# Création de requêtes Ajax avec jQuery

## 1. Objectifs

- Connaitre la méthode jQuery `$.ajax()`, ses options les plus courantes.
- Etre capable de créer des requêtes Ajax

## 2. Présentation

- **Ajax** pour **A**synchronous **J**avaScript **a**nd **X**ML, correspond à un groupe de méthodes et de moyens visant à permettre d'établir une communication asynchrone entre le navigateur et le serveur. Ajax permet d'effectuer des modifications parcellaires sur une page web, sans recharger l'ensemble de la page internet.
- **Ajax** est un ensemble de techniques de développement web sur le côté client qui permettent de créer des applications Web asynchrones.
- Les requêtes **AJAX** sont des méthodes permettant de charger un contenu personnalisé séparément du reste du document HTML, ce qui permet la mise en cache du document HTML complet, améliorant ainsi le temps de chargement final.
- L'objet **XMLHttpRequest** permet de répondre à ce besoin, cependant sont utilisation en JavaScript natif est relativement lourde à mettre en place. Jquery permet de réaliser des appels Ajax de manière beaucoup plus simple.

> *Ajax faisant appel à des technologies différentes, nativement une requête XMLHttpRequest peut se comporter de différente façon d'un navigateur à l'autre. Jquery est là pour pallier à ce problème et fournit un cadre pour réaliser des appels Ajax plus standardisés et surtout garantir qu'une requête Ajax s'exécute de la même façon sur chaque navigateur.*

- Pour simplifier la tache des développeurs , **jQuery** fournit plusieurs méthodes pour des fonctionnalités **AJAX**.
- **Jquery** propose un objet **ajax** auquel il suffit de passer les différents paramètres nécessaires pour effectuer une requête ajax : $.ajax.
- Avec **AJAX jQuery**, vous pouvez demander des données **TXT**, **HTML**, **XML** ou **JSON** d'un serveur distant en utilisant **HTTP Get** et **HTTP Post**.

> *Attention! la méthode `$.ajax()` fait bande à part dans le monde de **jQuery**. Contrairement aux autres méthodes de **jQuery**, elle ne crée pas un objet **jQuery** mais une requête **XMLHttpRequest**. On ne manipule plus le DOM, on gère une*

> *interface asynchrone avec le serveur Web. On est dans un monde nouveau, le Web 2.0.*

## 3. Les requêtes Ajax (ou XmlHttpRequest)

### 1. Paramètres de la requête Ajax

- Il existe de multiples façons d'instancier un appel via l'objet **XmlHttpRequest** .
- **Le minimum de paramètres** pour une utilisation judicieuse étant de deux(2) : **type** et **URL**.
  - **Type**: définit la méthode à employer **POST** ou **GET**.
  - **Url** : définit le script cible à appeler (ou la ressource)

> *Par convention il convient d'utiliser POST pour envoyer des données et GET pour en recevoir.*

- **Liste des paramètres couramment utilisés**

| Option | Description |
|---|---|
| **URL** | Adresse à laquelle la requête est envoyée |
| **type** | Le type de requête *GET* (par défaut) ou *$POST* |
| **data** | Les données à envoyer au serveur |
| **datatype** | Le type de données pouvant être transmises au serveur : *php*, *html*, *script*, *json* et *xml* |
| **dataFilter ( data , type )** | C'est une fonction utilisée pour gérer les données de réponse brut du XMLHttpRequest. |
| **success** | La fonction à appeler si la requête a abouti |
| **error** | La fonction à appeler si la requête n'a pas abouti |
| **timeout** | Le délai maximum en millisecondes de traitement de la demande. Passé ce délai, elle retourne le paramètre **error**. |
| **beforeSend** | Une fonction à exécuter avant d'envoyer la demande( la requête). Fonction appelée avant la requête. |

- **Exemple:**

apcpedagogie.com

2. **Envoyer la requête Ajax**

    - Il est possible d'envoyer une requête de deux façons :
        - **synchrone** : bloque le flux d'exécution de la page tant que le traitement n'est pas terminé.
        - **asynchrone**: ne bloque pas le flux d'exécution de la page même si le traitement traîne un peu.
        - Il suffit alors de rajouter le paramètre **async** et de lui assigner la valeur `true` ou `false`
        - Par défaut une requête **AJAX** est asynchrone donc **async:true**

