

## Writing of a code that accept within html code Ajax with Perl

Html code ajax.htm

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
3
4 <head>
5 <title>Comment lancer un script Perl depuis JavaScript</title>
6 <meta http-equiv="content-type"
content="text/html; charset=utf-8" />
7 <meta name="generator" content="Geany 0.14" />
8 <script type="text/javascript" src="CallServer.js"></script>
9 </head>
10 <body>
11 <input id="yourtext" type="text" size="50" value="Allez y
écrivez un texte et appuyez sur le bouton..." />
12 <input id="callPerl" type="submit" value="Vas-y script Perl !"
onclick="call_server.launch();" />
13 <br />
14 <br />
15 <div id="resultat"></div>
16 </body>
17 </html>
```

CallServer.js

```
//Constructeur de la "classe" Callserver

function CallServer ()
{
this.xhr_object;
this.server_response;

this.createXMLHttpRequest = createXMLHttpRequest;
this.sendDataToServer = sendDataToServer;
this.displayAnswer = displayAnswer;
this.launch = launch;
}

//On crée l'objet XMLHttpRequest

function createXMLHttpRequest()
{
this.xhr_object = null;

if(window.XMLHttpRequest)
{
this.xhr_object = new XMLHttpRequest();
}
}
```

```

else if(window.ActiveXObject)
{
this.xhr_object = new ActiveXObject("Microsoft.XMLHTTP");
}
else
{
alert("Your browser doesn't provide XMLHttpRequest
functionality");
return;
}
}

//On envoie des données au serveur et on reçoit la réponse en mode
synchrone dans this.server_response

function sendDataToServer (data_to_send)
{
var xhr_object = this.xhr_object;

xhr_object.open("POST", "cgi-bin/Server.cgi", false);

xhr_object.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");

xhr_object.send(data_to_send);

if(xhr_object.readyState == 4)
{
this.server_response = xhr_object.responseText;
}
}

//On injecte la réponse du serveur dans la div nommée resultat

function displayAnswer ()
{
document.getElementById("resultat").innerHTML =
this.server_response;
}

//Exécution du Javascript

function launch ()
{
this.sendDataToServer(document.getElementById("yourtext").value);

this.displayAnswer();
}

//Création de l'objet call_server

var call_server = new CallServer();
call_server.createXMLHttpRequest();

```

cgi-bin/Server.cgi

```
#!/usr/bin/perl

use strict;
use warnings;
use CGI;

#On crée notre objet qui initialise le CGI et crée une couche
d'abstraction envoi/réception pour communiquer avec notre client

package Ajax;

#Constructeur de la classe Ajax

sub new
{
my($classe) = shift;

my $self = {};

bless($self, $classe);

$self->{CGI} = CGI->new();

print $self->{CGI}->header('text/html;charset=UTF-
8;q=0.9,*/*;q=0.8');

return $self;
}

#Méthode qui nous permet de recevoir les données du client et les
renvoit sous forme de tableau

sub getDataFromClient
{
my ($self) = shift;

return $self->{CGI}->param("keywords");
}

#Méthode qui envoie des données au client

sub sendResultToClient
{
my ($self, $data_to_send) = @_;

print $data_to_send;
}

#Notre incroyable méthode qui transforme du texte

sub change
{
```

```
my ($self) = shift;

my $result = "Votre texte transformé : ";

my @texte = $self->getDataFromClient();

foreach(@texte)
{
    $result = $result.$_. "Rajout inutile juste pour montrer que ça fonctionne";
}

$self->sendResultToClient($result);
}

1;

## Début du script perl ##

my $ajax = Ajax->new();

$ajax->change();
```

This code is not yet analyzed but will be pretty soon.

This is website source:

<http://www.commentcamarche.net/faq/26976-ajax-communication-entre-un-script-perl-et-javascript>