Unit 7:

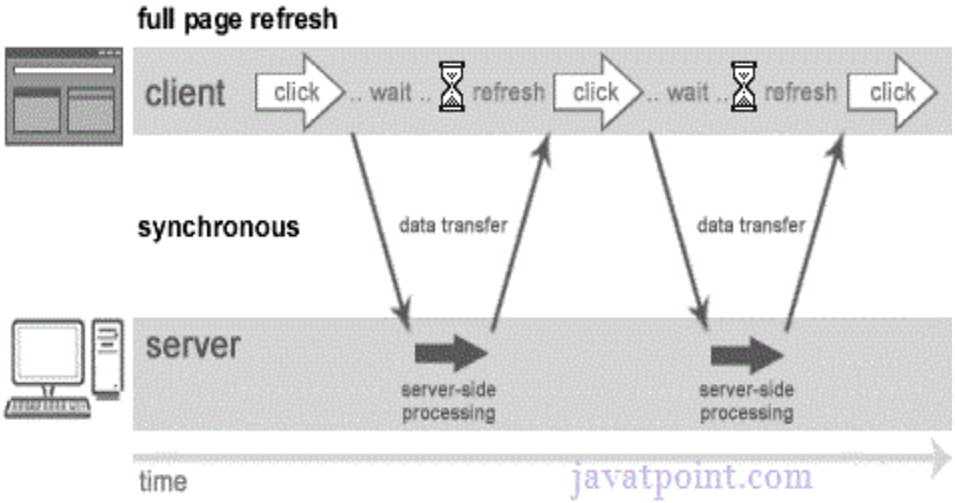
Advanced Web Programming concepts

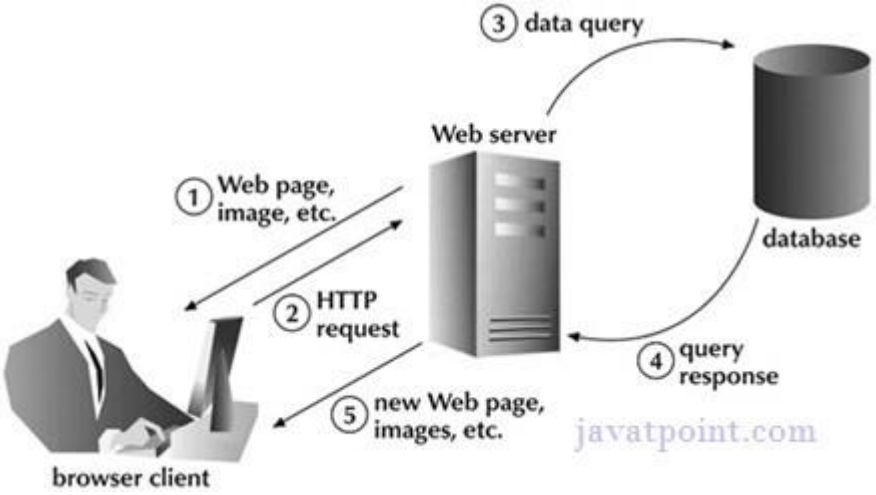
AJAX

- Before understanding AJAX, let's understand classic web application model and ajax web application model first.

Synchronous (Classic Web-Application Model)

- A synchronous request blocks the client until operation completes i.e. browser is unresponsive. In such case, javascript engine of the browser is blocked.



