K.T.S.P.Madal's

Hutatma Rajguru Mahavidyalaya, Rajgurunagar

Tal-Khed, Dist.-Pune 410505.

TY.BSc (Computer Science)

Semester-VI

Subject- Web Technologies II

According to new CBCS syllabus w.e.f.2019-2020

Prof.S.V.Patole

Department of Computer Science

Hutatma Rajguru Mahavidyalaya,

Rajgurunagar.

Chapter 4. AJAX

Introduction of AJAX:

Facebook, Instagram, Twitter etc are considered the situation when check news feed and if like someone post simply click the like button and the like count is added without refreshing the page. Now imagine the situation if there would be the case, click the like button and the complete page would be loaded again which will make such processes. Now the question whether clicking the button again for such a small task

Definition:

Ajax is an acronym for Asynchronous Javascript and XML. It is used to communicate with the server without refreshing the web page and thus increasing the user experience and better performance.

Prerequisites:

There are no such pre-requisites required to understand the latter portion of the article. Only the basic knowledge of HTML, CSS, and Javascript are good to go.

How does it work?

First, let us understand what does asynchronous actually mean. There are two types of requests synchronous as well as asynchronous. Synchronous requests are the one which follows sequentially i.e if one process is going on and in the same time another process wants to be executed, it will not be allowed that means the only one process at a time will be executed. This is not good because in this type most of the time CPU remains idle such as during I/O operation in the process which are the order of magnitude slower than the CPU processing the instructions. Thus to make the full utilization of the CPU and other resources use asynchronous calls.

Basic Syntax: The syntax of creating the object is given below

```
req = new XMLHttpRequest();
```

There are two types of methods open() and send(). Uses of these methods explained below.

```
req.open("GET", "abc.php", true);

req.send();
```

Application of Ajax:

AJAX is not a programming language instead, it is a method of accessing data from the server asynchronously and updating the web pages without refreshing/reloading them. In simple words, AJAX is a technique to perform operations that requires server interactions without reloading the web pages again and again.

AJAX stands for Asynchronous Javascript and XML, as the name suggests the process is out of sync and occurs in the background without disturbing the main process thread. The pre-requisites are basic knowledge of JavaScript, XML, and HTML.
Some of the most important applications of AJAX are as follows.

- Updating a webpage without reloading the page.
- Requesting data from the server after the page has been loaded.
- Receiving data from the server after the page has been loaded.
- Sending data to the server in the background without disturbing UI or other processes.

Advantages of Ajax:

1. Ajax improves performance

AJAX is used to retrieve or save data to and from the server without having to post the complete page. This way, you only perform a part instead of the complete postback.

2.Asynchronous processing

You can use AJAX to create asynchronous connections, where users can interact on the front end without waiting for replies from the server. This way, you can make asynchronous calls to the webserver.

3. Improves response time

Considering we are only transmitting required data to the server, the improvement in responsiveness comes as no surprise. In fact, AJAX is popularly used by developers to build responsive web pages.

4. Reduces bandwidth usage

One of the most frustrating issues that plague a website is the server's bandwidth. Gratefully, AJAX is well-equipped to handle such issues, as it reduces bandwidth usage significantly.

5. Multi-browser support

AJAX is supported by commonly used major internet browsers. A few of these browsers include Microsoft Internet Explorer 5 and above, Mozilla Firefox 1.0 and higher, Opera 7.6 and newer, and Apple Safari 1.2

6. Enhances user experience

As a user, you expect websites to offer you a faster browsing experience. The websites using AJAX techniques allow you to go through parts of the web pages without having to reload the page

Disadvantages of Ajax:

1. Dependency

You cannot run AJAX procedures on any browser. The reason: it is dependent on JavaScript. Therefore, the browser should support JavaScript or XMLHttpRequest.

2. Indexing problems

The importance of search engines cannot be underestimated. Hence, websites focus on enhancing their SEO (Search Engine Optimization). You can enjoy the perks of improved performance and speed with AJAX, but they come at a cost.

3. Server inaccessibility

You cannot fetch information from other servers when using AJAX. The reason lies in the fact an XMLHttpRequest object can fetch information only from those servers pages are hosted on.

4. Debugging difficulties

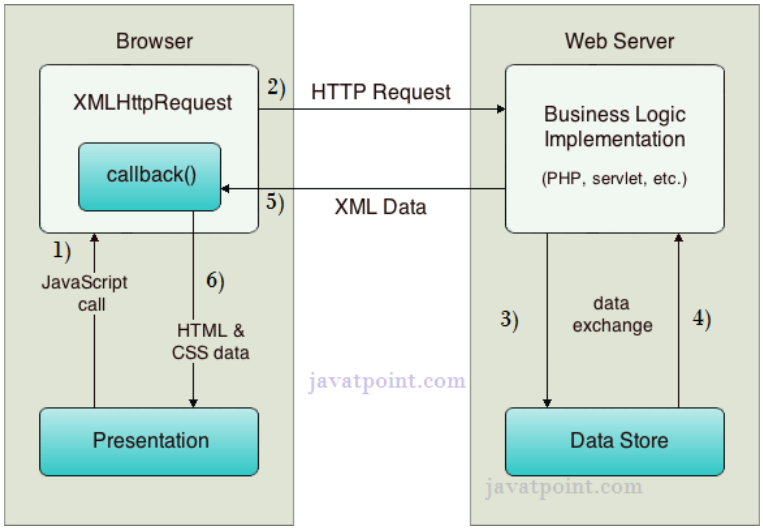
The debugging process can be quite tricky in AJAX requests/responses. Moreover, the error messages aren't robust either. This further increases the complexity of understanding the root of the problem.

