AJAX

Asynchronous JavaScript And XML

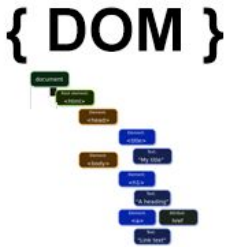
Why AJAX

- No refresh
- Can request external data from a server
- Can receive external data from a server
- Can Send Data to a Server

Typically when a user goes to a website the user has to wait for the server to respond with the data. This is not the case as with AJAX we have the option to load the data when the user is already on the page making the user really happy.

Meet the Players

JavaScript - front end scripting language provides functionality for web pages. Provides event listening, data handling and element manipulation



DOM - **Document Object Model a** cross-platform and language-independent application programming interface that treats an HTML, XHTML, or XML document as a tree structure wherein each node is an object representing a part of the document.

CSS - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language

Meet the Players

XMLHttpRequest : XMLHttpRequest (XHR) is an API in the form of an object whose methods transfer data between a web browser and a web server.

Server Side : uses any server side language to process request and output result. Can connect with databases, add logic, and any typically server functionality. Outputs http requests for AJAX

JavaScript Object Notation or JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types.

What is AJAX

- Not a programming language - group of technologies
- Combination of technology within the browser XHR object + JavaScript to connect the request data and the Document Object Model DOM
- Not only for XML but can be used and commonly used JSON and text data

AJAX is a set of Web development techniques using many Web technologies on the client side to create asynchronous Web applications.

With Ajax, Web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page

What is AJAX

By decoupling the data interchange layer from the presentation layer, Ajax allows for Web pages, and by extension Web applications, to change content dynamically without the need to reload the entire page. In practice, modern implementations commonly utilize JSON instead of XML due to the advantages of JSON being native to JavaScript

Ajax incorporates:

- standards-based presentation using XHTML and CSS;
- dynamic display and interaction using the Document Object Model;
- data interchange and manipulation using XML and XSLT;
- asynchronous data retrieval using XMLHttpRequest; and JavaScript binding everything together.

http://adaptivepath.org/ideas/ajax-new-approach-web-applications/ - first post about AJAX

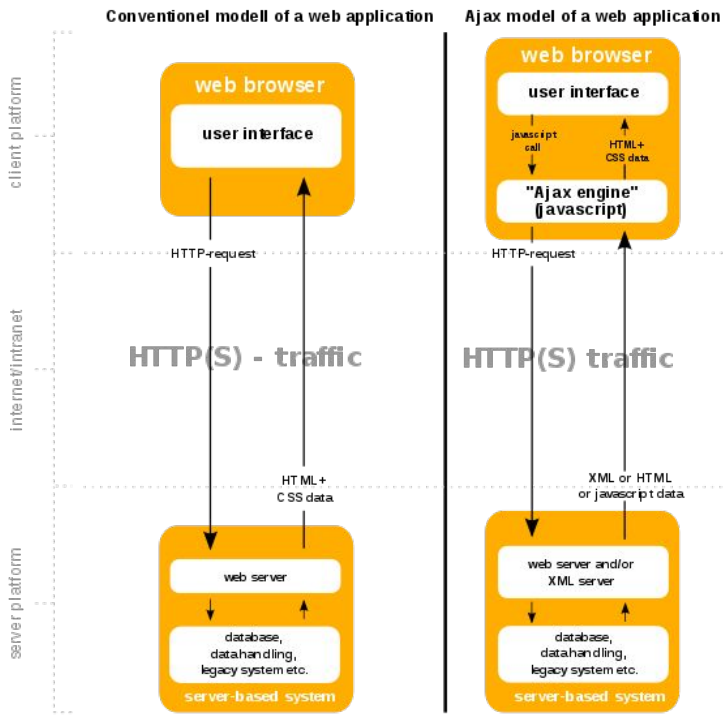
How AJAX works

JavaScript is used to send a request HTTP

JavaScript receives new information in the background. No page reload!!!!!

JavaScript can modify webpage to dynamically display – and allow the user to interact with — the new information

How it works


Conventionel modell of a web application
Ajax model of a web application
client platform
internet/intranet
server platform
web browser
user interface
HTTP-request
HTTP(S) - traffic
HTML+
CSS data
web server
database,
datahandling,
legacy system etc.
server-based system
web browser
user interface
javascript
call
HTML+
CSS data
"Ajax engine"
(javascript)
HTTP-request
HTTP(S) traffic
XML or HTML
or javascript data
web server and/or
XML server
database,
datahandling,
legacy system etc.
server-based system

Typical Example xHR

XHR has been around for a long time now and has very good cross-browser support

```
// This is the client-side script.

// Initialize the HTTP request.
var xhr = new XMLHttpRequest();
xhr.open('GET', 'send-ajax-data.php');

// Track the state changes of the request.
xhr.onreadystatechange = function () {
    var DONE = 4; // readyState 4 means the request is done.
    var OK = 200; // status 200 is a successful return.
    if (xhr.readyState === DONE) {
        if (xhr.status === OK) {
            console.log(xhr.responseText); // 'This is the output.'
        } else {
            console.log('Error: ' + xhr.status); // An error occurred during the request.
        }
    }
};

// Send the request to send-ajax-data.php
xhr.send(null);
```

