

# TD1-2 : Ajax

## V4.1.1

---



Cette œuvre de Mickaël Martin Nevot est mise à disposition sous licence Creative Commons Attribution - Utilisation non commerciale - Partage dans les mêmes conditions.

Document en ligne : [www.mickael-martin-nevot.com](http://www.mickael-martin-nevot.com)

---

Travail : **binôme**

## 1 Généralités

Ce TD s'inscrit dans la continuité du TD1-1 : JavaScript et jQuery.

N'oubliez pas de faire des recherches sur le Web à chaque fois que cela est nécessaire en prenant soin de vérifier que les informations trouvées soient correctes.

Vous visualiserez systématiquement votre travail dans différents navigateurs Web.

## 2 Côté serveur (PHP)

Tout d'abord, créez un fichier `data-processing.php` de sorte qu'il accepte les `XMLHttpRequest` en modifiant l'en-tête de la réponse :

```
<?php
// Type de la réponse : HTML, encodage : UTF-8
header('Content-Type: text/html ; charset=utf-8');
// Anti-cache pour HTTP/1.1
header('Cache-Control: no-cache, private');
// Anti-cache pour HTTP/1.0
header('Pragma: no-cache');
?>
```

Puis, récupérez les données transférées (attention à bien utiliser la super globale `$_REQUEST`) :

```
< ?php
if (isset($_REQUEST['input_email'])) {
    $email = $_REQUEST['input_email'];
}
...
?>
```

Faites un traitement de votre choix en PHP (envoi d'un *e-mail*, écriture dans un fichier et/ou écriture dans une base de données), puis affichez un texte personnalisé à l'écran :

```
<?php
// Traitement PHP
...
```

```
    echo 'Bonjour, ' . $user . '<br/>Votre inscription a bien été enregistrée,
merci.';
?>
```

### 3 Côté client (HTML et JavaScript)

Dans le fichier `script.js`, lors de la soumission du formulaire, initialisez une variable contenant l'ensemble des valeurs des champs du formulaire au format URL :

```
let param = "email=my@e-mail.com&...";
```

Créez un objet `XMLHttpRequest` et rendez votre code compatible avec le plus grand nombre possible de navigateurs :

```
function createXHR() {
    // Mozilla, Safari, Opera, ...
    try { return new XMLHttpRequest(); } catch(e) {}
    // Internet Explorer
    try { return new ActiveXObject("Msxml2.XMLHTTP.6.0"); } catch (e) {}
    try { return new ActiveXObject("Msxml2.XMLHTTP.3.0"); } catch (e) {}
    try { return new ActiveXObject("Msxml2.XMLHTTP"); } catch (e) {}
    try { return new ActiveXObject("Microsoft.XMLHTTP"); } catch (e) {}
    alert("XMLHttpRequest non supporté");
    // Non supporté
    return null;
}
```

```
// Création de l'objet XMLHttpRequest
```

```
let XHRObj = createXHR();
```

Créez ensuite une requête Ajax asynchrone vers la page `data-processing.php` en utilisant la méthode HTTP POST et en modifiant l'en-tête de la requête en conséquence :

```
// Ouvre une connexion asynchrone avec le serveur en POST
XHRObj.open("post", "data-processing.php", true);
// Internet media type: application/x-www-form-urlencoded, text/xml, etc.
XHRObj.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
```

Un processus asynchrone, par définition, ne s'exécute pas de manière séquentielle. Il est donc nécessaire de prévoir une fonction de rappel qui sera effectivement appelée à la réponse du serveur :

```
XHRObj.onreadystatechange = callback;
```

```
function callback() {
    // 0 non init., 1 en chargement, 2 chargé, 3 en traitement, 4 terminé
    if (XHRObj.readyState == 4) {
        // 404 page non trouvée, 403 accès refusé, 200 requête réussie, etc.
        if (XHRObj.status == 200) {
            ...
        }
    }
}
```

Affichez le texte de confirmation d'inscription à l'endroit de votre choix dans la page Web du formulaire. Prévoyez une balise du genre `<span id="subscribe_confirm"></span>` dans la page HTML et utilisez la fonction `text()` de jQuery par exemple dans la fonction `callback()` ci-dessus pour modifier l'élément depuis `script.js` :

```
$("#subscribe_confirm").text(XHRObj.responseText);
```

Enfin, envoyez la requête au serveur :

```
XHRObj.send(param);
```

## 4 Ajax avec jQuery

Conservez le code côté serveur inchangé.

Remplacez l'ensemble du code Ajax côté client par son équivalent en jQuery (en conservant le traitement de la fonction `callback()`) :

```
function callback(res) {  
    ...  
}  
  
$.ajax({  
    type: "POST",  
    url: "data-processing.php",  
    data: "email= my@e-mail.com&...",  
    success: callback  
});
```

## 5 JSON

Modifiez le type de données renvoyées par le serveur :

```
$.ajax({  
    type: "POST",  
    url: "data-processing.php",  
    dataType: "json",  
    data: "email= my@e-mail.com&...",  
    success: callback  
});
```

Modifiez ensuite l'affichage fait par la page Web `data-processing.php` en utilisant le format JSON :

```
<?php  
    echo '{"msg": "Bonjour, ' . $user . '<br/>Votre inscription a bien été enreg  
istrée, merci."}';  
?>
```

Enfin, modifiez le traitement de la fonction `callback()` en récupérant le texte de confirmation d'inscription grâce à la méthode jQuery `parseJSON()` :

```
let subscribConfirmMsg = $.parseJSON(res);  
$("#subscribe_confirm").text(subscribConfirmMsg.msg);
```

Dorénavant, la soumission du formulaire HTML envoie une requête JavaScript grâce à jQuery au serveur à travers la page `data-processing.php` qui renverra sa réponse à la fonction de rappel JavaScript `callback()`. Enfin, un texte de confirmation d'inscription apparaît sur la page Web sans rafraîchissement de celle-ci.