

Ajax

What is AJAX?

- AJAX = Asynchronous JavaScript and XML.
- AJAX is a technique for creating fast and dynamic web pages.
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
- Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.
- Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.
- Developed by JESSE JAMES GARRET. It is Ajax not AJAX(not an acronym)

Components of Ajax Technology

Ajax is based on many existing technologies such as CSS, HTML, DHTML and JavaScript. If a programmer knows all the above then learning Ajax is easy and it requires less time. A brief overview about the components of Ajax Technology. Ajax application uses the following technology in combination:

- **XHTML, HTML and CSS**
These are used for creating the user interface and styling the web pages to make it more appealing.
- **DOM**
The Document Object Model or DOM is used by the JavaScript code to produce interactive Ajax applications.
- **XMLHttpRequest or XMLHttpRequest**
XMLHttpRequest or XMLHttpRequest is used to retrieve the data from server.
- **JavaScript**
JavaScript is used to bind everything together.

Working of Ajax – Steps Followed

1) Getting the HTTP Request Object:

Ajax uses JavaScript to make a request from the server using request object. The following two classes can be used to create request object based on the browser support

- a. XMLHttpRequest (For Internet Explorer browser 4.0 , 5.0 and before).
- b. XMLHttpRequest (For Mozilla, Safari, IE 8.0etc.)

The browser independent code for creating a request object is as follows:

```
var req;  
if (window.XMLHttpRequest) {  
    // Mozilla, Safari, IE8.0...  
    var req = new XMLHttpRequest();  
}  
else if (window.ActiveXObject) {  
    // IE4.0 5.0  
    var req = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

2) Deciding the method to be called after receiving the response from the server:

The next step is to decide which JavaScript function is to be called after receiving the response from the server. In this method write the code to process and manipulate the data received from the server. There are two ways to set the function for this purpose:

- a. **Set the property onreadystatechange of the Http request object to the name of that JavaScript function**

For example, the JavaScript function **doSomething()** is to be executed after receiving the response from the server. Then write the code like below:

```
req.onreadystatechange = doSomething;  
// No bracket after the function
```

This line is not for calling the method **doSomething** but only for assigning a reference to the function, which will be called after receiving the response from the server.

- b. **Using anonymous function:**

Write the code of the function just at the same place instead of referencing the name of the function. For example, like below:

```
req.onreadystatechange = function(){//code inside}
```

3) Making an HTTP Request

Actually no request is made in the previous steps. Only the method that would process the response data after receiving the data from the server is decided. So now it's turn to send the request to the server.

For this two methods of Http request object' class:

i) **open ()**

ii) **send()**

These methods can be used as given in the code snippet below:

```
//post  
req.open("POST", "time.php", true);  
req.send("name=abc&age=25");
```

open():

This method can take three parameters:

- First Parameter:
Name of HTTP request method (GET, POST, HEAD etc.)
- Second Parameter:
URL of the requested page to read data from.
- Third Parameter:
A boolean value (TRUE, FALSE), to specify whether the request is asynchronous or synchronous.

If it is set to TRUE the request is set as asynchronous i.e. the browser continues the execution of JavaScript function even when the response has not been received from the server. If it is set to FALSE then request is set to synchronous. In that case the browser waits for the response of the server, which you may not prefer in case of fetching lot of data from the server.

Asynchronous - True or False?

AJAX stands for Asynchronous JavaScript and XML, and for the XMLHttpRequest object to behave as AJAX, the async parameter of the open() method has to be set to true: **req.open("GET","ajax_test.jsp",true);**

Sending asynchronous requests is a huge improvement for web developers. Many of the tasks performed on the server are very time consuming. Before AJAX, this operation could cause the application to hang or stop.

With AJAX, the JavaScript does not have to wait for the server response, but can instead:

- execute other scripts while waiting for server response
- deal with the response when the response ready

async=true

When using async=true, specify a function to execute when the response is ready in the onreadystatechange event:

Async=false

To use async=false, change the third parameter in the open() method to false:

req.open("GET","ajax_info.php",false);

Using async=false is not recommended, but for a few small requests this can be ok. The JavaScript will NOT continue to execute, until the server response is ready. If the server is busy or slow, the application will hang or stop. When async=false, then, do NOT write an onreadystatechange function - just put the code after the send() statement:

Send(): - POST Method

This method is used to send data to the server with the request. If no query string is required to be sent then pass null to send.

req.send(null)

The send() method can be used to send any data to the server as key and value pair. For example,