Typical Example fetch

```
fetch('send-ajax-data.php').then(function(response) {
    return response.text();
}).then(function(data) {
    console.log(data);
}).catch(function(error) {
    console.log('Error: ' + error);
});
```

Fetch and Promises, on the other hand, are a more recent addition to the web platform, although they're supported well across the browser landscape, with the exception of Internet Explorer and Safari

Typical Example jQuery

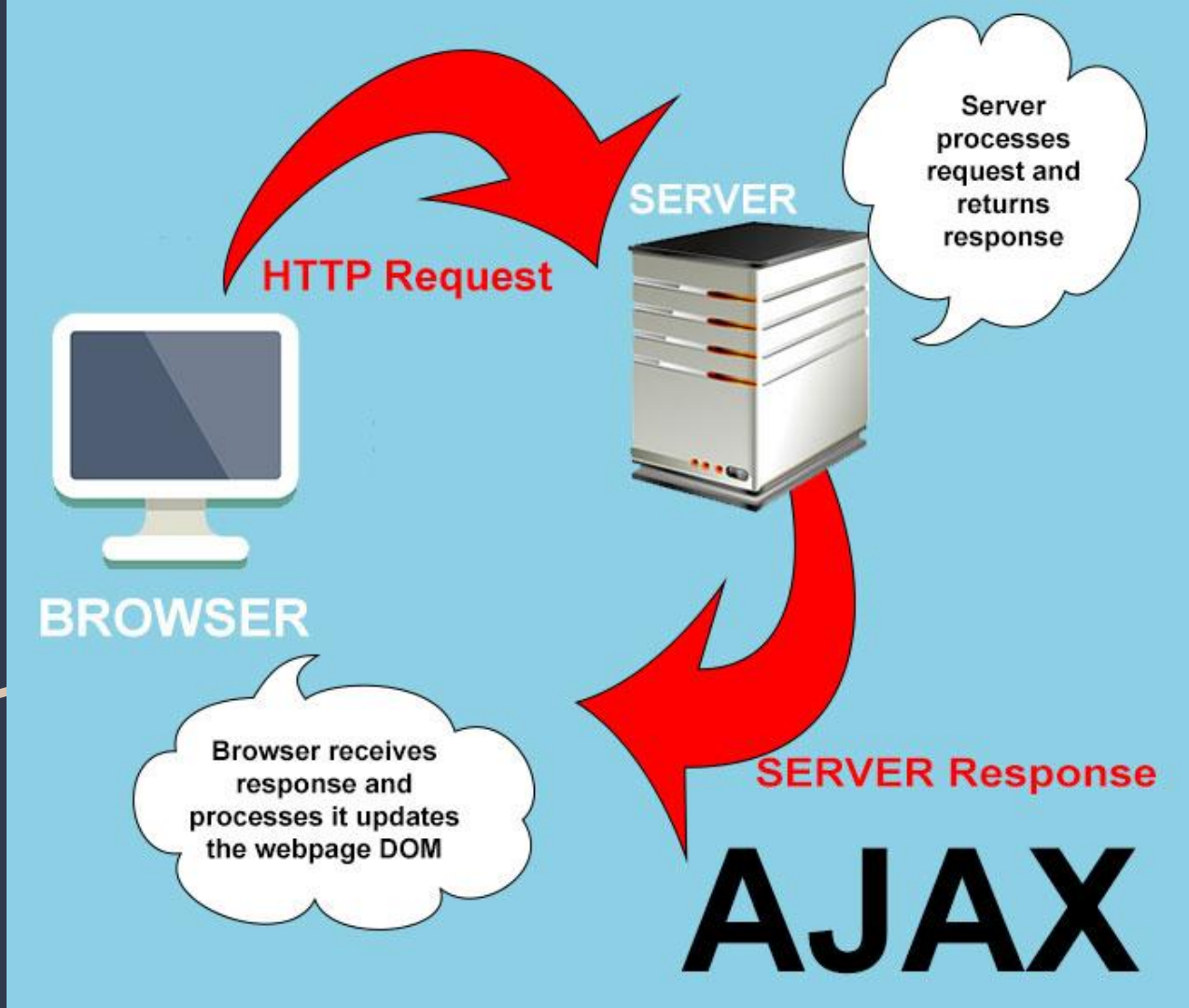
```
$.ajax({
    type: 'GET',
    url: 'send-ajax-data.php',
    dataType: "JSON", // data type expected from server
    success: function (data) {
        console.log(data);
    },
    error: function() {
        console.log('Error: ' + data);
    }
});
```

```
$.get('send-ajax-data.php').done(function(data) {
    console.log(data);
}).fail(function(data) {
    console.log('Error: ' + data);
});
```

More About AJAX

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data

AJAX


SERVER
Server processes request and returns response
HTTP Request
BROWSER
Browser receives response and processes it updates the webpage DOM
SERVER Response
AJAX

STEPS

1. Data is collected on the webpage. Usually a event trigger is used to start the process
2. XMLHttpRequest Object is created in the code
3. XHR sent to the server
4. Data is received by the server
5. Server issues response
6. Web page receives the response back
7. Updates to code using data retrieved from the server
8. Provides visual output to the user with new data