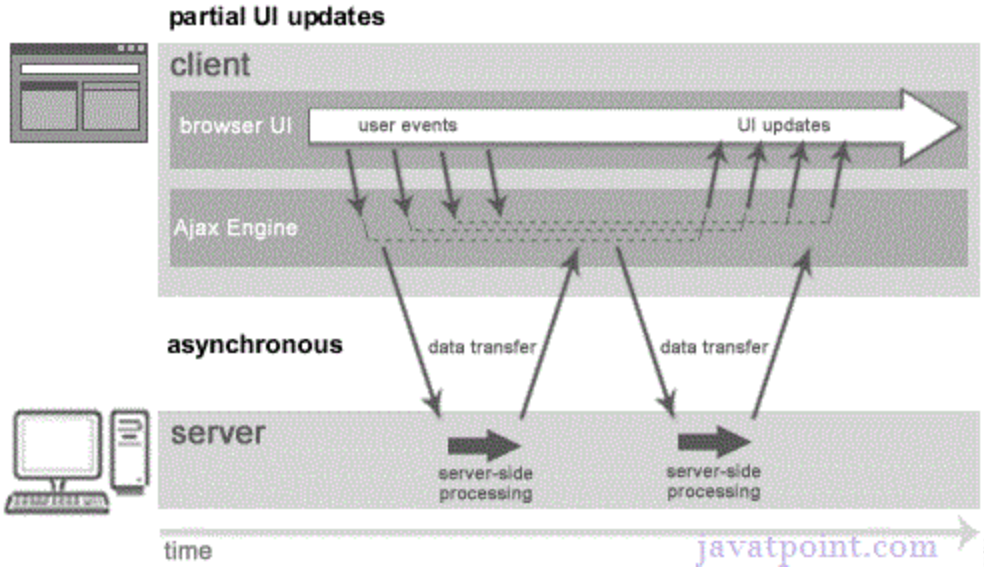


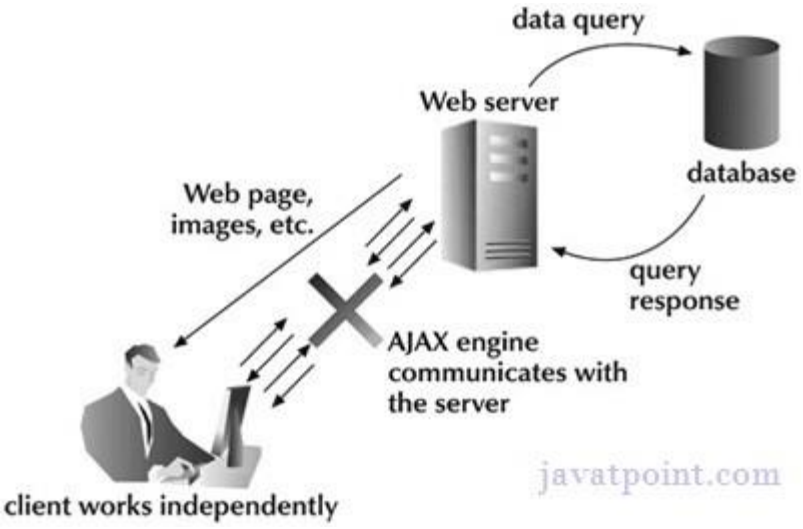
- As we can see in the above image, full page is refreshed at request time and user is blocked until request completes.

AJAX

Asynchronous (AJAX Web-Application Model)

- An asynchronous request doesn't block the client i.e. browser is responsive. At that time, user can perform another operations also. In such case, javascript engine of the browser is not blocked.





- As we can see in the above image, full page is not refreshed at request time and user gets response from the ajax engine.

AJAX

- AJAX is a web development technique for creating interactive web applications. If you know JavaScript, HTML, CSS, and XML, then you need to spend just one hour to start with AJAX.

Why to Learn Ajax?

- AJAX stands for **A**synchronous **J**avaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.
- Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.
- Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.

AJAX

- With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
- XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.
- AJAX is a web browser technology independent of web server software.
- A user can continue to use the application while the client program requests information from the server in the background.
- Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
- Data-driven as opposed to page-driven.

AJAX

- AJAX is the most viable Rich Internet Application (RIA) technology so far. It is getting tremendous industry momentum and several tool kit and frameworks are emerging. But at the same time, AJAX has browser incompatibility and it is supported by JavaScript, which is hard to maintain and debug.

AJAX is based on the following open standards –

- Browser-based presentation using HTML and Cascading Style Sheets (CSS).
- Data is stored in XML format and fetched from the server.
- Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
- JavaScript to make everything happen.

AJAX

- AJAX cannot work independently. It is used in combination with other technologies to create interactive webpages.

JavaScript

- Loosely typed scripting language,JavaScript function is called when an event occurs in a page.Glue for the whole AJAX operation.

DOM

- API for accessing and manipulating structured documents, Represents the structure of XML and HTML documents.

CSS

- Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript

XMLHttpRequest - JavaScript object that performs asynchronous interaction with the server.

AJAX

- Here is a list of some famous web applications that make use of AJAX.

Google Maps - A user can drag an entire map by using the mouse, rather than clicking on a button. https://maps.google.com/

Google Suggest - As you type, Google offers suggestions. Use the arrow keys to navigate the results. - https://www.google.com/webhp?complete=1&hl=en

Gmail - Gmail is a webmail built on the idea that emails can be more intuitive, efficient, and useful. https://gmail.com/

Yahoo Maps (new)

- Now it's even easier and more fun to get where you're going! https://maps.yahoo.com/

