

---

## Ajax - XMLHttpRequest

### The XMLHttpRequest

---

- ✿ All latest browsers lend their support for the new built-in JavaScript XMLHttpRequest object (apart from IE5 and IE6, as they use an [ActiveXObject](#)).

### Steps for using XMLHttpRequest:

---

1. Declare and initialize XMLHttpRequest object.  
`xmlhttp=new XMLHttpRequest();`
2. If using IE5 or IE6 browser, Declare and initialize an ActiveXObject.  
`xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");`
3. Open the request object to assign inputs.  
`xhttp.open("GET", "WikiTechy_Employee1_Details.txt", true);`
4. Send the created request object to the web server:  
`xmlhttp.send(null);`



5. Update the partial page with the response received from the web  
`server.document.getElementById("text").innerHTML=  
xhttp.responseText;`

## XMLHttpRequest Open—Using False

- ✱ The open method has third parameter value as "false". This means that the `XMLHttpRequest` object must wait until the response is received from the server. Hence this becomes a synchronous task. This makes the website to slow its `processing capabilities and hence not advisable to implement`.

### Example:

```
xhttp.open("GET", "WikiTechy_Employee1_Details.txt",  
false);  
xhttp.send();
```

## XMLHttpRequest Open—Using True

- ✱ Change the third parameter in the open method to the value "true". This means that the `XMLHttpRequest` object continues its other executions after `sending the request to the server`. Simple change the code as follows:

### Example:

```
xhttp.open("GET",  
"WikiTechy_Employee1_Details.txt", true);  
xhttp.send();
```



---

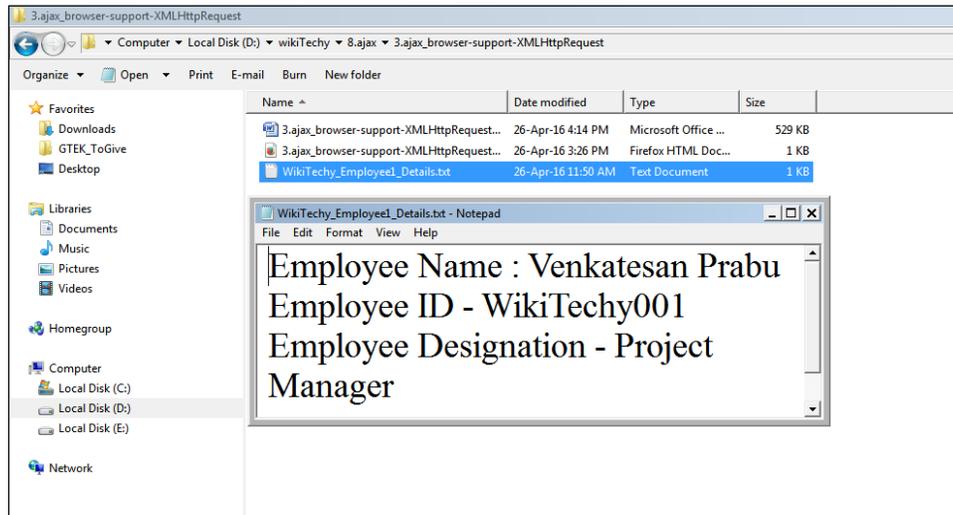
## AJAX Sample Code :

