



## TP1- UTILISATION D'AJAX AVEC TRAITEMENT SERVEUR ET JQUERY

### 1 INITIATION MIS EN ŒUVRE OBJET AJAX

AJAX (Asynchronous JavaScript and XML) n'est pas une technologie, c'est le résultat d'un ensemble de technologies du web qui fonctionnent ensemble pour arriver à un objectif simple : rafraîchir une partie de page web sans recharger la page complète.

#### AJAX... C'EST SIMPLE !

Exemple mettre à jour le fil de commentaire sans que la page soit rechargée entièrement quand vous laissez des « post »(s) sur un site. Le but du jeu est donc qu'au moment de la soumission du formulaire, une logique se mette en route côté jQuery, qui fera que le commentaire sera ajouté en base de données, et qu'il soit automatiquement ajouté au fil, le tout sans recharger la page !

C'est totalement possible... et c'est même très répandu. Si vous êtes fan des réseaux sociaux, comme Facebook, vous avez forcément été confronté à l'un de ces fils de commentaire dans votre vie de geek.

#### JQUERY NE SUFFIT PAS !

Nous avons parlé d'un fil de commentaire qui se recharge en AJAX plus haut, et nous avons même évoqué l'idée d'une base de données. Vous devriez donc comprendre que jQuery est aussi capable d'interagir avec ce qu'il se passe côté serveur !

Si c'est ce que vous pensez, sachez que ce n'est que partiellement vrai. Pour interagir avec une base de données, jQuery va devoir appeler des scripts côté serveur, qui eux sont capables de le faire. Vous commencez à comprendre ? **AJAX repose sur ce fondement même, la communication asynchrone d'un langage côté client, avec un langage côté serveur.** Vous devinez la suite, pour mettre en place un appel AJAX sur son site, jQuery ne va plus nous suffir. Voilà ce dont on va avoir besoin :

Un langage côté client : nous utiliserons bien sûr JavaScript, avec jQuery.

Un langage côté serveur : nous utiliserons ici le PHP.

Le script PHP appelé fais son travail : envoi de mail, insertion en base de données... et surtout, il renvoie un résultat que jQuery va intercepter. Ce résultat, c'est ce que jQuery utilisera pour mettre à jour la page

CRM	Auteur	TP	Version	Date MAJ	Page 1/10
	FCHATELOT	Les applications web client	2	11/09/2019	



## UN APPEL AJAX SANS LE VOCABULAIRE TECHNIQUE

Pour une première approche nous utiliserons les fonctions Ajax de jQuery pour éviter une surcharge de vocabulaire technique propre aux objets Ajax XMLHttpRequest surnommé objet « XHR » surtout pour la compatibilité avec les anciens navigateurs.

En effet :

### INSTANCIER XHR : UN CALVAIRE

Instancier un objet XHR peut devenir difficile... car il faut prendre en compte le problème de compatibilité entre les navigateurs. Les navigateurs Internet Explorer antérieurs à la version 7 utilisaient une implémentation différente de XHR : ActiveX, développé par Microsoft. Il va donc falloir prendre en compte ces navigateurs pour que notre appel AJAX soit mis en œuvre sur ces ceux-ci.

Exemple : `var xhr = null;`

```
if(window.XMLHttpRequest || window.ActiveXObject){
    if(window.ActiveXObject){
        try{
            xhr = new ActiveXObject("Msxml2.XMLHTTP");
        }catch(e){
            xhr = new ActiveXObject("Microsoft.XMLHTTP");
        }
    }else{
        xhr = new XMLHttpRequest();
    }
}else{
    alert("Votre navigateur ne supporte pas l'objet XMLHttpRequest...");
    return;
}
```

Ça fait peur n'est-ce pas ?

