Unit-4- ~~Ajax~~ AJAX

- AJAX - Asynchronous Javascript And XML

* Introduction

Ajax stands for Asynchronous javascript and xml

- it allows us to send and receive data asynchronously without reloading / refreshing The entire webpage.
- Ajax allows :

1) Update a webpage without reloading or refreshing entire page it means it is Possible To update Parts of webpage without reloading whole Page.
2) Request / Recieve Data from Server after The Page has loaded.
3) To Create fast and dynamic webPages
4) Creating faster and more ~~cutter~~ interactive web application.

- Ajax applications are browser independent
- Ajax applications are Platform Independent
- ~~Ajax~~ Applications There are number of web application That are using Ajax Technology Such as Google maps, Gmail, Youtube, facebook, Google Suggest etc.

* fundamentals of Ajax technology

Ajax is group of inter-related technologies includes :

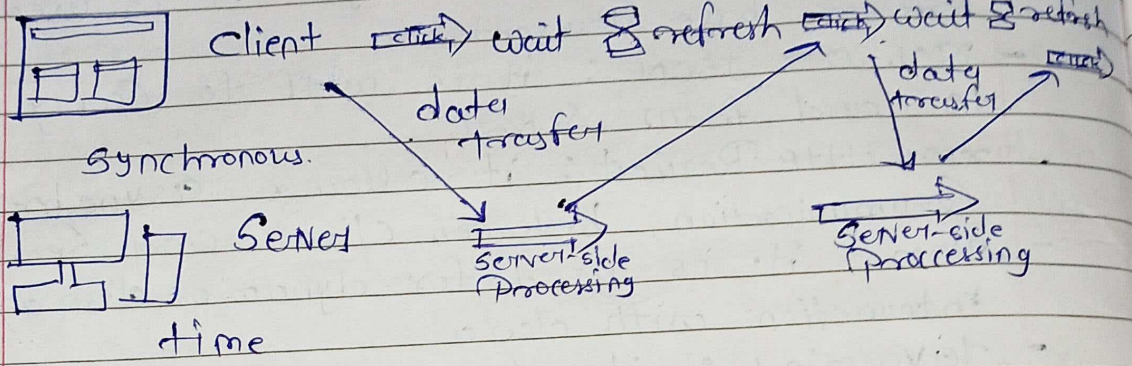
- HTML/XHTML and CSS : they are used to display Content and style. it is used for browser base Presentation.
- XML or JSON : it is used to Carry data to and from Server.
- XML HttpRequest : it allows asynchronous Communication between client and Server.
- Dom : it is used for dynamic display and interaction with data
- Javascript : it bring all Technologies Together.

* Web Application model

- Synchronous web Application Model (Classic)

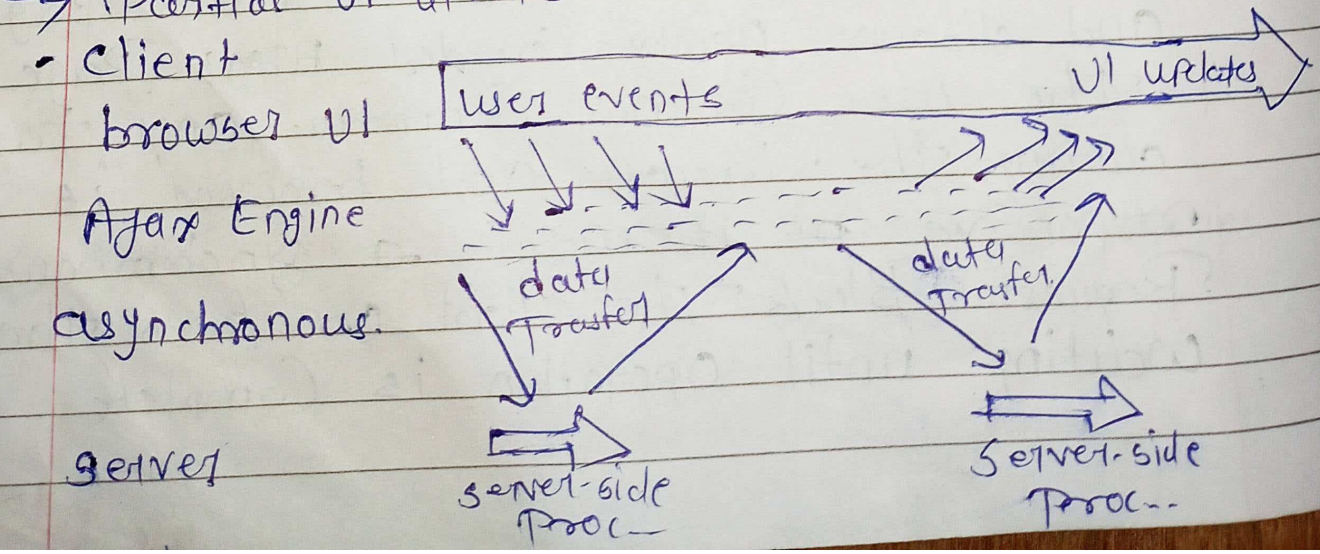
- classic or traditional web Application model is Synchronous it uses Start - Stop - Start - Stop Approach in Communication with Server.
- In order to responed User Action browser discard Current HTML Page Then request is sent to Server when Server finished the Processing request response is return to browser
- In Last browser refresh the entire Page and display newly Created HTML Page
- The user Can not Perform any task during the Process and browser is un-responsive it means synchronous Request Block The client or keep the user waiting until operation is Complete

