SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

AJAX Introduction

Course: 19CAT901 – Web Programming Essentials

Unit IV : Advanced Java Script

I Semester / I MCA

AJAX Introduction

AJAX is a developer's dream, because you can:
Update a web page without reloading the page
Request data from a server - after the page has loaded
Receive data from a server - after the page has loaded
Send data to a server - in the background

AJAX Introduction

Before understanding these terms see few practical examples to demonstrate the power 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 which require complete loading of a web page. Absolutely not, because no one will do such an absurd thing (in the latter case).

So it means that this feature of like is very helpful as it prevents the reloading of the complete page. It communicates only necessary information with the server and shows to the end user(in this case being the increase of like count).

AJAX Introduction

Consider another case when visit google to search for anything. Usually, observe that when start typing the desired keywords to search, observe that many suggestions are presented in the search bar.

Now, where do they come from? Of course not from the client side.

The results are again the power of communication with the server without reloading the page again. Like this, there are dozens of examples which can be considered. The whole power working behind is nothing but Ajax. Let’s discuss briefly Ajax and its implementation.

What is Ajax?

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.

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. For more information visit this link. Why the word javascript is present here. Actually, the requests are made through the use of javascript functions. Now the term XML which is used to create

XMLHttpRequestobject

Ajax allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes.

For implementing Ajax, only be aware of XMLHttpRequest object.
Now, what actually it is. It is an object used to exchange data with the server behind the scenes. Try to remember the paradigm of OOP which says that object communicates through calling methods (or in general sense message passing). The same case applied here as well.

Usually, create this object and use it to call the methods which result in effective communication. All modern browsers support the XMLHttpRequest object.

XMLHttpRequestobject

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

req = new XMLHttpRequest();

There are two types of methods open() and send().

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

req.send();

The above two lines described the two methods. req stands for the request, it is basically a reference variable. The GET parameter is as usual one of two types of methods to send the request. Use POST as well depending upon whether send the data through POST or GET method. The second parameter being the name of the file which actually handles the requests and processes them. The third parameter is true, it tells that whether the requests are processed asynchronously or synchronously. It is by default true which means that requests are asynchronous.

The open() method prepares the request before sending it to the server. The send method is used to send the request to the server. Sending the parameter through getting or POST request. The syntax is given below

XMLHttpRequestobject

req.open("GET", "abc.php?x=25", true);
req.send();

In the above lines of code, the specified query in the form of URL followed by ? which is further followed by the name of the variable then = and then the corresponding value. If sending two or more variables use ampersand(&) sign between the two variables. The above method as shown applies for GET request. Sending the data through the POST, then send it in the send method as shown below.

req.send("name=johndoe&marks=99");
req.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Events and handling mechanism

Any action performs on the clicking button, hovering over elements, page loading etc all are termed as events. Also aware of fact that javascript can detect events. So bind the code of specific event with its action which can be implemented by javascript. These are basically even handlers.

Implementing event handlers which actually hold the events. Events handlers are basically functions written in javascript which act on or set into action when an event is fired by the user. When sending the request through send method usually get the response from the server later. But getting of response time is not known. So track it.

Events and handling mechanism

Therefore to keep a track of the response onreadystatechange event which is binding with the event handler(function) which will get executed when a response comes.
When a request to the server is sending perform actions based on the response. The onreadystatechange event is triggered every time the readyState changes. So what actually a ready state is and when will the onreadystate event actually occur and how many times it will occur between the request and response?
The XMLHttpRequest object has a property called as readyState whose value changes in the complete request-response journey i.e when a request is prepared, sent, resolves, processed and when the response comes.

Events and handling mechanism

That’s why it is called os onreadystatechange.

The onreadystatechange stores a function (or the name of the function)to be called automatically each time the readyState property changes.

The readyState holds different values ranging from 0 to 4.

request not initialized

server connection established

request received

processing request

request finished and response is ready

XMLHttpRequest also has a property named as status. The status has following values

200: “OK”

404: “Page not found”

Now remember it always that when readyState is 4 and status is 200, the response is ready.

The whole thing described above is implemented in coding as given below

Events and handling mechanism

```
<p id = "dis">< /p>
req.onreadystatechange = function(){
if(req.readyState == 4 && req.status == 200){
            document.getElementById("dis").innerHTML = req.responseText;
}
}
```

(Note: This is only a section of code and moreover describing the communication between client and server so the code described above will not show any effect if run on IDE)

In the above code if the condition is true (i.e the response is ready) then the result is displayed.

Advantages & Disadvantages

Advantages:

Speed is enhanced as there is no need to reload the page again.

AJAX make asynchronous calls to a web server, this means client browsers avoid waiting for all the data to arrive before starting of rendering.

Form validation can be done successfully through it.

Bandwidth utilization – It saves memory when the data is fetched from the same page.

More interactive.

Disadvantages:

Ajax is dependent on Javascript. If there is some Javascript problem with the browser or in the OS, Ajax will not support.

Ajax can be problematic in Search engines as it uses Javascript for most of its parts.