```
req.send("name=abc")
```

Any number data can be sent to the server by creating the query string like this:

```
req.send("name=abc&age=25")
```

If HTTP request method is of type "POST" then change the MIME type of the request otherwise the server will not accept such data. So, change the code snippet given below:

```
req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
```

```
req.open("POST", "time.php", true);  
req.send("name=abc&age=25")  
req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
```

Send() get method

This method is used to send data to the server with the request. The query string, if any, must be appended to the server-side script in the open function and a null is sent to the send function.

```
var url = "/SearchEng/SearchServlet?search="+search;  
req.open("get", url, true);  
req.send(null);  
req.onreadystatechange = callback;
```

4) Working with the Response data:

The response from server can be either a Text or XML. If the response is Text then it can be processed directly as String and if the response is XML then the DOM should be used to process it.

i) Checking the state of the request:

The readyState property of the HTTP request object req is used to check the state of the request. This property can have different values indicating the state of the request.

Value	State of the request
0	Un-initialized
1	Loading

2	Loaded
3	Interactive
4	Complete

The method above checks first the state of the request. If it is equal to 4 it means the request has completed and the response is received from the server.

ii) Checking the Status Code of the response:

The next thing that should be checked is the Status Code of the HTTP server response. The server can send different status code depending on the request processing of the server.

Status	Code Definition
200	OK
400	Bad Request
401	Unauthorized
404	Not Found
405	Method Not Allowed
500	Internal Server Error

But the status code “200” is checked normally to check if everything is “OK”. After checking the state and status code then it is safe to proceed further. Now data from the server can be manipulated.

iii) Accessing the response data:

We have two options to collect the data:

- req.responseText:** It is used to access plain text response from the server
- req.responseXML**

To dynamically determine the response types you can write:

The possible returned content types are ‘text/xml’ or ‘text/plain’ indicating XML or plain text content respectively.

```
var contentType = xmlhttp.getResponseHeader("content-type");
```

Using anonymous function

```
req.onreadystatechange=function ()
{
  if (req.readyState==4)
  {
    alert (req.responseText) ;
  }
}
```

```

document.getElementById("p").innerHTML=req.responseText;
document.getElementById("p").style.display="none";
document.aj.datel.value =
document.getElementById("p").innerHTML;
    }
}

req.open("GET","dp.jsp",true);
req.send(null);

```

Using named function to process and access response data

```

var url="getuser.jsp"
    url=url+"?emp_id="+emp_value
    xmlHttp.onreadystatechange=stateChanged
    xmlHttp.open("GET",url,true)
    xmlHttp.send(null)

function stateChanged()
{
    document.getElementById("ename").value ="";
    document.getElementById("emp_id").value ="";
    if (xmlHttp.readyState==4)
    {
        var showdata = xmlHttp.responseText;

        var strar = showdata.split(":");
        if(strar.length==1)
        {

document.getElementById("emp_id").focus();
alert("Please Select Employee Id");
document.getElementById("ename").value = " ";
document.getElementById("emp_id").value = " ";

        }
        else if(strar.length>1)
        {
            document.getElementById("emp_id").value= strar[0];
            document.getElementById("ename").value= strar[1];
        }
    }
}

```

Ajax with PHP

Aj.html

```
<html>
<head><title>AJAX</title>
<script>
var req;
function dateDisplay()
{
    if( window.XMLHttpRequest)
        req = new XMLHttpRequest();
    else
        req = new ActiveXObject("Microsoft.XMLHttp");

    req.onreadystatechange=function()
    {
        if (req.readyState==4 && req.status==200)
        {
            document.aj.datel.value = req.responseText;
        }
    }

    req.open("GET","d.php",true);
    req.send();
}

</script>
<body>
<form name="aj">
<pre>
Enter UserName      :<input type="text" name="txt" onblur
="dateDisplay()">
Date of Registration:<input type="text" name="datel">
</form>
</pre>
</body>
</html>
```

d.php