Full Page refresh



* Ajax web Application model [Asynchronous]
Ajax web Application ~~may~~ model is asynchronous. it doesn't use start--stop-start-stop approach in Communication with Server, it uses Ajax ~~and~~ Engine. Ajax Engine is written in Javascript. Ajax Engine Performs interaction between user interface and Ajax Engine on the client side. Asynchronous interaction ~~doesn't~~ does not block the client it means browser is responsive at that time user can Perform another operation or interact with browser.

→ Partial UI updates



* XMLHttpRequest (XHR) object.

- XHR object is a key Ajax
- it is use to interact or exchange the data with Server.
- This is an asynchronous communication as a result it is Possible to update Possion of webPage without reloading the entire Page
- it is supported by all moden browser.
- it Can be use with javascript, VB.Script and other Scripting Languages to transfer and manupulate xml data. it Can be used to retrive data in verity of formate Such as PlanText, JSON and So on.
- XHR Perform following Tasks...

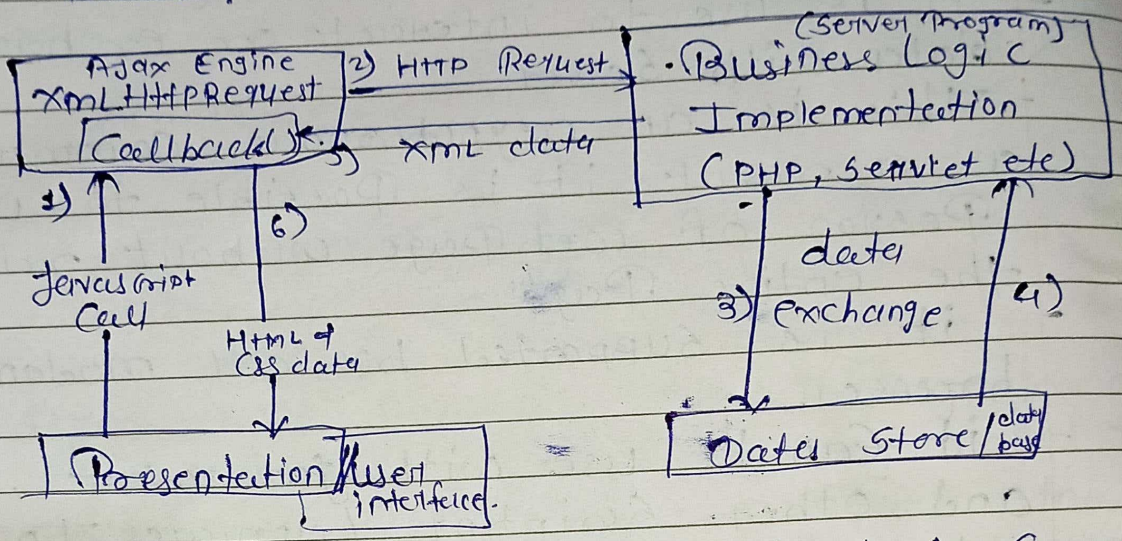
1) Send data from the client to Server in the background
2) Recieve data from Server
3) Update the webPage without having to reload it.

* Create an XMLHttpRequest object.

