

Ajax – Read Xml Data

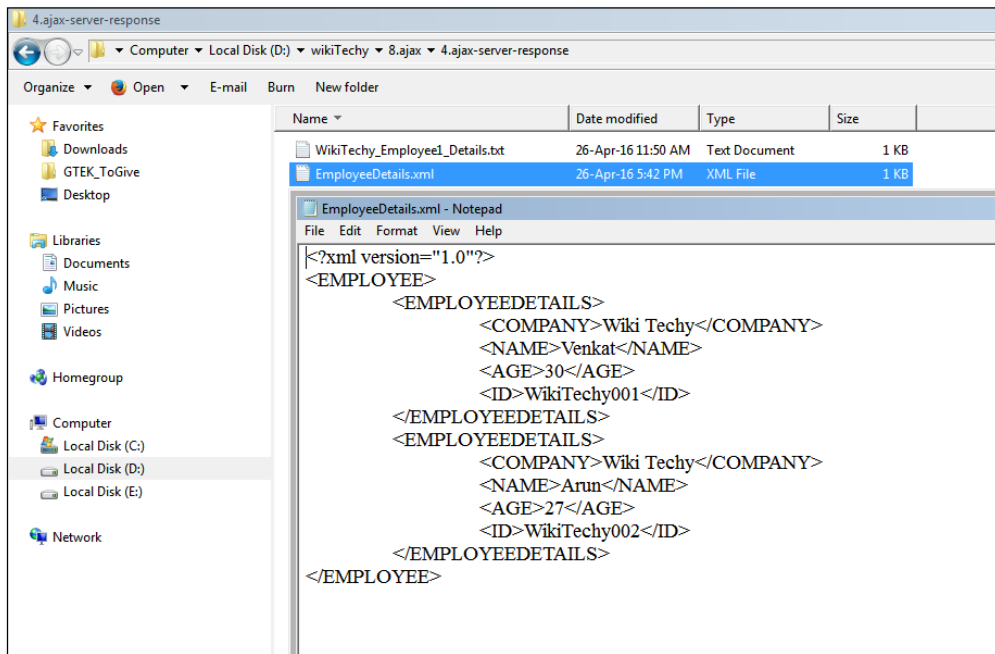
ajax coding - Sample code - Read Xml Data:

```
<html>
<head>
<script>
function loadDoc( )
{
var xhttp, xmlDoc, txt, x, i;
xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function()
{
if (xhttp.readyState == 4 && xhttp.status == 200)
{
xmlDoc = xhttp.responseXML;
txt = "";
x = xmlDoc.getElementsByTagName("NAME");
for (i = 0; i < x.length; i++)
{
txt = txt + "Employee " + i + " Name is : "
+ x[i].childNodes[0].nodeValue + "<br>";
}
document.getElementById("xmldata").innerHTML = txt;
}
};
```



```
xhttp.open("GET", "EmployeeDetails.xml", 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>
  <p id="xmldata"> </p>
</body>
</html>
```

✱ The file content of - EmployeeDetails.xml



ajax Code Explanation

```
<html>
<head>
  <script>
    function loadDoc()
    {
      var xhttp, xmlDoc, txt, x, i;
      xhttp = new XMLHttpRequest();
      xhttp.onreadystatechange = function()
      {
        if (xhttp.readyState == 4 && xhttp.status == 200)
        {
          xmlDoc = xhttp.responseXML;
          txt = "";
          x = xmlDoc.getElementsByTagName("NAME");
          for (i = 0; i < x.length; i++)
          {
            txt = txt + "Employee " + i + " Name is : "
              + x[i].childNodes[0].nodeValue + "<br>";
          }
          document.getElementById("xmldata").innerHTML = txt;
        }
      };
      xhttp.open("GET", "EmployeeDetails.xml", 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>
  <p id="xmldata"></p>
</body>
</html>
```

1 <script> HTML element to contain ajax code.

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

3 Declare the required variables.

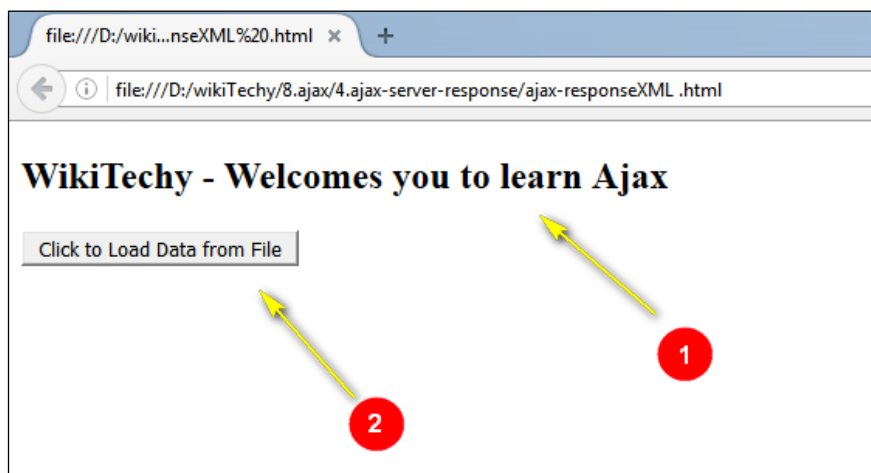


- 4 Initialize the XMLHttpRequest object.
- 5 Capture the state change of the created object. This occurs when the server receives a request and sends its response.
- 6 Check for the readyState and status property values. It should be equal to 4 and 200 respectively. This means that response from the server has been received completely.
- 7 Assign the response xml to a declared variable.
- 8 Initialize the txt variable with empty string.
- 9 Use the xmlhttprequest method getElementByTagName method to read the content of the <NAME> element in XML file into the variable x.
- 10 Now the variable x is a string array containing values of both NAME elements from the XML document.
- 11 Assign the names one by one to the txt variable.
- 12 Add
 tag to create a separation between the names.
- 13 Now assign the value of txt variable to the <p> element – xmldata's innerHTML attribute.
- 14 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.

- 15 Finally call the `send()` to send the request.
- 16 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()`.
- 17 Create a `<p>` element with id as "xmldata" to bind the server response from ajax code.

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 `<p>` element displays the [xml file names](#) due to [ajax request and response](#) processes.