```
<?php
    echo date("M/D/Y");
?>
```

Ajax with JSP

Aj1.html

```
<html>
  <head><title>AJAX</title>
    <script type="text/javascript">
var req;
function dateDisplay()
{
  if( window.XMLHttpRequest)
    req = new XMLHttpRequest();
  else
    req = new ActiveXObject("Microsoft.XMLHttp");
  req.onreadystatechange=function()
  {
    if (req.readyState==4){

document.getElementById("p").innerHTML=req.responseText;
    document.getElementById("p").style.display="none";
document.aj.datel.value =
document.getElementById("p").innerHTML;
    }

  }
req.open("GET","dp.jsp",true);
req.send(null);
}
</script></head>
  <body>
<form name="aj" method="get" action="re.jsp">
  <pre>
UserName:<input type="text" name="txt"
      onblur="dateDisplay();">
Date Registration:<input type="text" name="datel" >
<input type="submit" name="submit" value="register">
<p id="p"></p>
</pre>
</form>
</body>
</html>
```

document.getElementById ("p").style.display="none"; is used to hide the element with id "p".

dp.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@page import ="java.util.*" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%
            response.getWriter().write( new
Date().toString());
        %>
    </body>
</html>
```

re.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@page import = "java.sql.*" %>

<html>
<head><title>JSP Page</title></head>
<body>
<%
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:330
6/e","root","amirtha");PreparedStatement
st=con.prepareStatement("insert into std values (?,?)" );
        st.setString( 1 , request.getParameter("txt") );
        st.setString( 2 ,request.getParameter("date1"));
        st.executeUpdate();
        st.close();
        con.close();
    }catch (ClassNotFoundException err) { out.println(err); }
        catch (SQLException err) { out.println(err); }
    %>
</body>
</html>
```

Search Engine using AJAX with servlets

Ser1.html

```
<html>
<head>
<title>AJAX Search</title>
<style type="text/css">
div#result {
display: none;
border:2px solid blue;
}
</style>
<script language="javascript">
var req;
function searchText() {
    var search = document.getElementById("search").value;
    if(window.xmlHttpRequest){
        req = new xmlHttpRequest();
    }
    else if(window.ActiveXObject){
        req = new ActiveXObject("Microsoft.XMLHttp");
    }
    var url = "/SearchEng/SearchServlet?search="+search;
    req.open("get",url,true);
    req.onreadystatechange = callback;
    req.send(null);
}

function callback() {
if( req.readyState==4 ){
    var result = document.getElementById('result');
    result.innerHTML = '';
    var respText = req.responseText;
    alert(respText);
    if((respText.length)!=0){
        document.getElementById('result').style.display =
'block';
        var str = req.responseText.split("\n");
        var items;
        for(i=0; i < str.length - 1; i++) {
            items = '<div onclick="setText(this.innerHTML);"
';
            items += '>' + str[i] + '</div>';
            result.innerHTML += items;
        }
    }
    else
```

```

        {
            document.getElementById('result').style.display =
            'none';
        }
    }
}
function focusIn(){
document.getElementById("search").focus( );
}
function setText(value) {
document.getElementById('search').value = value;
document.getElementById('result').style.display = 'none';
document.getElementById('result').innerHTML = '';
}

</script>
</head>
<body onload="focusIn();">
<h2>Ajax Search Example</h2>
<form>
<b>Enter Search Keyword</b>
<table border = "0">
<tr>
<td><input type="text" id="search" onkeyup="searchText()"
/></td>
</tr>
<tr>
<td><div id="result"></div></td>
</tr>
</table>
<input type="submit" id="search" name="find" value="find"/>
</form>
</body>
</html>

```

Ser.html

```

<html>
<head>
<title>AJAX Search</title>
<style type="text/css">
.normal {
background-color: white;
padding: 2px 6px 2px 6px;
}
.over {
background-color: blue;

```