Syntax for Creating an XMLHttpRequest object:

Variable = new XMLHttpRequest();

* coorking of Ajax and it's Architecture.



Ajax uses XMLHttpRequest object to Communicate with Server. The diagram above illustrates the flow of coorking of Ajax. In diagram following symbol notations are use

1) Javascript Call
2) Http Request
3) 4) Data exchange
5) XML or JSON Data.
6) HTML & CSS Data

- Ajax Create an instant of XHR object. and Send HttpRequest to the Server the Server Process the request Send data back to the browser
- The web browser Processes the data using Javascript and update the data of web page The following steps shows the flow of Ajax.

1) When user Sends a Request Javascript Create XMLHttpRequest object.
2) XHR object Create HTTP Request and Send to the Server using Send() method
3) The Server Process the Request and interact with database
4) Data is Retrive
5) Server Sents XML data or JSON data to the XHR Callback function means Server Sends a Response back to the web Page
6) The Response is Read and Proper action like Page update, display data on browser is Perform.

* Properties of XMLHttpRequest Object.

1) onReadyStateChange :-
it is Called whenever ~~red~~ readystate attribute change. it must not be used with synchronous requests. (it Specify name of function / define function or event handler to Called automatically when readystate Property change)

2) readystate :-
represents the state of the request. it ranges from 0 to 4.
- 0 UNOPENED open() is not Called. (request not initialized)
- 1 OPENED open is Called but Send() is not Called. (Server Connection established)
- 2 HEADERS_RECEIVED Send() is Called, and headers and status are available (request recieve)
- 3 LOADING Downloading data; responseText holds the (Processing request) data
- 4 DONE the operation is Completed fully (request finished & response ready)

3) ~~response~~ reponse text :-
returns response as text. (it is a readonly)

4) response xml :-
returns response as xml.
(it is read only)

5) Status :- ~~The Status Property and status Text Property holds the~~

* Methods of XHR object.

1) Void open (method, URL) :-
Opens the request specifying get or Post method and url. (it is used to open and initialize an HTTP request for sending)

2) Void open (method, URL, async)
Same as above but specifies asynchronous or not.

3) Void open (method, URL, async, username, Password)
Same as above but specifies username and Password.

4) Void send()
Sends get request. (it used to send request to the server)

5) Void send (string)
Send Post request (~~string~~ data to be send in the request it is used when request is send to the server using post if the request method is get string parameter is ignore & the request string is set to null

6) Set Request Header (header, value)
it adds request headers

5] Status Property :-

- The Status Property & statusText Proped holds of the status of the XHR object.

ex: 200 , Ok
403 , 4
404 Page not found

6] SetRequestHeader (header , value) [method]

- it adds a header / value Paired to the http request header to be Send it must Call after open() method but before Calling Send() method.

- header = the name of the header who's value to be Set
- value = the value to the Set as the body of the header.

* Example Ajax.

Ex.-1 Ajax retriving the text file from the Server.

```
<head>
<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("POST", "ay.text", true);
      xhttp.send();
    }
</script>
</head>
<body>
    <h2> — 11 — <h2>
    <div id = "demo">
    <p> — 11 — </p>
    <button type = "button" onclick = "loadDoc()">
      Change Content </button>
    </div>
</body>
```

Ex. 2 fetch retriving the xml file data load coise from Server.

```
<body>
    <h2> ---- </h2>
    <p id="demo"></p>
<script>
    Const http = new xmlHttpRequest();
    xhttp.onload = function() {
     Const xmlDoc = this.responseXml;
     Const x = xmlDoc.getElementBYTagName("
       ARTIST");
     Let txt = " ";
     for (let i=0; i<x.length ;i++)
     {
       txt = txt + x[i].childNodes[0].
       nodeValue + "<br>";
     }
     document.getElementBYId ("demo").inner
      HTML = txt;
    }
    xhttp.open ("GET", "cd.xml", true);
    xhttp.Send();
</script>
 </body>
```

cd.xml

```
<?xml Version="1.0" encoding="UTF-8">
<CATALOG>
 <CD>
<TITLE> Enpire burlesque </TITLE>
```

<ARTIST> —"— </ARTIST>
<COUNTR> —"— </COUNTR>
<COMPANY>
<YEAR> 1987 </YEAR>
</CO>
</CATALOG>

9g.txt

<h1> AJAX </h1>
<P> —"— </P>
<P> —"— </P>