AJAX

- AJAX Example

```
<!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>
```

AJAX

- 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();
}
```

AJAX

- 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)

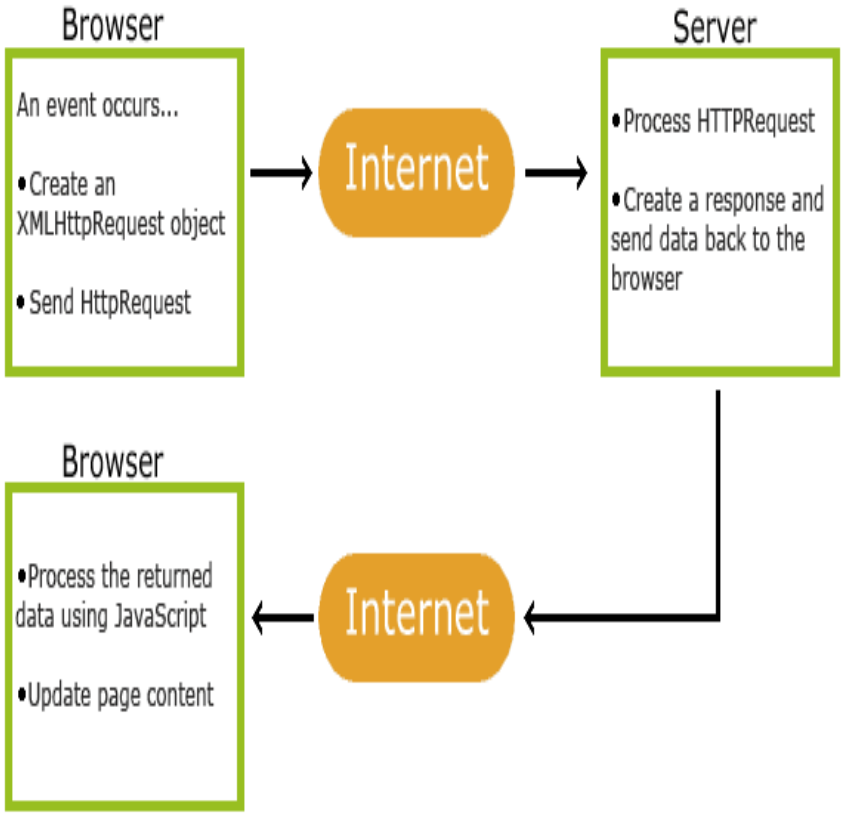
- AJAX is a misleading name. AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.

- 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.

AJAX

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



AJAX

AJAX - The XMLHttpRequest Object

- The keystone of AJAX is the XMLHttpRequest object.

The XMLHttpRequest Object

- All modern browsers support the XMLHttpRequest object.
- The XMLHttpRequest object can be used to exchange 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.

AJAX

AJAX - The XMLHttpRequest Object

Create an XMLHttpRequest Object

- All modern browsers (Chrome, Firefox, IE7+, Edge, Safari, Opera) have a built-in XMLHttpRequest object.
- Syntax for creating an XMLHttpRequest object:

  variable = new XMLHttpRequest(); - example

Access Across Domains

- For security reasons, modern browsers do not allow access across domains.
- This means that both the web page and the XML file it tries to load, must be located on the same server.
- If you want to use the example above on one of your own web pages, the XML files you load must be located on your own server.

AJAX

AJAX - The XMLHttpRequest Object

Create an XMLHttpRequest Object

Modern Browsers (Fetch API)

- Modern Browsers can use Fetch API instead of the XMLHttpRequest Object.
- The Fetch API interface allows web browser to make HTTP requests to web servers.
- If you use the XMLHttpRequest Object, Fetch can do the same in a simpler way.

Old Browsers (IE5 and IE6)

- Old versions of Internet Explorer (5/6) use an ActiveX object instead of the XMLHttpRequest object: variable = new ActiveXObject("Microsoft.XMLHTTP");
- To handle IE5 and IE6, check if the browser supports the XMLHttpRequest object, or else create an ActiveX object:

AJAX

AJAX - The XMLHttpRequest Object

XMLHttpRequest Object Methods

Method	Description
new XMLHttpRequest()	Creates a new XMLHttpRequest object
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(*method, url, async, user, psw*)	Specifies the request *method*: the request type GET or POST *url*: the file location *async*: true (asynchronous) or false (synchronous) *user*: optional user name *psw*: optional password
send()	Sends the request to the server Used for GET requests
send(*string*)	Sends the request to the server. Used for POST requests
setRequestHeader()	Adds a label/value pair to the header to be sent

AJAX

AJAX - The XMLHttpRequest Object

XMLHttpRequest Object Properties

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

AJAX

The XMLHttpRequest object is used to exchange data with a server.

Send a Request To a Server

- To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

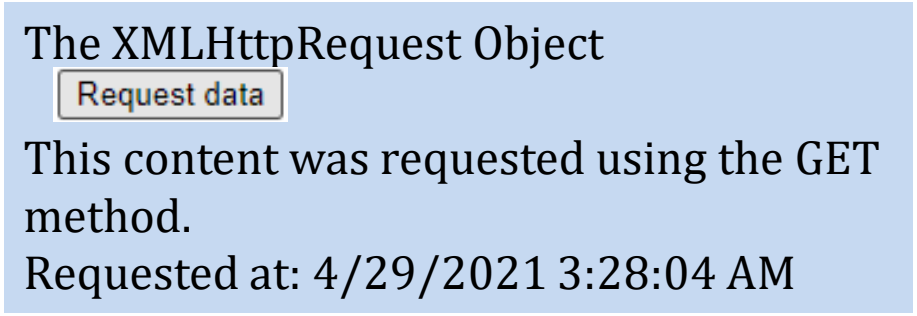
Method	Description
open(*method, url, async*)	Specifies the type of request *method*: the type of request: GET or POST *url*: the server (file) location *async*: true (asynchronous) or false (synchronous)
send()	Sends the request to the server (used for GET)
send(*string*)	Sends the request to the server (used for POST)

AJAX

The XMLHttpRequest object is used to exchange data with a server.

- Send a Request To a Server using GET Requests - A simple GET request:

