages on the server.

The second function `makeGetRequest()` is used to make the actual request to the server. The first line invokes the `open()` method on the XMLHttpRequest object. This method lets you specify the page to call on the server and also lets you specify that you want to use GET for the request. You can append parameters to the URL (the normal GET style) if you want to pass in some values to the page you are going to call. The other thing worth noting is that 'processResponse' is the name of the function that must be called when the response is received from the server. You can name this anything you want, but use the same name for the third function as well.

The third function `processResponse()` basically checks whether the response from the server has been received (the response is basically the text of the page returned) and assigns that text to a local variable called 'response'. After that the text within the `<div>` tag on the page (this text can be accessed using the property called `innerHTML`) is replaced with the text that was received as

a response.

### How can I improve this code? And make it more generic?

There are lots of things that could be changed here (and should be in production code).

- The filename 'definition.jsp' in the `makeGetRequest()` should not be hardcoded. That should be configurable so that you can use the same function to make AJAX calls to different server side pages. Maybe the filename could be sent to the function as one of its parameters.
- The parameter 'wordId' to `makeGetRequest()` must actually be name/value pairs instead of just a single value as in this case. The function should then loop through all the name/value pairs and append them to the URL before making the GET request. This lets you pass in multiple parameters to the server side pages when you make the AJAX calls.
- The id for the `<div>` tag ('description' in this case) should not be hardcoded within the `processResponse()` method. That too should be configurable so that the response can be displayed within different elements on your main page. This too could be made a parameter to the function `processResponse()`. You could also explore the `visibility` property of `<div>` tags to show and hide `<div>` from within this function for more interesting options.

I hope this article has helped you get started with AJAX. To get more information, visit the Mozilla site (<http://developer.mozilla.org/en/docs/AJAX>) for excellent articles on AJAX

*Written by Kiran Pai*

*Please send your feedback or comments to [feedback \[at\] codecoffee \[dot\] com](mailto:feedback[at]codecoffee[dot]com)*

**[Back To List of Tutorials](#)**