Source code written in AJAX is easily human readable. There will be some security issues in Ajax.

Debugging is difficult.

Problem with browser back button when using AJAX enabled pages.

AJAX Example

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

<div id="demo">
  <h2>Let AJAX change this text</h2>
  <button type="button" onclick="loadDoc()">Change Content</button>
</div>

</body>
</html>
```

The HTML page contains a <div> section and a <button>.

The <div> section is used to display information from a server.

The <button> calls a function (if it is clicked).

The function requests data from a web server and displays it:

AJAX Example

Function loadDoc()

```
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();
}
```

The "ajax_info.txt" file used in the example above, is a simple text file and looks like this:

AJAX Example

```
<h1>AJAX</h1>
<p>AJAX is not a programming language.</p>
<p>AJAX is a technique for accessing web servers from a web page.</p>
<p>AJAX stands for Asynchronous JavaScript And XML.</p>
```

What is AJAX?

AJAX = **A**synchronous **J**avaScript **A**nd **X**ML.

AJAX is not a programming language.

AJAX just uses a combination of:

A browser built-in XMLHttpRequest object (to request data from a web server)

JavaScript and HTML DOM (to display or use the data)

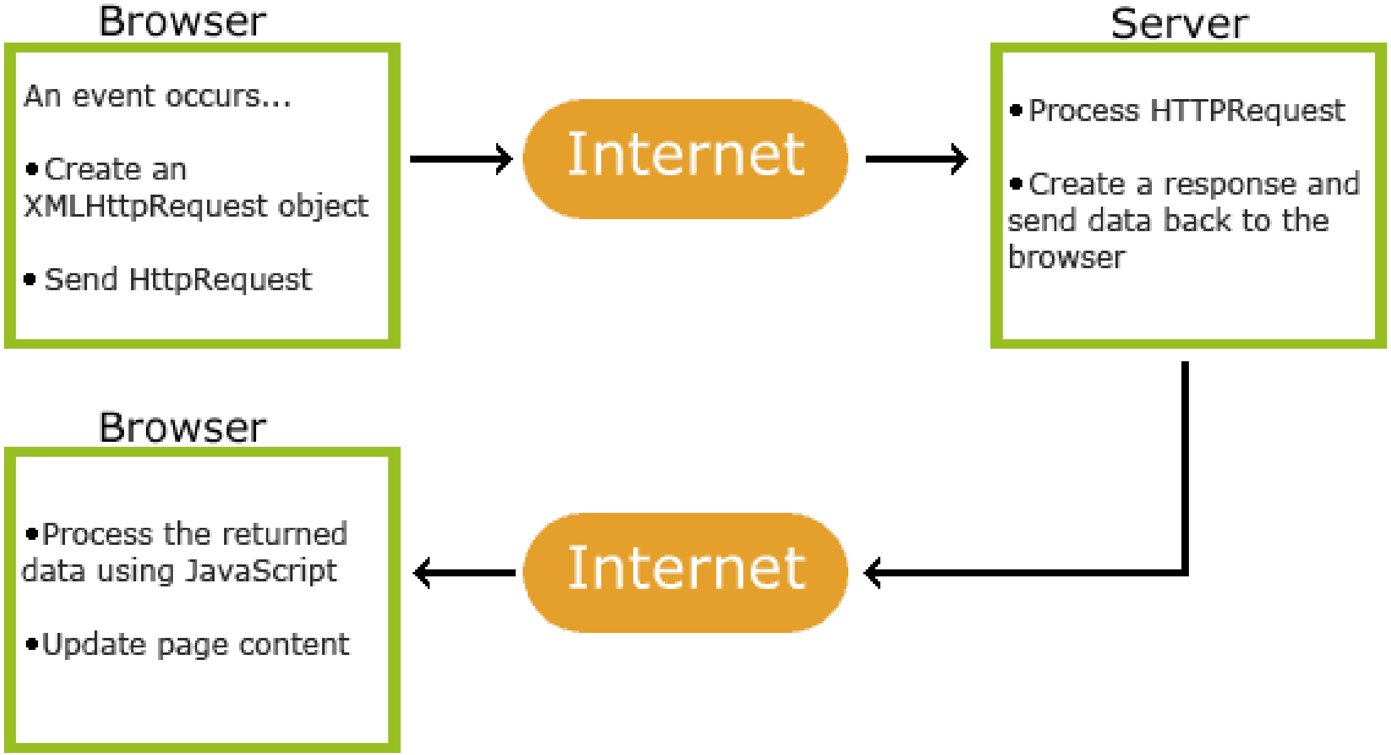
AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

How AJAX Works

1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

How AJAX Works

How AJAX Works



Browser

An event occurs...

- Create an XMLHttpRequest object
- Send HttpRequest

Internet

Server

- Process HTTPRequest
- Create a response and send data back to the browser

Internet

Browser

- Process the returned data using JavaScript
- Update page content

References

https://www.w3schools.com/xml/ajax_intro.asp

https://www.geeksforgeeks.org/ajax-introduction/

https://www.tutorialspoint.com/ajax/index.htm