Difference between ajax and javascript

Parameters of Comparison	Javascript	Ajax
Definition	A scripting, open-source programming language in web development used for client-side tasks.	It is not a programming language but a technology that is a part of javascript and used for standalone applications also.
Functions	It performs client-side operations and makes a request to the server.	It does all the work of the server-side including sending and receiving information from the server.
Language supported	It supports client-side scripting language.	It supports server-side scripting language.
Web page loading	It doesn't support page loading once it is done for the first time.	It supports page loading multiple times once the page is loaded for the first time.

How AJAX works?

AJAX communicates with the server using XMLHttpRequest object. Let's try to understand the flow of ajax or how ajax works by the image displayed below.



As you can see in the above example, XMLHttpRequest object plays a important role.

1. User sends a request from the UI and a javascript call goes to XMLHttpRequest object.
2. HTTP Request is sent to the server by XMLHttpRequest object.
3. Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
4. Data is retrieved.
5. Server sends XML data or JSON data to the XMLHttpRequest callback function.
6. HTML and CSS data is displayed on the browser.

Creating Sample AJAX Application:

```
<!DOCTYPE html>
<html>
<body>

<div id="demo">
<h1>The XMLHttpRequest Object</h1>
<button type="button" onclick="loadDoc()">Change Content</button>
</div>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

Output:

HandlingPHP Data using Php and Ajax:

```
<html>
<body>

<h2>The XMLHttpRequest Object</h2>
<h3>Start typing a name in the input field below:</h3>

<p>Suggestions: <span id="txtHint"></span></p>
<p>First name: <input type="text" id="txt1" onkeyup="showHint(this.value)">
</p>

<script>
function showHint(str) {
  if (str.length == 0) {
    document.getElementById("txtHint").innerHTML = "";
    return;
  }
  const xhttp = new XMLHttpRequest();
  xhttp.onload = function() {
    document.getElementById("txtHint").innerHTML =
    this.responseText;
  }
  xhttp.open("GET", "gethint.php?q="+str);
  xhttp.send();
}
</script>
```

Output:

The XMLHttpRequest Object

Start typing a name in the input field below:

Suggestions: Anna, Amanda

First name: a

Handling XML Data Using PHP and AJAX:

Below example demonstrate how to parser xml with web browser.

```
<html>
  <head>

    <script>
      function showCD(str) {
        if (str == "") {
          document.getElementById("txtHint").innerHTML = "";
```

```
            return;
         }

         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("txtHint").innerHTML = xmlhttp.responseText;
            }
         }
         xmlhttp.open("GET","getcourse.php?q="+str,true);
         xmlhttp.send();
      }
   </script>

</head>
<body>

   <form>
      Select a Course:
      <select name = "cds" onchange = "showCD(this.value)">
         <option value = "">Select a course:</option>
         <option value = "Android">Android </option>
         <option value = "Html">HTML</option>
         <option value = "Java">Java</option>
         <option value = "Microsoft">MS technologies</option>
      </select>
   </form>

   <div id = "txtHint"><b>Course info will be listed here...</b></div>

</body>
</html>
```

The above example will call getcourse.php using with GET method. getcourse.php file loads catalog.xml. getcourse.php is as shown below −

```
<?php
   $q = $_GET["q"];

   $xmlDoc = new DOMDocument();
   $xmlDoc->load("catalog.xml");

   $x = $xmlDoc->getElementsByTagName('COURSE');
```

```
  for ($i = 0; $i<=$x->length-1; $i++) {
    =
    if ($x->item($i)->nodeType == 1) {
      if ($x->item($i)->childNodes->item(0)->nodeValue == $q) {
        $y = ($x->item($i)->parentNode);
      }
    }
  }

  $cd = ($y->childNodes);

  for ($i = 0;$i<$cd->length;$i++) {
    if ($cd->item($i)->nodeType == 1) {
      echo("<b>" . $cd->item($i)->nodeName . ":</b> ");
      echo($cd->item($i)->childNodes->item(0)->nodeValue);
      echo("<br>");
    }
  }
?>
```

Catalog.xml

XML file having list of courses and details.This file is accessed by getcourse.php

```
<CATALOG>
  <SUBJECT>
    <COURSE>Android</COURSE>
    <COUNTRY>India</COUNTRY>
    <COMPANY>TutorialsPoint</COMPANY>
    <PRICE>$10</PRICE>
    <YEAR>2015</YEAR>
  </SUBJECT>

  <SUBJECT>
    <COURSE>Html</COURSE>
    <COUNTRY>India</COUNTRY>
    <COMPANY>TutorialsPoint</COMPANY>
    <PRICE>$15</PRICE>
    <YEAR>2015</YEAR>
  </SUBJECT>

