
Ajax- XMLHttpRequest

XMLHttpRequest - A Read only field

- ✱ Returns a value such as [ArrayBuffer](#), [Blob](#), [Document](#), [JavaScript object](#), or a [DOMString](#), based on the value of [XMLHttpRequest.responseType](#). This property holds the entire response value. In case of invalid request this property returns null.
- ✱ To get the response from a server, use the [responseText](#) or [responseXML](#) property of the [XMLHttpRequest object](#).

XMLHttpRequest Object:

- ✱ XMLHttpRequest is an [API to deliver](#) certain range of operations to transfer data between a [client](#) and a [server](#) behind the scenes.
- ✱ It acts as a source to do [partial page updates](#) without the need to perform full page refresh.
- ✱ This eases the task of handling data inputs and outputs in a web page without breaking user's other tasks.
- ✱ It is used intensely in [AJAX](#) programming and is supported by all present date browsers such as [IE7](#), [Chrome](#), [Firefox](#), [Safari](#) and [Opera](#).
- ✱ This is one of the frequent learning ajax tutorial and provide lots of ajax basics. This will be best entry with ajax for beginners.



Sending an AJAX Request to a Server

✿ Requests are made to web server with the `open()` and `send()` methods.

1. The `open()` method has three parameters:

✿ **First Parameter Value: GET or POST**

⌘ This is the type of [request sent to server](#).

⌘ [GET is simpler](#) request but has size limitations and security issues.

⌘ [POST is faster](#), can accept larger quantity of data and secure in a way [than GET](#).

✿ **Second Parameter Value: The location of the server file.**

⌘ The server file [URL needs](#) to be called with a query string parameter along with it value.

⌘ This value is taken from the variable "[str](#)", which is a parameter for the function [getData\(\)](#).

✿ **Third parameter value:**

⌘ `true` – indicates that it is an asynchronous call to the server.

If the value is `false` then it indicates a synchronous call. By default, the value is "`true`".

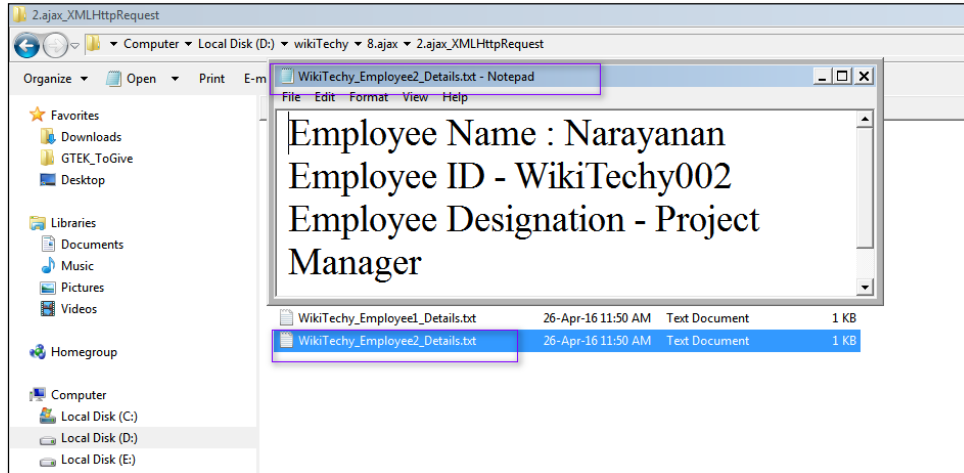
2. `send()` – sends request to the server URL.

Sample Code:

```
<html>
  <head>
    <script>
      function loadDoc(url )
      {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function()
        {
          if (xhttp.readyState == 4 && xhttp.status
          == 200)
          {

            document.getElementById("text").innerHTML =
            xhttp.responseText;
          } };
        xhttp.open("GET", url, true);
        xhttp.send();
      }
    </script>
  </head>
  <body>
    <div id="text"><h2>WikiTechy - Welcome you to learn
Ajax</h2></div>
    <button type="button"
onclick="loadDoc('WikiTechy_Employee1_Details.txt')">Employee1
Details</button>
    <button type="button"
onclick="loadDoc('WikiTechy_Employee2_Details.txt')">Employee2
Details</button>
  </body>
</html>
```

✳ The file content of - WikiTechy_Employee2_Details.txt



Code Explanation:

```
<html>
<head>
<script>
    function loadDoc( url )
    {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function()
        {
            if (xhttp.readyState == 4 && xhttp.status == 200)
            {
                document.getElementById("text").innerHTML =xhttp.responseText;
            }
        };
        xhttp.open("GET", url, true);
        xhttp.send();
    }
</script>
</head>
<body>
    <div id="text"><h2>WikiTechy - Welcome you to learn Ajax</h2></div>

    <button type="button" onclick="loadDoc('WikiTechy_Employee1_Details.txt')">
    Employee1 Details</button>
    <button type="button" onclick="loadDoc('WikiTechy_Employee2_Details.txt')">
    Employee1 Details</button>

</body>
</html>
```

- 1 <script> HTML element to contain [ajax code](#).
- 2 Create a function with name "loadDoc()" containing the [url](#) of the file as input.
- 3 Declare and [initialize the XMLHttpRequest object](#) for browser other than IE5 and IE6 as *xhttp = new XMLHttpRequest();*.
- 4 Capture the [state change](#) of the created object. This occurs when the [server receives](#) a request and [sends its response](#).
- 5 Check for the [readyState](#) and status property values. It should be [equal to 4 and 200 respectively](#). This means that response from the server has been received completely and assign the response text to a [div element – text's innerHTML](#) attribute.
- 6 Call the [open\(\)](#) to send [HTTP](#) request to the server with 'GET' message type, text filename [url](#) as input, along with explicitly setting asynchronous [property to true](#). The "true" value indicates that this data transfer to and from [server without disturbing any parallel task](#) that is happening in the HTML code.
- 7 Finally call the [send\(\)](#) to send the request.
- 8 End the <script> element to indicate the end of [ajax code](#).
- 9 Define a <div> element with name "[text](#)" and type a header text.

- 10 Add a button and in `onclick()` event call the ajax function `loadDoc()` and pass the file name "`WikiTechy_Employee1_Details.txt`" as parameter for the function.
- 11 Add another button and in `onclick()` event call the ajax function `loadDoc()` and pass the file name "`WikiTechy_Employee2_Details.txt`" as parameter for the function.

Output:



- 1 The header text inside the `<div>` element is displayed.
- 2 Click the `Employee1 Details` button to call ajax function with a text file as input.
- 3 Click the `Employee2 Details` button to call ajax function with another text file as input.



- 1 As you can see the full page refresh has not happened here after the button click. Instead the partial page, that is the `<div>` element alone is updated and it means that things **change without any effect** on other elements of the page due to the implementation of ajax.
- 2 Employee1 Details Button is clicked.
- 3 Now the `<div>` element displays the text **file content** due to ajax request and response processes.



- 1 As you can see the [full page refresh](#) has not happened here after the button click. Instead the partial page, that is the `<div>` element alone is [updated](#) and it means that things change without any effect on other elements of the page [due to the implementation of ajax](#).
- 2 [Employee2 Details Button](#) is clicked.
- 3 Now the `<div>` element displays the [text file content](#) due to ajax request and response processes.