### XMLHTTPREQUEST AVEC JQUERY

XMLHttpRequest devient nettement plus facile à instancier avec jQuery. Une seule ligne de code, cela va aller très vite !

```
$(document).ready(function(){
    /*
```

CRM	Auteur	TP	Version	Date MAJ	Page 2/10
	FCHATELOT	Les applications web client	2	11/09/2019	



```
* Utilisons $.ajax pour créer une instance de XMLHttpRequest
*/

$.ajax();

});
```

Intéressons-nous de plus près, à la fonction la plus importante en JQuery pour manipuler l'objet XHR et donc faire des requêtes Ajax...

```
$.ajax(options);
```

JQuery permet d'émettre des requêtes Ajax vers un serveur, en effectuant l'appel Ajax via la méthode \$.ajax(options), et ceci indépendamment du navigateur utilisé.

Options de la methode \$.ajax(options)

url : Correspond à l'url du programme sur le serveur qui traitera la requête. Cette option est obligatoire, sinon aucun traitement sur le serveur ne peut avoir lieu...

Data : Objet ou chaine de caractères qui sera transmis au serveur. Si on utilise une chaine de caractères, elle doit être de la forme name1=value1 & name2=value2..., chaque name étant le nom d'un paramètre et value la valeur correspondante, encodée en UTF-8.

Si on utilise un objet, jQuery encode lui-même en UTF-8 chacune des valeurs et transmet au serveur une chaine de la forme name1=value1& name2=value2...

Type : Mode de transmission des paramètres indiqués dans l'option data précédente. 2 valeurs sont possibles : « GET » (valeur par défaut) ou « POST »

Complete : Méthode de la forme : complete(xhr, textStatus) appelé à la fin de la requête Ajax, que celle-ci ait réussie ou non.

Le paramètre textStatus contient « success » si la requête a réussi sinon un message d'erreur, comme « error », « timeout » ou « parseerror ».

Dans le cas où la requête a réussi, le paramètre « xhr » donne accès à l'objet « XMLHttpRequest » contenant la réponse serveur (dans xhr.responseText).

Username : Nom d'utilisateur à indiquer si le serveur nécessite une autorisation.

Password : Mot de passe associé à l'option username.

CRM	Auteur	TP	Version	Date MAJ	Page 3/10
	FCHATELOT	Les applications web client	2	11/09/2019	



## MISE EN APPLICATION

Vous réaliserez l'exemple suivant d'utilisation Ajax. On saisit un nom dans un champs text, puis on envoie à l'aide d'un bouton, le nom saisie vers le serveur qui le transforme en majuscule et le retourne vers le navigateur, qui affiche le nom transformé.

Le code de la partie serveur se trouve dans le fichier suivant :

```
< ?
```

```
$nom=$_REQUEST['nom'];
```

```
echo strtoupper($nom); //
```

```
?>
```

A vous de faire la page client .html contenant le code Javascript qui effectue le traitement souhaité dans l'énoncé.

## 2 CHARGEMENT DE DONNEES AU FORMAT JSON

Le format de données **JSON** (JAVASCRIPT OBJECT NOTATION) constitue le standard actuel pour les échanges de données sur le Web, notamment avec AJAX.

Il s'agit d'une syntaxe pour décrire des informations structurées sous une forme proche des objets JavaScript. Voici un exemple de document JSON qui décrit deux voitures (source).

```
{
  "voitures" : [
    { "modèle" : "Peugeot",
      "couleur" : "bleu",
      "immatriculation" : 2008,
      "révisions" : [ 2012, 2014 ]
    },
    { "modèle" : "Citroën",
      "couleur" : "blanc",
      "immatriculation" : 1999,

```

CRM	Auteur	TP	Version	Date MAJ	Page 4/10
	FCHATELOT	Les applications web client	2	11/09/2019	



```
    "révisions" : [ 2003, 2005, 2007, 2009, 2011, 2013 ]
  }
]
}
```

Vous allez vous rendre compte que ce format ne présente pas beaucoup de défauts, principalement parce qu'il reste vraiment très simple à lire, comprendre et utiliser.