  <SUBJECT>
    <COURSE>Java</COURSE>
    <COUNTRY>India</COUNTRY>
    <COMPANY>TutorialsPoint</COMPANY>
    <PRICE>$20</PRICE>
    <YEAR>2015</YEAR>
  </SUBJECT>

  <SUBJECT>
```

```
    <COURSE>Microsoft</COURSE>
    <COUNTRY>India</COUNTRY>
    <COMPANY>TutorialsPoint</COMPANY>
    <PRICE>$25</PRICE>
    <YEAR>2015</YEAR>
  </SUBJECT>
</CATALOG>
```

It will produce the following result −

connecting database using php and ajax on tutorial point

HTML file which is ajax.html and it will have following code

```
<html>
  <body>

    <script language = "javascript" type = "text/javascript">
      <!--
        //Browser Support Code
        function ajaxFunction(){
          var ajaxRequest;  // The variable that makes Ajax possible!

          try {
            // Opera 8.0+, Firefox, Safari
            ajaxRequest = new XMLHttpRequest();
          }catch (e) {
            // Internet Explorer Browsers
            try {
              ajaxRequest = new ActiveXObject("Msxml2.XMLHTTP");
            }catch (e) {
              try{
                ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
              }catch (e){
                // Something went wrong
                alert("Your browser broke!");
                return false;
```

```
            }
          }
        }

        // Create a function that will receive data
        // sent from the server and will update
        // div section in the same page.

        ajaxRequest.onreadystatechange = function(){
          if(ajaxRequest.readyState == 4){
            var ajaxDisplay = document.getElementById('ajaxDiv');
            ajaxDisplay.innerHTML = ajaxRequest.responseText;
          }
        }

        // Now get the value from user and pass it to
        // server script.

        var age = document.getElementById('age').value;
        var wpm = document.getElementById('wpm').value;
        var sex = document.getElementById('sex').value;
        var queryString = "?age=" + age ;

        queryString +=  "&wpm=" + wpm + "&sex=" + sex;
        ajaxRequest.open("GET", "ajax-example.php" + queryString, true);
        ajaxRequest.send(null);
      }
    //-->
  </script>

  <form name = 'myForm'>
    Max Age: <input type = 'text' id = 'age' /> <br />
    Max WPM: <input type = 'text' id = 'wpm' />
    <br />

    Sex: <select id = 'sex'>
      <option value = "m">m</option>
      <option value = "f">f</option>
    </select>

    <input type = 'button' onclick = 'ajaxFunction()' value = 'Query MySQL'/>

  </form>

  <div id = 'ajaxDiv'>Your result will display here</div>
 </body>
</html>
```

NOTE − The way of passing variables in the Query is according to HTTP standard and the have formA.

URL?variable1=value1;&variable2=value2;

Now the above code will give you a screen as given below

NOTE − This is dummy screen and would not work.

Max Age:

Max WPM:

Sex:

Your result will display here

Server Side PHP file

So now your client side script is ready. Now we have to write our server side script which will fetch age, wpm and sex from the database and will send it back to the client. Put the following code into "ajax-example.php" file.

```
<?php

  $dbhost = "localhost";
  $dbuser = "dbusername";
  $dbpass = "dbpassword";
  $dbname = "dbname";

  //Connect to MySQL Server
  mysql_connect($dbhost, $dbuser, $dbpass);

  //Select Database
  mysql_select_db($dbname) or die(mysql_error());

  // Retrieve data from Query String
  $age = $_GET['age'];
  $sex = $_GET['sex'];
  $wpm = $_GET['wpm'];

  // Escape User Input to help prevent SQL Injection
  $age = mysql_real_escape_string($age);
  $sex = mysql_real_escape_string($sex);
  $wpm = mysql_real_escape_string($wpm);

  //build query
  $query = "SELECT * FROM ajax_example WHERE sex = '$sex'";

  if(is_numeric($age))
  $query .= " AND age <= $age";

  if(is_numeric($wpm))
  $query .= " AND wpm <= $wpm";
```

```
  //Execute query
  $qry_result = mysql_query($query) or die(mysql_error());

  //Build Result String
  $display_string = "<table>";
  $display_string .= "<tr>";
  $display_string .= "<th>Name</th>";
  $display_string .= "<th>Age</th>";
  $display_string .= "<th>Sex</th>";
  $display_string .= "<th>WPM</th>";
  $display_string .= "</tr>";

  // Insert a new row in the table for each person returned
  while($row = mysql_fetch_array($qry_result)) {
    $display_string .= "<tr>";
    $display_string .= "<td>$row[name]</td>";
    $display_string .= "<td>$row[age]</td>";
    $display_string .= "<td>$row[sex]</td>";
    $display_string .= "<td>$row[wpm]</td>";
    $display_string .= "</tr>";
  }
  echo "Query: " . $query . "<br />";

  $display_string .= "</table>";
  echo $display_string;
?>
```

Now try by entering a valid value in "Max Age" or any other box and then click Query MySQL button.

Max Age:

Max WPM:

Sex: