AJAX Tutorial

# Introduction to AJAX:

AJAX is an acronym for **Asynchronous JavaScript and XML**. It is a group of inter-related technologies like [JavaScript](https://www.javatpoint.com/javascript-tutorial)

, DOM, [XML](https://www.javatpoint.com/xml-tutorial)

, [HTML](https://www.javatpoint.com/html-tutorial)

/[XHTML](https://www.javatpoint.com/xhtml-tutorial)

, [CSS](https://www.javatpoint.com/css-tutorial)

, [XMLHttpRequest](https://www.javatpoint.com/understanding-xmlhttprequest)

etc.

AJAX allows you to send and receive data asynchronously without reloading the web page. So it is fast.

AJAX allows you to send only important information to the server not the entire page. So only valuable data from the client side is routed to the server side. It makes your application interactive and faster.

### Where it is used?

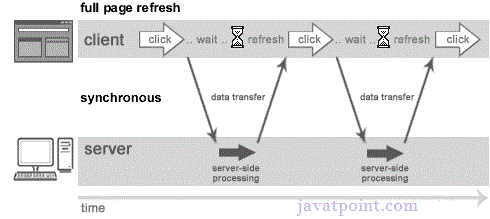
There are too many web applications running on the web that are using ajax technology like **gmail**, **facebook**,**twitter**,**google map**, **youtube** etc.

# Understanding Synchronous vs Asynchronous

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

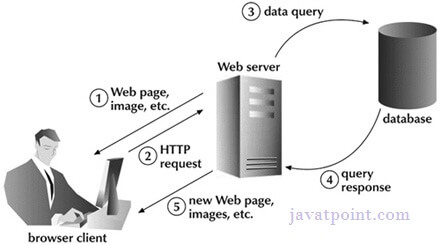
### Synchronous (Classic Web-Application Model)

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



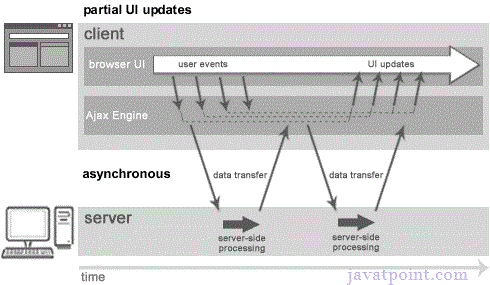
As you can see in the above image, full page is refreshed at request time and user is blocked until request completes.

Let's understand it another way.



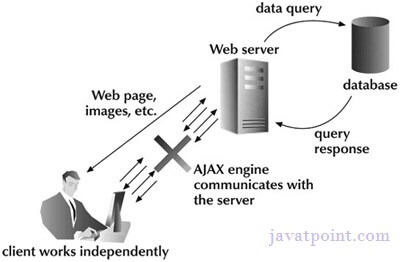
## Asynchronous (AJAX Web-Application Model)

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

Let's try to understand asynchronous communication by the image given below.



#### Note: every blocking operation is not synchronous and every unblocking operation is not asynchronous.

## AJAX Technologies

As describe earlier, ajax is not a technology but group of inter-related technologies. AJAX

technologies includes:

* HTML
* XHTML
* CSS
* DOM
* XML
* or JSON
* XMLHttpRequest
* JavaScript

# Understanding XMLHttpRequest

An object of XMLHttpRequest is used for asynchronous communication between client and server.

It performs following operations:

1. Sends data from the client in the background
2. Receives the data from the server
3. Updates the webpage without reloading it.

## Properties of XMLHttpRequest object

The common properties of XMLHttpRequest object are as follows:

|  |  |
| --- | --- |
| **Property** | **Description** |
| onReadyStateChange | It is called whenever readystate attribute changes. It must not be used with synchronous requests. |
| readyState | represents the state of the request. It ranges from 0 to 4.  **0** UNOPENED open() is not called.  **1** OPENED open is called but send() is not called.  **2** HEADERS\_RECEIVED send() is called, and headers and status are available.  **3** LOADING Downloading data; responseText holds the data.  **4** DONE The operation is completed fully. |
| reponseText | returns response as text. |
| responseXML | returns response as XML |

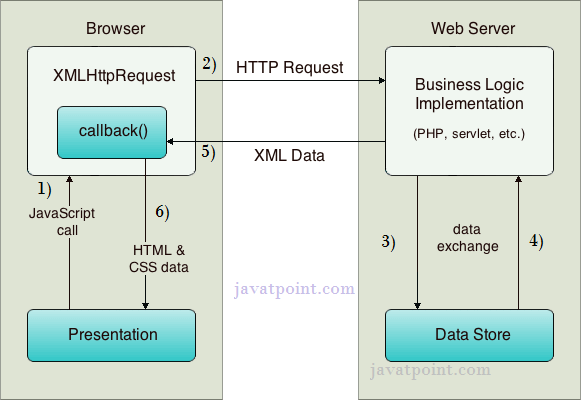
## Methods of XMLHttpRequest object

The important methods of XMLHttpRequest object are as follows:

|  |  |
| --- | --- |
| **Method** | **Description** |
| void open(method, URL) | opens the request specifying get or post method and url. |
| void open(method, URL, async) | same as above but specifies asynchronous or not. |
| void open(method, URL, async, username, password) | same as above but specifies username and password. |
| void send() | sends get request. |
| void send(string) | send post request. |
| setRequestHeader(header,value) | it adds request headers. |

# 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.

# AJAX JSON Example

We can get JSON data by AJAX code. AJAX provides facility to get response asynchronously. It doesn't reload the page and saves bandwidth.

### AJAX JSON Example

Let's see a simple example of getting JSON data using AJAX code.

1. ***<html>***
2. ***<head>***
3. ***<meta content="text/html; charset=utf-8">***
4. ***<title>AJAX JSON by Javatpoint</title>***
5. ***<script type="application/javascript">***
6. ***function load()***
7. ***{***
8. ***var url = "http://date.jsontest.com/";//use any url that have json data***
9. ***var request;***
11. ***if(window.XMLHttpRequest){***
12. ***request=new XMLHttpRequest();//for Chrome, mozilla etc***
13. ***}***
14. ***else if(window.ActiveXObject){***
15. ***request=new ActiveXObject("Microsoft.XMLHTTP");//for IE only***
16. ***}***
17. ***request.onreadystatechange  = function(){***
18. ***if (request.readyState == 4  )***
19. ***{***
20. ***var jsonObj = JSON.parse(request.responseText);//JSON.parse() returns JSON object***
21. ***document.getElementById("date").innerHTML =  jsonObj.date;***
22. ***document.getElementById("time").innerHTML = jsonObj.time;***
23. ***}***
24. ***}***
25. ***request.open("GET", url, true);***
26. ***request.send();***
27. ***}***
28. ***</script>***
29. ***</head>***
30. ***<body>***
32. ***Date: <span id="date"></span><br/>***
33. ***Time: <span id="time"></span><br/>***
35. ***<button type="button" onclick="load()">Load Information</button>***
36. ***</body>***
37. ***</html>***