```
<!DOCTYPE html>
<html>
<body><h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="loadDoc()">Request data</button>
<p id="demo"></p>
<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", "demo_get.asp", true);
 xhttp.send(); }
</script>  </body> </html>
```

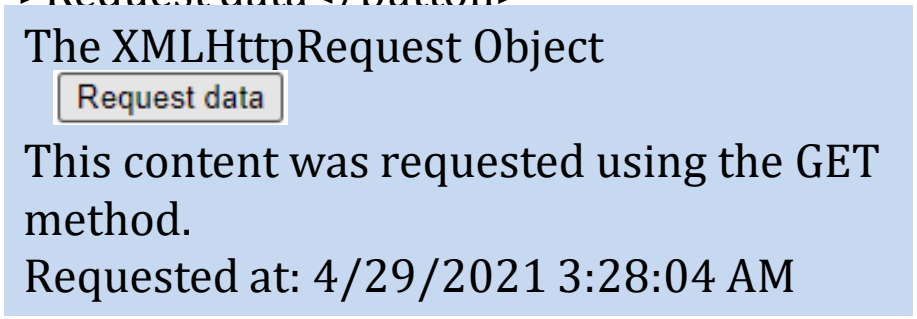


AJAX

The XMLHttpRequest object is used to exchange data with a server.

- Send a Request To a Server using POST Requests

```
<!DOCTYPE html>
<html>
<body><h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="loadDoc()">Request data</button>
<p id="demo"></p>
<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", "demo_get.asp", true);
 xhttp.send(); }
</script>  </body> </html>
```



AJAX

The url - A File On a Server

- The url parameter of the open() method, is an address to a file on a server:

  xhttp.open("GET", "ajax_test.asp", true);

- The file can be any kind of file, like .txt and .xml, or server scripting files like .asp and .php (which can perform actions on the server before sending the response back).

Asynchronous - True or False?

- Server requests should be sent asynchronously.
- The async parameter of the open() method should be set to true:
  xhttp.open("GET", "ajax_test.asp", true);
- By sending asynchronously, 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 after the response is ready

AJAX

The onreadystatechange Property

- With the XMLHttpRequest object you can define a function to be executed when the request receives an answer.
- The function is defined in the onreadystatechange property of the XMLHttpRequest object: - Example

Synchronous Request

- To execute a synchronous request, change the third parameter in the open() method to false: xhttp.open("GET", "ajax_info.txt", false);
- Sometimes async = false are used for quick testing. You will also find synchronous requests in older JavaScript code.
- Since the code will wait for server completion, there is no need for an onreadystatechange function - Example

AJAX

Synchronous Request

- Synchronous XMLHttpRequest (async = false) is not recommended because the JavaScript will stop executing until the server response is ready. If the server is busy or slow, the application will hang or stop.

- Synchronous XMLHttpRequest is in the process of being removed from the web standard, but this process can take many years.

- Modern developer tools are encouraged to warn about using synchronous requests and may throw an InvalidAccessError exception when it occurs.

AJAX

AJAX - Server Response

- The onreadystatechange Property
- The readyState property holds the status of the XMLHttpRequest.
- The onreadystatechange property defines a function to be executed when the readyState changes.
- The status property and the statusText property holds the status of the XMLHttpRequest object.

AJAX

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 403: "Forbidden" 404: "Page not found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

AJAX

AJAX - Server Response

- The onreadystatechange function is called every time the readyState changes.
- When readyState is 4 and status is 200, the response is ready - Example
- The onreadystatechange event is triggered four times (1-4), one time for each change in the readyState.