## LES PLUS

Format compréhensible par tous (humain et machine). Aucun apprentissage n'est requis puisque la syntaxe n'utilise que quelques marques de ponctuations (nous le verrons plus tard).

Ne dépend d'aucun langage (format d'échange de données ouvert). Comme ce format est très ouvert, il est pris en charge par de nombreux langages : JavaScript, PHP, Perl, Python, Ruby, Java,...

Permet de stocker des données de différents types : chaînes de caractères (y compris des images en base64), nombres, tableaux (array), objets, booléens (true, false), la valeur null.

Sa structure en arborescence et sa syntaxe simple lui permet de rester très "léger" et efficace.

## LES MOINS

Comme pour tout moyen de stockage de données, il faudra prendre des mesures de sécurité pour garantir la sécurité des données sensibles.

Le fait que la syntaxe soit rudimentaire peut être un inconvénient dans certains cas. Par exemple, contrairement à XML, il n'y a pas d'identification précise des données (sous forme de balise par exemple), la structure doit donc être connue avant utilisation.

## LA SYNTAXE

Je vous parlais précédemment de quelques signes de ponctuations qui permettent de structurer les données dans un fichier .json. En voici la liste (vous ne serez normalement pas dépayés puisqu'ils sont utilisés exactement de la même manière dans d'autres langages, comme JavaScript par exemple) :

- { . . . } : les **accolades** définissent un objet.
- "language" : "Java" : Les **guillemets** (double-quotes) et les **double-points** définissent un couple clé/valeur (on parle de membre).
- [ . . . ] : Les **crochets** définissent un tableau (ou array en anglais).

CRM	Auteur	TP	Version	Date MAJ	Page 5/10
	FCHATELOT	Les applications web client	2	11/09/2019	



- `{"id":1, "language":"json", "author":"Douglas Crockford"}` : Les **virgules** permettent de séparer les membres d'un tableau ou, comme ici, d'un objet . A noter : pas de virgule pour le dernier membre d'un objet, sinon, il ne sera pas valide et vous aurez des erreurs lors de l'analyse du fichier.

C'est tout ? Ben oui, c'est tout ! Une dernière précision cependant, et elle n'est pas anodine : tout doit être encodé en utf-8 (ou utf-16 ou utf-32).

Il est temps de découvrir un exemple de fichier .json basique :

```
{
  "titre_album":"Abacab",
  "groupe":"Genesis",
  "annee":1981,
  "genre":"Rock"
}
```

Suffisamment simple pour qu'il y ai très peu d'explication...

Un fichier .json contient un objet (`{...}`) qui, ici, contient 4 membres identifiés par leurs paires clé/valeur. Les valeurs possibles sont :

- Une chaîne de caractères : `"titre":"Le format json", "description":"Le format <strong>simple</strong> et <strong>léger</strong>,"`  
`"contenu":"<p>L'avantage de json est son incroyable simplicité d'apprentissage et de mise en oeuvre. C'est le \"Petit Poucet\" de l'échange de données.</p>"`
- Un nombre (pas de guillemets requis dans ce cas) : `"pi":3.14, "g":9.81, "v_son":340`
- Un tableau : `[...]`
- Un objet : `{...}`
- D'autres valeurs possibles : un booléen (true ou false), null, rien (`"crmnaute":true, "autreinternaute":null, "auteur":""`). Attention, ces valeurs doivent être écrites en minuscule.

