

# AJAX

Ajax (Asynchronous Javascript and XML) est méthode qui permet de rafraîchir une partie de page Web sans recharger la page complète. Il permet aussi d'afficher dynamiquement des données obtenues en PHP (exemple des données issues d'une base de données).

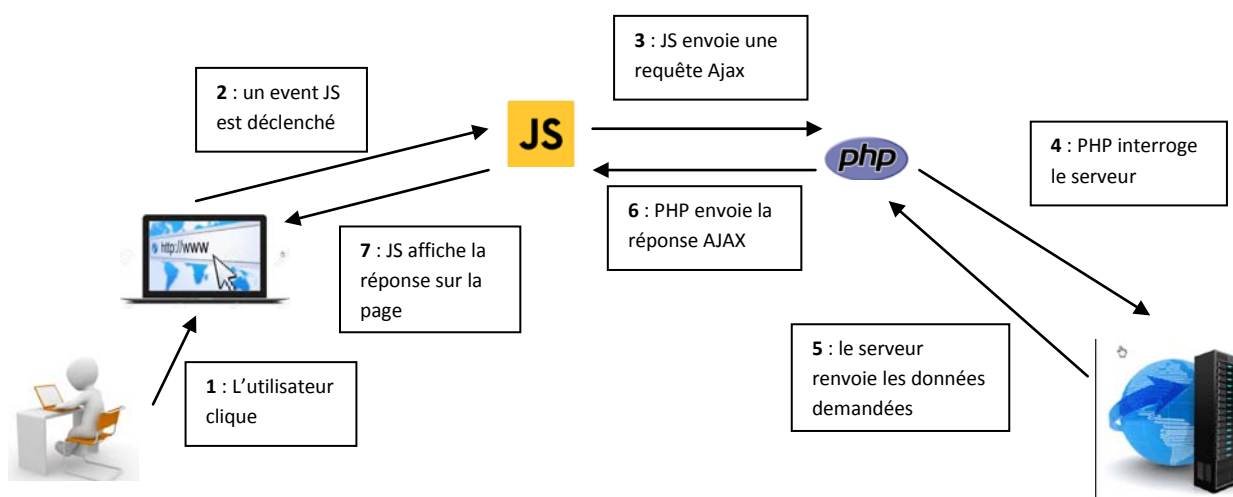
Un exemple : Facebook !

Lorsque vous créez une publication, il faut un système qui enregistre votre publication et en même temps l'affiche sur votre fil d'actualité. Ajax fait ça !

Pour mettre à jour une base de données, de quoi a-t-on besoin ? D'un langage côté serveur (donc php) et pour rafraîchir la page : d'un langage côté client. Ajax met en relation ces 2 langages !

## a. Comment fonctionne un appel Ajax :

- L'utilisateur crée un évènement (un clic ou autre)
- Le navigateur envoie sa requête ajax(en jquery) à un script PHP
- Le script PHP fait ce qu'il a à faire (insertion ou consultation de bdd, envoi de mail...)
- Le script PHP renvoie une réponse à l'appli JQuery
- JQuery traite la réponse et l'affiche



→ C'est ce que fait Facebook

## b. Renvoi d'un appel

Un appel AJAX va renvoyer différents types de données :

- Des données simples (chaînes de texte) → Dans le fichier PHP, nous aurons soit juste du texte, soit une fonction qui renvoie du texte

- Du html → Du code HTML dans le fichier PHP
- Du Json → Le fichier php renvoie un JSON avec la fonction `json_encode()`

### c. Comment fait-on ?

Il y a plusieurs façons d'écrire une requête Ajax, la syntaxe en JavaScript pur est assez complexe, la voici :

Code provenant d'openclassrooms :

<https://openclassrooms.com/fr/courses/245710-ajax-et-lechange-de-donnees-en-javascript/244798-lobjet-xmlhttprequest>

Création de l'objet XHR en fonction des navigateurs :

```
1 function getXMLHttpRequest() {
2   var xhr = null;
3
4   if (window.XMLHttpRequest || window.ActiveXObject) {
5     if (window.ActiveXObject) {
6       try {
7         xhr = new ActiveXObject("Msxml2.XMLHTTP");
8       } catch(e) {
9         xhr = new ActiveXObject("Microsoft.XMLHTTP");
10      }
11    } else {
12      xhr = new XMLHttpRequest();
13    }
14  } else {
15    alert("Votre navigateur ne supporte pas l'objet XMLHttpRequest...");
16    return null;
17  }
18
19  return xhr;
20 }
```

Envoie d'une requête Ajax :

```
1 var xhr = getXMLHttpRequest(); // Voyez la fonction getXMLHttpRequest() définie dans la partie
  précédente
2
3 xhr.open("GET", "handlingData.php", true);
4 xhr.send(null);
```

### C'est pourquoi on utilisera plus facilement la syntaxe en JQuery:

*Pour une appel avec une méthode GET(données en URL):*

`$.get('script.php','données',fonction de retour,'format des données reçues');`

*Pour une appel avec une méthode POST(données non visibles) :*

`$.post('script.php','données',fonction de retour,'format des données reçues');`

*Pour une appel avec une méthode GET json(données en URL reçues au format JSON):*

`$.getJSON('script.php','données',fonction de retour,'format des données reçues');`

Pour construire un appel, nous avons besoin :

- D'un fichier HTML
- D'un fichier JS
- D'un fichier PHP