---

```
<html>
  <head>
    <script>
      function loadDoc( )
      {
        var xhttp = new XMLHttpRequest();
        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:

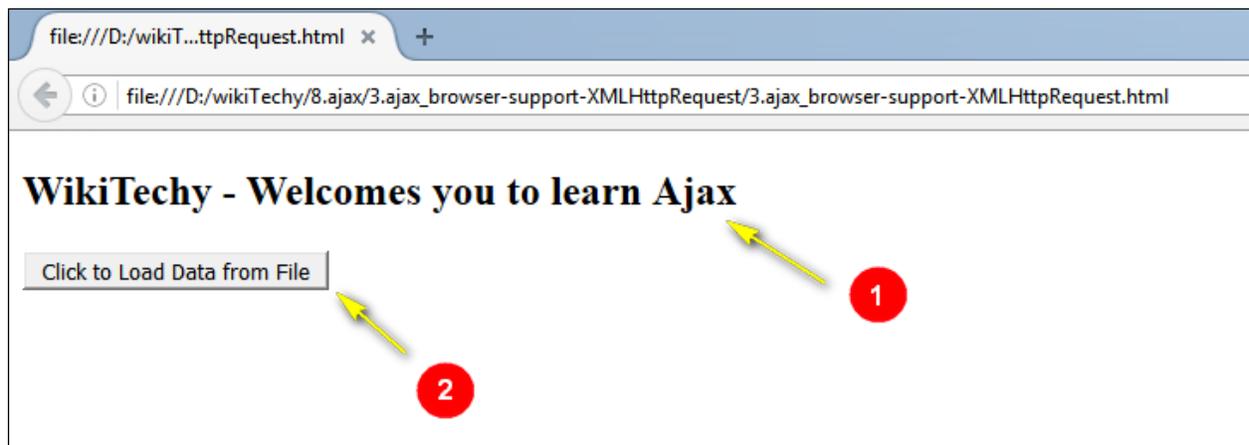
```
<html>
<head>
<script>
function loadDoc( )
{
var xmlhttp;
if (window.XMLHttpRequest)
{ // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp = new XMLHttpRequest();
}
else
{ // code for IE6, IE5
  xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange = function()
{
  if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
  {
    document.getElementById("text").innerHTML = xmlhttp.responseText;
  }
};
xmlhttp.open("GET", "WikiTechy_Employee1_Details.txt", true);
xmlhttp.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 xmlhttp`.
- 4 Check for the `browser type`.
- 5 Initialize the `XMLHttpRequest` object for browser other than `IE5` and `IE6` as `xmlhttp = new XMLHttpRequest();`
- 6 Initialize the `XMLHttpRequest` object for browser `IE5` or `IE6` as `xmlhttp = 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

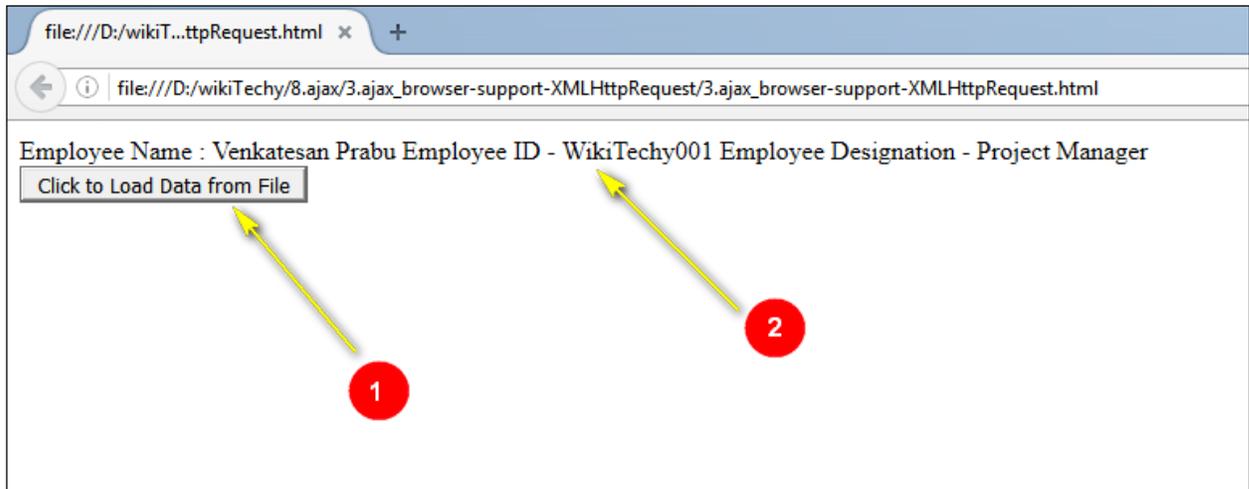
and from server without disturbing any parallel task that is happening in the [HTML code](#).

- 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](#).