

# Casa discografica.

5B IT – A.S. 2018-2019

## Consegna

Esempio funzionante con dati inventati e capacità di integrarlo con le funzioni create dai colleghi.

## Relazione

AJAX è una tecnica di accesso ai web service da una pagine web che permette di aggiornare il contenuto di una pagina web senza aggiornare la pagina e di mandare dati in background al server web. AJAX sta per Asynchronous JavaScript And XML. Per trasportare i dati usa il testo in chiaro, JSON o l'XML. AJAX utilizza una XMLHttpRequest object per comunicare una richiesta al server che risponde mandando i dati che vengono processati usando il javascript che aggiorna gli elementi della pagina.

## XMLHttpRequest object

È un oggetto supportato da tutti I browser moderni che permette di inviare e ricevere informazioni dal e al server in background permettendo di creare una pagina dinamica.

```
var xhttp = new XMLHttpRequest();
```

L'oggetto viene creato come un qualsiasi oggetto in javascript. Ometto la creazione dell'oggetto in browser più vecchi per risparmiare tempo. Nel caso sevisse si prega di cercarlo autonomamente nelle fonti.

Ora riporto i metodi e le proprietà dell'oggetto:

## 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
	Specifies the request
open( <i>method</i> , <i>url</i> , <i>async</i> , <i>user</i> , <i>psw</i> )	<i>method</i> : the request type GET or POST <i>url</i> : the file location <i>async</i> : true (asynchronous) or false (synchronous) <i>user</i> : optional user name <i>psw</i> : optional password

<code>send()</code>	Sends the request to the server Used for GET requests
<code>send(string)</code>	Sends the request to the server. Used for POST requests
<code>setRequestHeader()</code>	Adds a label/value pair to the header to be sent

## XMLHttpRequest Object Properties

Property	Description
<code>onreadystatechange</code>	Defines a function to be called when the <code>readyState</code> property changes
<code>readyState</code>	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
<code>responseText</code>	Returns the response data as a string
<code>responseXML</code>	Returns the response data as XML data
<code>status</code>	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the <a href="#">Http Messages Reference</a>
<code>statusText</code>	Returns the status-text (e.g. "OK" or "Not Found")

## Mandare una richiesta ad un server

Per mandare una richiesta bisogna usare due metodi: `open` e `send`.

Il metodo `open` (vedi tabelle sopra) serve per creare la richiesta e il metodo `send` (vedi tabelle sopra) serve per inviare la richiesta.

Il metodo `open` chiede il metodo, l'url e se è una richiesta `asynchronous` o `synchronous`. I metodi che si possono scegliere sono GET o POST. GET è più facile e veloce da usare, ma meno sicuro. Per i nostri scopi il GET è consigliato.

```
xhttp.open("GET", "...", true);
xhttp.send();
```

Questi sono i comandi per mandare una richiesta. Al posto dei tre punti va inserito l'url della risorsa richiesta. Il `true` indica che la richiesta è `asynchronous` cioè che non è bloccante quindi per fare in modo di ricevere la richiesta bisogna sfruttare la proprietà `onreadystatechange`.

## Onreadystatechange

È una delle proprietà in cui puoi definire una funzione da eseguire quando il server risponde e il `readyState` (un'altra proprietà) è sul valore che definisce la riuscita dell'azione. La funzione è quasi sempre simile:

```
xhttp.onreadystatechange = function() //dichiarazione della funzione
{
    if (this.readyState == 4 && this.status == 200)
        //se la richiesta è stata approvata (this.status) e è andata a buon fine
        //(readyState) esegue i comandi
        {
            document.getElementById(". . .").innerHTML = this.responseText;
            //cambia l'elemento desiderato (da mettere al posto dei tre punti)
            //con la risposta
        }
};
```

in seguito a questa definizione bisogna inserire la open e la send:

```
xhttp.onreadystatechange = function()
{
    if (this.readyState == 4 && this.status == 200)
        {
            document.getElementById(". . .").innerHTML = this.responseText;
        }
};
//al posto dei tre punti bisogna mettere il file php che gestisce il database
xhttp.open("GET", ". . .?q="+str, true);

xhttp.send();
L'intera funzione è così:
```

```
function . . .() //al posto dei tre punti va il nome della funzione
{
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200)
            {
                document.getElementById(". . .").innerHTML=this.responseText;
            }
    };
    xhttp.open("GET", ". . .?q="+str, true);
    xhttp.send();
}
```

“this.responseText” contiene la risposta in testo.

Per gli utilizzi che ne facciamo usare il testo in chiaro basta quindi non approfondire come utilizzare l'xml.

## Lato server

Fino ad adesso ho spiegato come funziona AJAX da lato client. Ora parlerò del lato server.

Per usare AJAX con PHP bisogna usare il metodo intval:

```
$q = intval($_GET['q']);
```

Questo metodo mette nella variabile \$q il parametro q dell'url preso dall'a richiesta. La pagina PHP cambia a seconda dell'utilizzo quindi non si possono fare esempi generici.

Per inviare i dati all'utente basta fare delle echo e AJAX farà il resto.

## Esempio:

### File php:

```
<?php
//sta in attesa di essere chiamato e quando succede memorizza il parametro
//q
$q = intval($_GET['q']);

//file con gli accessi
include "accessi.php";

//si connette e fa dei controlli
$conn = new mysqli($servername,$username,$password,$dbname);
if($conn->connect_error)
    die("Errore di connessione: ".$conn->connect_error);

//crea la query
$sql="SELECT u.nome,c.contatto FROM utente AS u, contatto AS c WHERE c.id_utente
= u.id AND u.id = '". $q ."'";

//esegue la query
$result = $conn->query($sql);

//crea la tabella
if($result->num_rows>0)
{
    echo "<div class=\"table-responsive\">";

    echo "<table class=\"table table-hover table-bordered table-condensed \">
<tr>
<th>Nome</th>
<th>Contatto</th>
</tr>";
    while($row=$result->fetch_assoc())
    {
        echo "<tr>";
        echo "<td>" . $row['nome'] . "</td>";
        echo "<td>" . $row['contatto'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
    $conn->close();
?>
```

### File html:

```
<html>
<head>

//link per bootstrap
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

<!-- jQuery library -->
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></
script>

//funzione per connettersi al server
<script>
function showUser(str)
{
    if (str == "")
    {
        document.getElementById("txtHint").innerHTML = "";
        return;
    }
    else
    {
        if (window.XMLHttpRequest)
        {
            // code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp = new XMLHttpRequest();
        }
        else
        {
            // code for IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }

        xmlhttp.onreadystatechange = function()
        {
            if (this.readyState == 4 && this.status == 200)
            {
                document.getElementById("txtHint").innerHTML =
this.responseText;
            }
        };
        xmlhttp.open("GET","service.php?q="+str,true);
        xmlhttp.send();
    }
}
</script>
</head>
<body>

//per la selezione dei dati
<form>
<select name="users" onchange="showUser(this.value)">
    <option value="">Select a person:</option>
    <option value="1">Gianni</option>
    <option value="2">Carlo</option>
</select>
</form>
<br>
<div id="txtHint"><b>Person info will be listed here...</b></div>

</body>
```

</html>

Risultato:

Gianni

Nome	Contatto
gianni	333 345 3452

Fonti:

<https://www.w3schools.com>