

University Mohamed Boudiaf - Msila
Faculty of Mathematics and Computer Science
Department of Computer Science
1st Year Master - RTIC



Web Technologies

AJAX

Mohamed Kamel

Academic Year 2023/2024

Table of Contents

- 1 Introduction
- 2 Examples
- 3 AJAX Operation
- 4 Syntax
- 5 XMLHttpRequest Object
- 6 AJAX & jQuery

Introduction

AJAX stands for **A**synchronous **J**avaScript **A**nd **X**ML

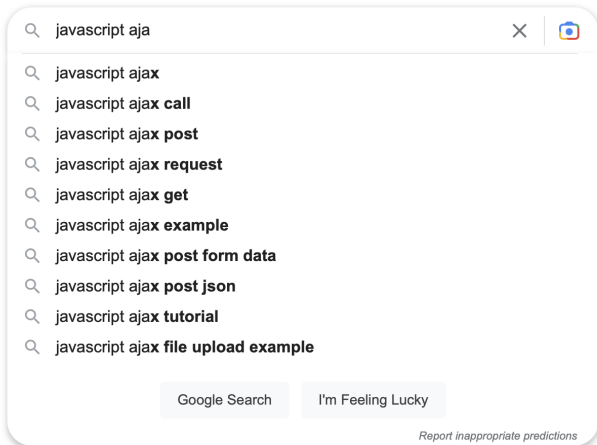
It is a group of interrelated web development techniques used on the client-side to create interactive web applications

With AJAX we can:

- Read data from a web server - after the page has loaded
- Update a web page without reloading the page
- Send data to a web server - in the background

Example

Auto suggest

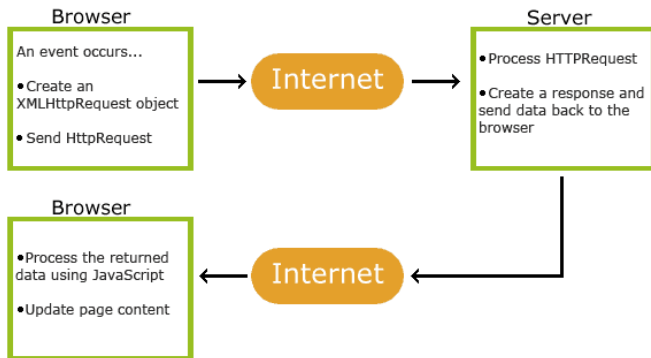


A screenshot of a search engine's auto-suggest dropdown. The search bar contains the text "javascript aja". Below the search bar, a list of suggestions is displayed, each preceded by a magnifying glass icon. The suggestions are:

- javascript ajax
- javascript ajax call
- javascript ajax post
- javascript ajax request
- javascript ajax get
- javascript ajax example
- javascript ajax post form data
- javascript ajax post json
- javascript ajax tutorial
- javascript ajax file upload example

At the bottom of the dropdown, there are two buttons: "Google Search" and "I'm Feeling Lucky". Below the buttons, there is a link that says "Report inappropriate predictions".

AJAX Operation



AJAX Operation

- An event occurs in a web page (load, click, ...)
- An **XMLHttpRequest** object is created by JavaScript
- The **XMLHttpRequest** object sends a request to a web server
- The server processes the request
- The server sends a response back to the web page
- The response is read by JavaScript
- Proper action (like page update) is performed by JavaScript

Syntax

example

```
<div id="demo">
<h2>AJAX Example</h2>
<button type="button" onclick="loadDoc()">Change Content</button>
</div>
```

XMLHttpRequest example

```
const xhttp = new XMLHttpRequest();
xhttp.onload = function() {
  document.getElementById("demo").innerHTML = this.responseText;
}
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

Methods

All modern browsers support the XMLHttpRequest object

XMLHttpRequest Object

```
1 new XMLHttpRequest();
```

Callback Function

```
1 xhttp.onload = function() {  
2   // What to do when the response is ready  
3 }
```

Request

```
1 xhttp.open("GET", "url");  
2 xhttp.send();
```


Methods

- **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

Properties

- **onload**, Defines a function to be called when the request is received (loaded)
- **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"
- **statusText**, Returns the status-text (e.g. "OK" or "Not Found")

AJAX & jQuery

jQuery provides many methods to deal with AJAX operations. We will only see **\$.ajax** method

\$.ajax syntax

```
1 $.ajax({name:value, name:value, ... })
```

Syntax

Request

```
1 $("button").click(function(){
2     $.ajax({
3         url: "demo_test.txt",
4         success: function(result){
5             $("#div1").html(result);
6         });
7     });
```

Syntax

Request

```
1  $("#button").click(function(){
2  $.ajax(
3  {
4    type: "POST",
5    url: "add.php",
6    data: $("#myform").serialize(),
7    success: function(result){
8      $("#result").html(result);
9    }
10 }
11 )
12     });
```

\$.ajax parameters

- **async**, A Boolean value indicating whether the request should be handled asynchronous or not. Default is true
- **beforeSend(xhr)**, A function to run before the request is sent
- **cache**, A Boolean value indicating whether the browser should cache the requested pages. Default is true
- **complete(xhr,status)**, A function to run when the request is finished (after success and error functions)
- **data**, Specifies data to be sent to the server
- **error(xhr,status,error)**, A function to run if the request fails.
- **ifModified**, A Boolean value specifying whether a request is only successful if the response has changed since the last request. Default is: false.
- **password**, Specifies a password to be used in an HTTP access authentication request.
- **success(result,status,xhr)**, A function to be run when the request succeeds
- **timeout**, The local timeout (in milliseconds) for the request
- **type**, Specifies the type of request. (GET or POST)
- **url**, Specifies the URL to send the request to. Default is the current page
- **username**, Specifies a username to be used in an HTTP access authentication request
- **xhr**, A function used for creating the XMLHttpRequest object

Questions ?