```
padding: 2px 6px 2px 6px;
}
div#result {
display: none;
border:3px solid blue;
}
</style>
<script language="javascript">
var req;
function searchText() {
    var search = document.getElementById("search").value;
    var url = "/SearchEng/SearchServlet?search="+search;
    if(window.XMLHttpRequest) {
        req = new XMLHttpRequest();
    }
    else if(window.ActiveXObject) {
        req = new ActiveXObject("Microsoft.XMLHTTP");
    }
    req.open("get",url,true);
    req.onreadystatechange = callback;
    req.send(null);
}

function callback() {
if( req.readyState==4 ){
    var result = document.getElementById('result');
    result.innerHTML = '';
    var respText = req.responseText;
    if((respText.length)!=0){
        show_div('result');
        show_div('close');
        var str = req.responseText.split("\n");
        var items;

        for(i=0; i < str.length - 1; i++) {
            items = '<div onmouseover="overText(this);" ' ;
            items += 'onmouseout="outText(this);" ' ;
            items += 'onclick="setText(this.innerHTML);" ' ;
            items += 'class="normal">'
            items+= str[i];
            items+= '</div>';
            result.innerHTML += items;

        }

    }
    else

```

```

        {
            hide_divs();
        }
    }
}
function focusIn(){
document.getElementById("search").focus( );
}
function overText(div_value) {
div_value.className = 'over';
}
function outText(div_value) {
div_value.className = 'normal';
}
function setText(value) {
document.getElementById('search').value = value;
hide_divs();
document.getElementById('result').innerHTML = '';
}
function show_div(div_id) {
document.getElementById(div_id).style.display = 'block';
}
function hide_divs() {
    document.getElementById('result').style.display =
'none';
    document.getElementById('close').style.display = 'none';
}
</script>
</head>
<body onload="focusIn();">
<h2>Ajax Search Example</h2>
<form method="get" action="serServlet">
<b>Enter Search Keyword</b>
<table border = "0">
<tr>
<td><input type="text" id="search" name="search"
onkeyup="searchText()" /></td>
</tr>
<tr>
<td><div id="result"></div></td>
</tr>
<tr>
<td><div id="close" align="right" style="display: none;"><a
href="" onclick="hide_divs();"
return false">close</a></div></td>
</tr>
</table>

```

```
<input type="submit" id="search" name="find" value="find"/>
</form>
</body>
</html>
```

Servlet class called by Ajax- SearchServlet.java – compile to create a class(SearchServlet)

***Create a MySQL table keyword under database search**

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.IOException;
import java.sql.*;

public class SearchServlet extends HttpServlet {

    public void doGet( HttpServletRequest request,
        HttpServletResponse response ) throws
        ServletException, IOException{

        String search = request.getParameter("search");
        response.setContentType("text/html");
        if(!((search.trim()).equals(""))){
            String searchString = getSearchResult(search);
            response.getWriter().write(searchString);
        }
        else
            response.getWriter().write("");
    }

    public String getSearchResult(String search) {
        Connection conn = null;
        Statement st=null;
        ResultSet res=null;
        String finalSearch = "";
        try{
            Class.forName("com.mysql.jdbc.Driver");

            conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/search","root","amirtha");
            st=conn.createStatement();
            String s = "SELECT keyword FROM se WHERE keyword like '"+search+"%' ORDER BY keyword";
            res = st.executeQuery(s) ;
            while (res.next())
```

```

        {

            String un = res.getString(1);
            finalSearch+= un+"\n";

        }
        st.close();
        conn.close();
    }catch(Exception e){}

return finalSearch;
}
}

```

Servlet called on submit –serServlet.Create files f1.txt,f2.txt,f3.txt,f11.html and populate with some content and store it in web directory of the current project file

```

import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class serServlet extends HttpServlet {

    public static boolean searchFile(String S1 , String
S2)throws
        FileNotFoundException,IOException

    {

        File file = new File("E:\\java\\SearchEng\\web\\"+S1);

        String str1;
        FileReader fin = new FileReader(file);
        BufferedReader f = new BufferedReader(fin);
        int c;
        S2 = S2.toUpperCase();
        while ((str1 = f.readLine()) != null )
        {
            str1=str1.toUpperCase();
            if(str1.indexOf(S2) != -1)
            {
                return true;
            }
        }
    }
}

