

Ajax - XMLHttpRequest Object

The XMLHttpRequest object

- ✱ All present day browsers extend their support for [XMLHttpRequest object](#) as it helps to transfer [data to and forth to server](#), in a background process. Its main advantage is the possibility to update parts of a web page.

Create an XMLHttpRequest object:

- ✱ Browsers such as [Chrome](#), [IE7+](#), [Firefox](#), [Safari](#), and [Opera](#) have a in-built support for [XMLHttpRequest object](#) and hence the syntax will be as:

```
variable = new XMLHttpRequest();
```

- ✱ Browsers [IE5](#) and [IE6](#) use its own [ActiveX Object](#):

```
variable = new ActiveXObject("Microsoft.XMLHTTP");
```

- ✱ [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.



- ✱ [XMLHttpRequest](#) is used profoundly 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.

XMLHttpRequest.onreadystatechange

- ✱ An [EventHandler](#) that preserves a function to be triggered each time when the [readyState property](#) value changes.

XMLHttpRequest.readyState - A Read only field

- ✱ This property contains the status of the server's response. Below are the list of possible values:
 - ✱ 0: The request object is just been declared.
 - ✱ 1: The request has been initialized.
 - ✱ 2: The request has been sent and its status is available.
 - ✱ 3: The request is in process of receiving a response from the server.
 - ✱ 4: The response is received completely.

XMLHttpRequest.response - A Read only field

- ✱ Returns a value such as [ArrayBuffer](#), [Blob](#), [Document](#), [JavaScript object](#), or a [DOMString](#), based on the value of



`XMLHttpRequest.responseText`. This property holds the entire response `value`. In case of invalid request this property returns null.

XMLHttpRequest.responseText - A Read only field

- ✱ The response `received from the server` in text format.

XMLHttpRequest.status - A Read only field

- ✱ Returns an `unsigned short containing the status` of the response.

Sample Code:

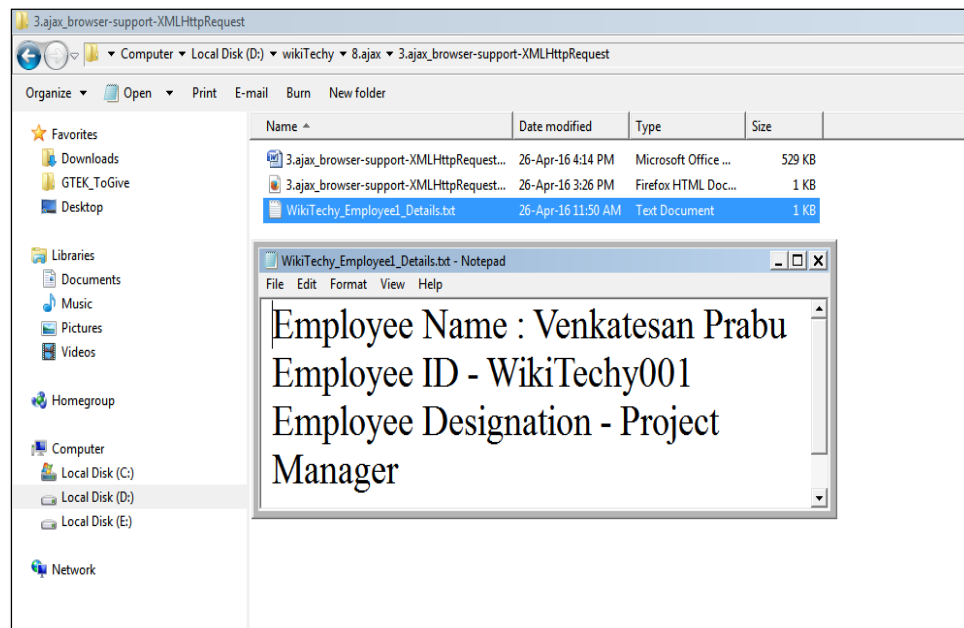
```
<html>
  <head>
    <script>
      function loadDoc()
      {
        var xhttp;
        if (window.XMLHttpRequest)
        {
          xhttp = new XMLHttpRequest();    // code for IE7+,
          Firefox, Chrome, Opera, Safari }
        else {
          xhttp = new ActiveXObject("Microsoft.XMLHTTP");// code for IE6, IE5 }
          xhttp.onreadystatechange = function()
          {
            if (xhttp.readyState == 4 && xhttp.status == 200)
            {
              document.getElementById("text").innerHTML =
                xhttp.responseText;
            } };
          xhttp.open("GET", "WikiTechy_Employee1_Details.txt", true);
          xhttp.send();
        }
      </script>
```



```
</head>
<body>
  <div id="text"><h2>WikiTechy - Welcomes you to learn
  Ajax</h2></div>
  <button type="button" onclick="loadDoc()"> Click to Load
  Data from File</button>

</body>
</html>
```

✳ The file content of - WikiTechy_Employee1_Details.txt



Code Explanation:

```
<html>
<head>
  <script>
    function loadDoc()
    {
      var xhttp;
      if (window.XMLHttpRequest)
      {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xhttp = new XMLHttpRequest();
      }
      else
      {
        // code for IE6, IE5
        xhttp = new ActiveXObject("Microsoft.XMLHTTP");
      }
      xhttp.onreadystatechange = function()
      {
        if (xhttp.readyState == 4 && xhttp.status == 200)
        {
          document.getElementById("text").innerHTML = xhttp.responseText;
        }
      };
      xhttp.open("GET", "WikiTechy_Employee1_Details.txt", true);
      xhttp.send();
    }
  </script>
</head>
<body>
  <div id="text"><h2>WikiTechy - Welcomes you to learn Ajax</h2></div>
  <button type="button" onclick="loadDoc()"> Click to Load Data from File</button>
</body>
</html>
```

1 <script> HTML element to contain ajax code.

2 Create a function with name "loadDoc()".

3 Declare a variable xhttp.

4 Check for the browser type.

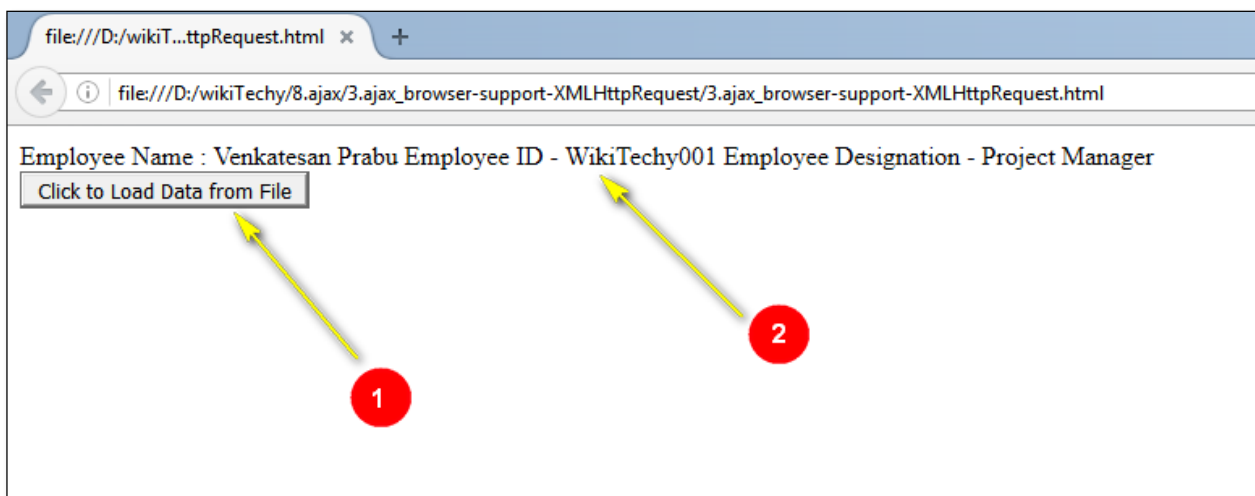
- 5 Initialize the `XMLHttpRequest` object for browser other than IE5 and IE6 as `xhttp = new XMLHttpRequest();`.
- 6 Initialize the `XMLHttpRequest` object for browser IE5 or IE6 as `xhttp = new ActiveXObject("Microsoft.XMLHTTP");`
- 7 Capture the state change of the created object. This occurs when the server receives a request and sends its response.
- 8 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.
- 9 Assign the response text to a div element – `text's innerHTML` attribute.
- 10 Call the `open()` to send HTTP request to the server with 'GET' message type, text filename 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.
- 11 Finally call the `send()` to send the request.
- 12 End the `<script>` element to indicate the end of ajax code.
- 13 Define a `<div>` element with name "text" and type a header text and add a button and in `onclick()` event call the ajax function `loadDoc()`.

Output:



1 The header text [inside the <div>](#) element is displayed.

2 Click the button to call [ajax function](#).



1 Button click event is finished.

2 Now the [<div>](#) element displays the text file content due to ajax request and [response processes](#).