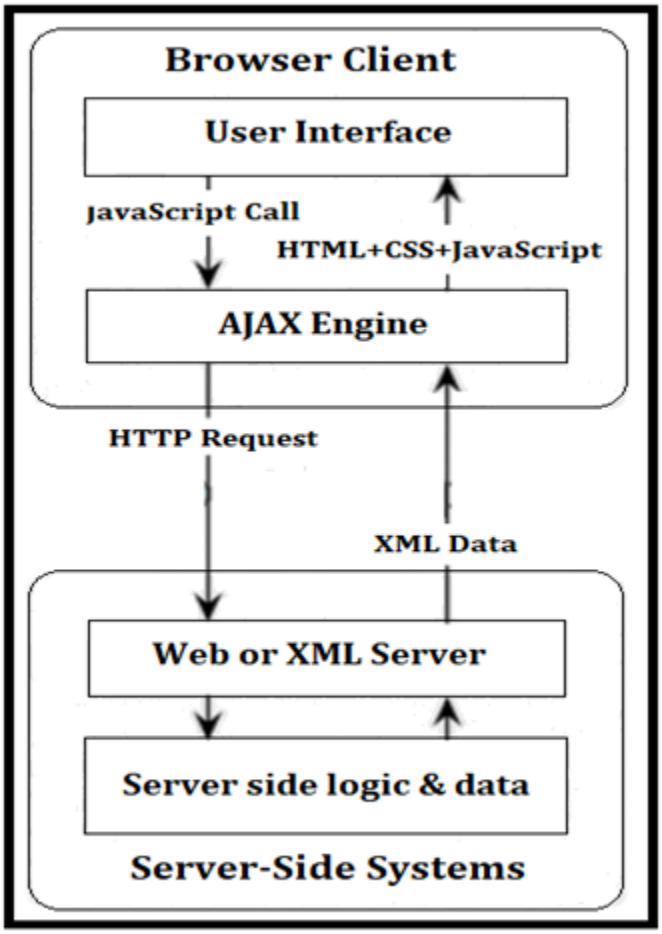
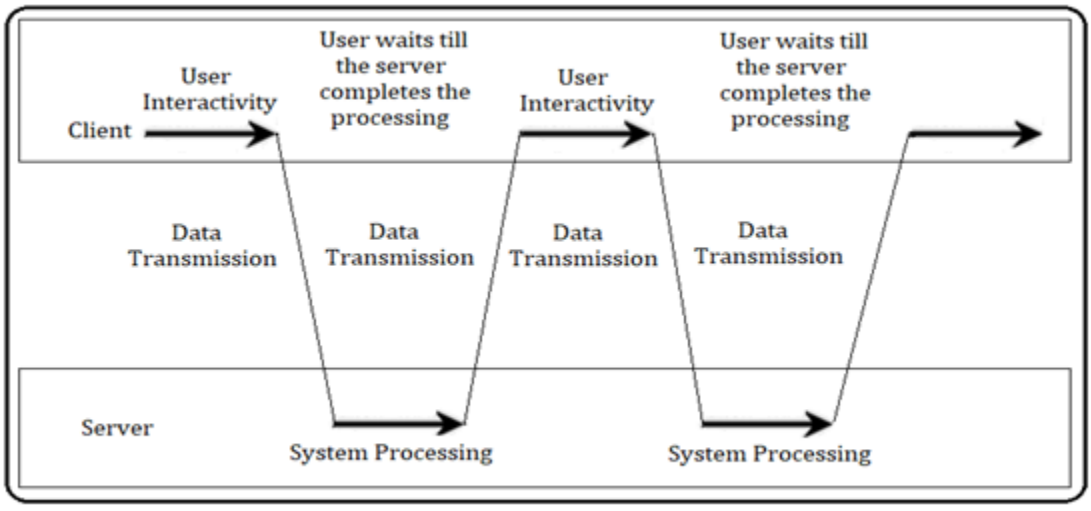
AJAX Web Application Model

- AJAX Wen application model uses JavaScript and XMLHttpRequest object for asynchronous data exchange. The JavaScript uses the XMLHttpRequest object to exchange data asynchronously over the client and server.
- AJAX Web application model resolve the major problem of synchronous request-response model of communication associated with classical Web application model, which keeps the user in waiting state and does not provide the best user experience.
- AJAX, a new approach to Web applications, which is based on several technologies that help in developing applications with better user experience. It uses JavaScript and XML as the main technology for developing interactive Web applications.