```

```

    }
    return false;

    }
    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException {
        int i;
        boolean f;
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String s = request.getParameter("search");
        String
filenames[]={ "f1.txt", "f2.txt", "f3.txt", "f11.html" };
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Search Engine</title>");
        out.println("</head>");
        out.println("<body>");
        String f1;
        for (i = 0 ; i<filenames.length; i++)
        {
            f = searchFile(filenames[ i ],s);

            if (f){
                f1 =
"http://localhost:8084/SearchEng/"+filenames[i];

                out.println("<a href="+ f1
+">"+"file"+i+"</a>");
                out.println("<br>");
            }

        }
        out.close();

    }
    protected void doGet(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

```



```

        protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
            processRequest(request, response);
        }
    }
}

```

Ajax with database and JSP –jcombo.jsp

```

<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>
<html>
<head>
<script type="text/javascript">
function showEmp(emp_value)
{
    if(document.getElementById("emp_id").value!="-1")
    {
        xmlHttp=GetXmlHttpRequestObject()
        if (xmlHttp==null)
        {
            alert ("Browser does not support HTTP
Request")
            return
        }
        var url="getuser.jsp"
        url=url+"?emp_id="+emp_value
        xmlHttp.onreadystatechange=stateChanged
        xmlHttp.open("GET",url,true)
        xmlHttp.send(null)

    }
    else
    {
        alert("Please Select Employee Id");
    }
}

function stateChanged()
{
    document.getElementById("ename").value = "";
    document.getElementById("emp_id").value = "";
    if (xmlHttp.readyState==4)
    {
        var showdata = xmlHttp.responseText;
    }
}

```

```

        var strar = showdata.split(":");
        if(strar.length==1)
        {

document.getElementById("emp_id").focus();
                alert("Please Select Employee Id");
                document.getElementById("ename").value ="
";
                document.getElementById("emp_id").value
=" ";

        }
        else if(strar.length>1)
        {
                document.getElementById("emp_id").value=
strar[0];
                document.getElementById("ename").value=
strar[1];
        }

    }
}

function GetXmlHttpRequestObject()
{
    var xmlhttp=null;
    try
    {
        // Firefox, Opera 8.0+, Safari
        xmlhttp=new XMLHttpRequest();
    }
    catch (e)
    {
        //Internet Explorer
        try
        {
            xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
        }catch (e){
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    return xmlhttp;
}
</script>

</head>
<body>

```

```

<form name="employee">
<br><br>
<table border="0" width="400px" align="center"
bgcolor="aqua">
    <tr><td><b>Select Employee Id</b></td><td>
        <select name="semp_id"
onchange="showEmp(this.value);">
            <%
                Connection conn;
                Statement st;
                ResultSet res;
                try{

                    Class.forName("com.mysql.jdbc.Driver");

conn=DriverManager.getConnection("jdbc:mysql://localhost:33
06/e","root","amirtha");
                    st=conn.createStatement();
                    res=st.executeQuery("select * from employee2");
                    int i=0;

                    while(res.next())

                        {%>
                            <option
value="<%=res.getInt(1)%>"><%=res.getInt(1)%></option>

                                <% } %>

                            <%
                                st.close();
                                conn.close();
                                }catch (ClassNotFoundException err) { }
                                catch (SQLException err) { } %>

                        </select>
                    </td></tr>
                    <tr><td><b>Employee Id:</b></td><td>
                        <input type="text" name="emp_id" id="emp_id"
value=""></td></tr>
                    <tr><td><b>Employee Name:</b></td><td>
                        <input type="text" name="emp_name" id="ename"
value=""></td></tr>

```

```
</table>
</form>
<table border="0" width="100%" align="center">
<br>
<br>
</table>
</body>
</html>
```

getuser.jsp called from ajax

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>
<%

String emp_id = request.getParameter("emp_id");
String data = "";
    Connection conn;
    Statement st;
    ResultSet res;
    try{

        Class.forName("com.mysql.jdbc.Driver");

conn=DriverManager.getConnection("jdbc:mysql://localhost:33
06/e","root","amirtha");
        st=conn.createStatement();
        String query = "select * from employee2 where
eid= '"+emp_id+"'";
        st = conn.createStatement();
        res = st.executeQuery(query);
        while(res.next())
        {
            data = res.getInt(1) + ":" + res.getString(2)
;
        }
        out.println(data);
    }
    catch (Exception e) {
out.println("exception occured");
    }
%>
```