3. **Les paramètres success, error et complete**

    - **success**
        - `jQuery` propose un paramètre pour gérer le retour de la fonction `$.ajax()` en cas de réussite. Le paramètre s'appelle **`success`**, et il ne sera exécuté que si l'appel AJAX a abouti. Ce paramètre prend en charge une fonction qui sera exécutée :

```
$("#identifiant").click(function(){
     $.ajax({
      url : 'source.php',
      type : 'GET',
      dataType : 'html', // On désire recevoir du HTML
      success : function(code_html, statut){
      // code_html contient le HTML renvoyé
      }
    });
});
```

    - L'argument **statut** est une chaîne de caractère automatiquement générée par jQuery pour donner le statut de la requête

4. **error**
    - Si l'appel AJAX a rencontré une erreur, le paramètre qui va être employé sera **error**. Le paramètre exécute une fonction si l'appel AJAX a échoué :

```
$("#identifiant").click(function(){
     $.ajax({
      url : 'source.php',
```

apcpedagogie.com

```
      type : 'GET',
      dataType : 'html', // On désire recevoir du HTML
      success : function(code_html, statut){
      // code_html contient le HTML renvoyé
      },
      error : function(resultat, statut, erreur){
      }
    });
});
```

5. La fonction exécutée par **error** prend en charges trois arguments :
   - Le **résultat**: est un objet XHR renvoyé par jQuery
   - Le **statut**: est le même que pour success,
   - L'**erreur**: est une exception, vous pouvez ici placer une chaîne de caractère à afficher à votre visiteur si l'appel AJAX n'a pas fonctionné.
6. **complete**
   - Le paramètre **complete**, est exécuté une fois l'appel AJAX effectué.
   - Le paramètre va prendre deux arguments, l'objet **resultat** dont nous avons parler plus haut ainsi qu'un **statut**:

```
$("#identifiant").click(function(){
     $.ajax({
      url : 'source.php',
      type : 'GET',
      dataType : 'html', // On désire recevoir du HTML
      success : function(code_html, statut){
      // code_html contient le HTML renvoyé
      },
      error : function(resultat, statut, erreur){
      },
      complete : function(resultat, statut){
      }
    });
});
```

## 4. Les fonctions de post-traitement (ou callback) et pre-traitement (ou preprocessing) avec Ajax Jquery

- **Javascript** peut effectuer des requêtes HTTP (de type GET ou POST), et récupérer le texte retourné par le serveur grâce à une fonction `callback`.

apcpedagogie.com

- Les requêtes **Ajax** incluent **une** fonction de `pré-tratement` (qui s'éxécutera avant la requête) et **trois** fonctions de `post traitement` (qui s'éxécuteront après l'exécution de la requête).
- **(pre-traitement)** `beforeSend`: Le fragment de javascript à éxécuter avant le lancement de la requête Ajax avec Jquery (permets d'afficher un loader, de dissimuler un formulaire, etc).
- **(post-traitement)** `done` : Le code qui sera opéré après l'exécution de la requête AJAX si aucune erreur n'a lieu (cad : si le code HTTP de retour est strictement égal à 200 > succès de la requête).
- **(post-traitement)** `fail`: Le javascript qui sera joué après l'appel de la requête AJAX si une erreur a lieu (cad : si le code HTTP de retour est différent de 200).
- **(post-traitement)** `always`: Le bout de code javascript à démarrer après done ou fail de la requête AJAX (permet de dissimuler un loader, d'afficher un formulaire, de rafraîchir des champs,etc).

## 5. Différence entre une requête client/serveur et une requête client avec AJAX

- Jesse James Garrett, celui qui a inventé le nom AJAX pour désigner l'ensemble de ces technologies mises en relation, établit une comparaison entre les schémas classiques du fonctionnement client/serveur, et la nouvelle approche (AJAX).
- Sur le schéma ci-dessous, nous voyons le concept introduit par Ajax : ce n'est plus le navigateur web qui fait les demandes serveur, mais le module JavaScript(Ajax engine dans le schéma).

  Cela change toute la manière de naviguer de l'utilisateur
  Source de l'image: *http://igm.univ-mlv.fr/~dr/XPOSE2005/ajax/presentation.php*

Sommaire du cours AJAX
La méthode $.ajax()

---

**Sommaire du cours AJAX**

Source : https://apcpedagogie.com/creation-de-requetes-ajax-avec-jquery · téléchargé le 04/11/2026 · © APC Pédagogie · apcpedagogie.com