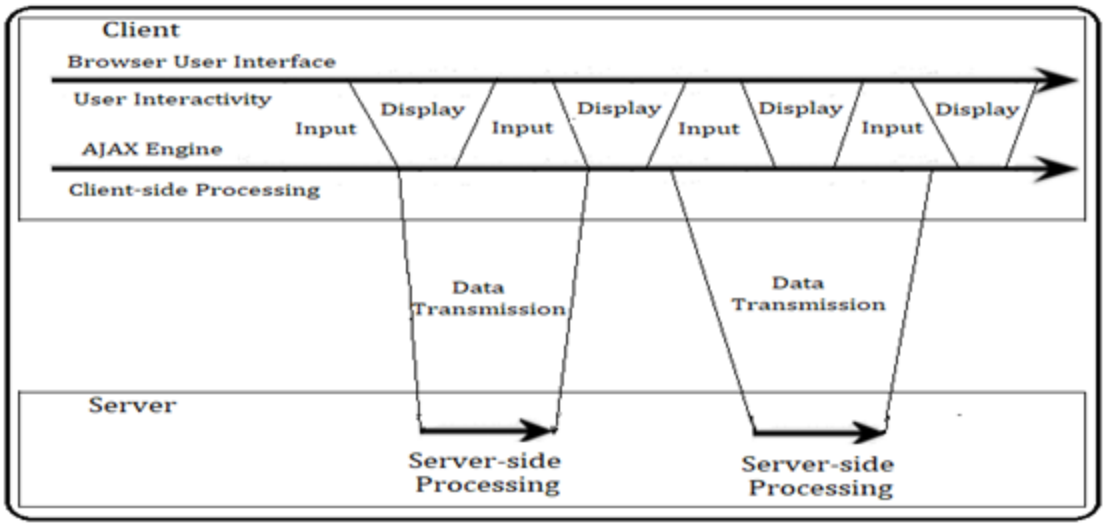
AJAX Web Application Model

- The AJAX application eliminates the start-stop-start-stop nature or the click, wait, and refresh criteria of the client-server interaction by introducing intermediary layer between the user and the Web server.
- Instead of loading the Web page at the beginning of the session, the browser loads the AJAX engine written in JavaScript.
- Every user action that normally would generate an HTTP request takes the form of a JavaScript call to the AJAX Engine.
- The server response comprises data and not the presentation, which implies that the data required by the client is provided by the server as the response, and presentation is implemented on that data with the help of markup language from Ajax engine.
- The JavaScript does not redraw all activities instead only updates the Web page dynamically.
- In JavaScript, it is possible to fill Web forms and click buttons even when the JavaScript has made a request to the Web server and the server is still working on the request in the background. When server completes its processing, code updates just the part of the page that has changed. This way client never has to wait around. That is the power of asynchronous requests.
- AJAX Engine between the client and the application, irrespective of the server, does asynchronous communication. This prevents the user from waiting for the server to complete its processing.
- The AJAX Engine takes care of displaying the UI and the interaction with the server on the user's behalf.
- In traditional Web applications, the synchronous mode of communication existed between the client and the server as shown in following figure:



Synchronous Mode of Communication

- Since AJAX is essentially used for a partial update and asynchronous communication, the AJAX model used for programming and it is not restricted for use with specific data exchange format, specific programming language, or the specific communication mechanism.