Sachez qu'il est possible d'ajouter autant d'espaces (au sens large du terme, c'est à dire que les retours à la ligne et les tabulations sont également considérés comme des espaces) que l'on veut. Le fichier ci-dessus pourra donc également s'écrire de la manière suivante, par exemple :

```
{
  "titre_album"
    : "Abacab",
  "groupe"
    : "Genesis",
```

CRM	Auteur	TP	Version	Date MAJ	Page 6/10
	FCHATELOT	Les applications web client	2	11/09/2019	



```
"annee"
  :1981,
"genre"
  : "Rock"
}
```

Voyons à présent un exemple plus complexe contenant des objets et des tableaux :

```
{
  "fruits": [
    { "kiwis": 3,
      "mangues": 4,
      "pommes": null
    },
    { "panier": true }
  ],
  "legumes": {
    "patates": "amandine",
    "poireaux": false
  },
  "viandes": ["poisson", "poulet", "boeuf"]
}
```

Dans cet exemple nous pouvons observer qu'il y a 3 membres (fruits, legumes et viandes). fruits est constitué d'un seul membre qui est un tableau de 2 objets : le premier objet contient 3 membres et le second un seul. legumes est défini par un objet constitué par 2 membres. viandes, quant à lui, est défini par un tableau de 3 éléments.

Cet exemple démontre l'extraordinaire imbrication (illimitée) de ce type de format...

Pour vérifier votre syntaxe json lors de la création du fichier, vous pouvez utiliser le site « parser » suivant

<http://jsonviewer.stack.hu/>

Webographie :

Pour de la doc. détaillée sur le format Json :

<http://json.org/>

<http://www.tutorialspoint.com/json/index.htm>

Site en Français.

<https://www.xul.fr/ajax-format-json.php>

CRM	Auteur	TP	Version	Date MAJ	Page 7/10
	FCHATELOT	Les applications web client	2	11/09/2019	



## MISE EN APPLICATION AJAX JQUERY ET BOOTSTRAP

### ETAPE 1 – COMPOSANT BOOTSTRAP

A l'aide du code ci-dessous et du Framework CSS Bootstrap vous réaliserez le diaporama de photo ci-dessous en mettant en œuvre le composant « Carrousel » de Bootstrap avec les flèches de défilement.

----- Code html avec les « class » Bootstrap -----

```
<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">

<div class="carousel-inner">

<div class="carousel-item active">



</div>

<div class="carousel-item">



</div>

<div class="carousel-item">



</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">
```

CRM	Auteur	TP	Version	Date MAJ	Page 8/10
	FCHATELOT	Les applications web client	2	11/09/2019	





<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

-----fin de code -----

Le résultat doit ressembler au visuel ci-dessous...Les photos libres de droits vous sont fournis dans le dossier nommé « photos\_volcans » ...



CRM	Auteur	TP	Version	Date MAJ	Page 9/10
	FCHATELOT	Les applications web client	2	11/09/2019	



## ETAPE 2 INTEGRATION DE CONTENUS DYNAMIQUES AJAX AVEC JQUERY

A l'aide du fichier document word fourni qui contient les titres et numéro d'image. Vous devez créer un document Json structuré qui contient une liste d'objet Json...Chaque objet aura 3 propriétés un id :numero d'image , un titre (le texte correspondant) , un « alt » le texte correspondant plus les mots «- volcans en indonésie » pour le référencement

Ensuite à l'aide de ce fichier Json que vous devez charger Dynamiquement en Ajax avec les fonctions jquery vous devez afficher le titre et le numéro de chaque image, et le contenu de l'attribut alt qui s'affichera en légende en bas de chaque image ...

## ETAPE 3 INTEGRATION DE CONTENUS DYNAMIQUES AJAX SANS JQUERY

Même opération que précédemment mais les balises `<div class="carousel-item">` sont générées dynamiquement en javascript en fonction du nombre d'image, le chargement de données Json également... Le framework JQuery ne sera pas utilisé dans cet exercice...

## ETAPE 4 AJAX AVEC JQUERY ET TRAITEMENT SERVEUR (PHP)

Vous devez faire en sorte que :

- Avec un mini-formulaire dans la page et un script serveur en php, on puisse déterminer le nombre d'images du diaporama avec une variable « min » et une variable « max ».

CRM	Auteur	TP	Version	Date MAJ	Page 10/10
	FCHATELOT	Les applications web client	2	11/09/2019	