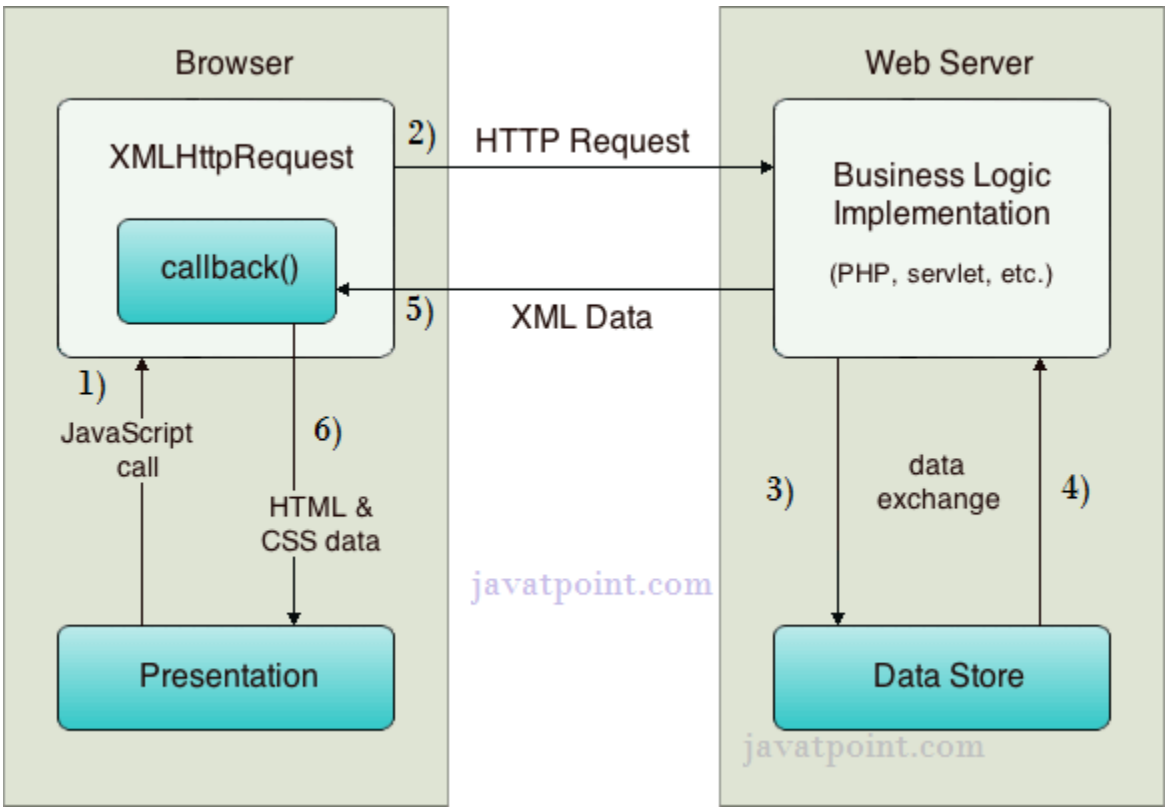


Asynchronous Mode of Communication

- Above diagram clarify that every user action generates an HTTP request that takes the form of a JavaScript to call the AJAX engine.
- Responses to the user actions do not involve the trip back to the server as in the classical Web application model. Instead, the AJAX Engine handles on its own, such as data validation, some navigation and editing data in memory.
- If the AJAX Engine needs something from the server, like retrieving new data or loading additional interface code, then the engine makes the asynchronous interaction with the server using JavaScript and XMLHttpRequest object for asynchronous data exchange.
- The engine's interaction with the server does not interrupt the user's interaction with the application. In this way, the asynchronous communication is done with the help of the AJAX engine.

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.

XHR Object's methods and properties

Method/Property	Description
`abort()`	Stops the current request.
`getAllResponseHeaders()`	Returns the response headers as a string.
`getResponseHeader("headerLabel")`	Returns a single response header as a string.
`open("method", "URL"[, asyncFlag[, "userName"[, "password"]]])`	Initializes the request parameters.
`send(content)`	Performs the HTTP request.
`setRequestHeader("label", "value")`	Sets a label/value pair to the request header.
`onreadystatechange`	Used to set the callback function that handles request state changes.
`readyState`	Returns the status of the request: 0 = uninitialized 1 = loading 2 = loaded 3 = interactive 4 = complete
`responseText`	Returns the server response as a string.
`responseXML`	Returns the server response as an XML document.
`Status`	Returns the status code of the request.
`statusText`	Returns the status message of the request.