

# Ajax : Vos premiers pas dans les nouvelles technologies

par [Nicolas PIED](#)

Date de publication : 30/07/2006

Dernière mise à jour : 30/07/2006

Ajax pour « Asynchronous JavaScript and XML » désigne en réalité un ensemble de technologies Web couramment utilisé, comme : XHTML, CSS, DOM et JavaScript, XML, XLS et XMLHttpRequest. Cette technologie, dont le terme a vu le jour en 2005, est aujourd'hui de plus en plus utilisée sur Internet notamment par Google au travers de sa page d'accueil personnalisée, sa boîte email GMail, etc. Le cœur d'Ajax repose principalement sur XMLHttpRequest, initialement développé par Microsoft, qui permet la lecture asynchrone de ressources présentes sur un serveur Web. Le but de cet article, est donc de se familiariser avec Ajax à travers un exemple concret. Celui-ci a pour but de récupérer la liste des DVD décrits au sein d'un fichier XML ; fichier qui peut par exemple, être le résultat d'un traitement ou d'une requête spéciale effectué par le serveur.

- 1 - Présentation du fichier XML
- 2 - Principales fonctions Ajax
- 3 - Obtenir le contenu du fichier XML
- 4 - Traiter et afficher l'arbre XML reçu
- 5 - Sécurité
- 7 - Conclusion

## 1 - Présentation du fichier XML

Le fichier XML a pour but de décrire sommairement les informations sur des DVD. La structure de celui-ci se présente comme ci-dessous :

```
<?xml version="1.0" encoding="utf-8"?>
<dvd>
  <item>
    <title>Princesse Mononoké</title>
    <director>Hayao Miyazaki</director>
    <price>14,99</price>
  </item>
  <item>
    <title>Mon voisin Totoro</title>
    <director>Hayao Miyazaki</director>
    <price>19,99</price>
  </item>
  <item>
    <title>Blood, The Last Vampire</title>
    <director>Hiroyuki Kitakubo</director>
    <price>11,99</price>
  </item>
</dvd>
```

Ce fichier XML dispose de plusieurs balises XML qui sont :

- **dvd** : racine du document XML
- **item** : représente la description sommaire d'un DVD
- **title** : titre de l'œuvre
- **director** : principal producteur du film
- **price** : prix en euros du DVD

Pour des raisons de simplicité, tous les fichiers présentés dans cet article, seront situés dans un même répertoire.

## 2 - Principales fonctions Ajax

Afin de pouvoir récupérer via JavaScript, le contenu du fichier XML présenté ci-dessus, il est nécessaire d'avoir recours à certaines fonctions ; fonctions qui ont été réalisées de telles façons à être compatibles avec les navigateurs standards (pour sûr Mozilla, Firefox et Internet Explorer 6.x).

- **getXMLHttpRequest** : cette fonction permet de retourner un objet XMLHttpRequest suivant les types de navigateurs rencontrés.
- **nodeCleaner et cleanXML** : ces fonctions permettent de fournir un arbre XML identique quelque soit le navigateur. En effet, les navigateurs basés sur le moteur Gecko et ceux sur Internet Explorer n'interprète pas les espaces et tabulations entre les balises de la même façon.

Ces fonctions sont à placer dans un fichier que l'on nommera « ajaxlib.js ».

### 3 - Obtenir le contenu du fichier XML

L'étape suivante consiste dans un premier temps à créer une instance de XMLHttpRequest. Celle-ci va permettre de récupérer le contenu du fichier XML contenant la liste des DVD qui sera ensuite traitée et affichée en utilisant le JavaScript.

Les 4 lignes de codes suivantes, vont effectuer toute la première partie, c'est-à-dire de fournir un arbre XML valide.

```
var xhr = getXMLHttpRequest();
/* La fonction handleHttpResponse est fournit plus loin dans cet article */
xhr.onreadystatechange = handleHttpResponse;
var url = "dvd.xml";
xhr.open("GET", url, true);
/* A préciser pour les requêtes de type POST
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
*/
xhr.send(null);
```

Il faut donc créer tout d'abord une instance de XMLHttpRequest grâce à l'appel de fonction getXMLHttpRequest. Suite, à cela, il faut définir la fonction qui va traiter l'arbre XML lorsqu'un évènement va survenir sur l'instance XMLHttpRequest.

La fonction indiquée ici, est « handleHttpResponse » et va permettre de traiter l'arbre XML et son affichage. La troisième et quatrième ligne permettent d'envoyer la requête au serveur afin que celui-ci retourne le fichier XML.

Notez qu'ici, la méthode GET est envoyée et qu'il faut donc dans ce cas spécifier « null » comme argument à la fonction « send » (car les arguments sont placés à la suite de l'URL). Si la méthode avait été de type POST alors les variables auraient été passées à la méthode « send » de la façon suivante :

```
xhr.send("var1=value&var2=othervalue");
```

## 4 - Traiter et afficher l'arbre XML reçu

Le traitement et l'affichage des données sont réalisés au sein de la fonction « handleHttpResponse ». Celle-ci doit effectuer son traitement seulement si le serveur est disponible (« xhr.readyState == 4 ») et si la lecture du fichier XML a bien été effectué (« xhr.status == 200 »). Si ces conditions sont remplies, alors on procède au nettoyage du fichier XML grâce à l'appel de la fonction cleanXML.

```
function handleHttpResponse() {
  if(xhr.readyState == 4 && xhr.status == 200) {
    /* xhr.responseXML permet d'obtenir le fichier XML
       xhr.responseText aurait retourné le fichier sous format texte */
    response = cleanXML(xhr.responseXML.documentElement);
    // ...
  }
}
```

Maintenant, l'affichage des informations sur les DVD nécessite l'emploi de l'API DOM JavaScript, comme indiqué ci-dessous :

```
/* Récupérer la liste des items donc la liste des DVD */
var items = response.getElementsByTagName("item");
var html = ''; /* Présentation HTML de la liste des DVD */
/* Nombre de DVD */
count = items.length;
for(i = 0; i < count; i++) { /* POUR CHAQUE item */
  html += '<div class="item">';
  html += '<h1>' + items[i].getElementsByTagName("title")[0].firstChild.nodeValue + '</h1>';
  html += '<div class="author">Réalisateur : ';
  html += items[i].getElementsByTagName("director")[0].firstChild.nodeValue + '</div>';
  html += '<div class="price">Prix : ';
  html += items[i].getElementsByTagName("price")[0].firstChild.nodeValue + ' #</div>';
  html += '</div>';
}
document.getElementById('content').innerHTML += html;
```

Ensuite tout ceci est affiché dans la balise DIV ayant l'identifiant « content ».

### Princesse Mononoké

Réalisateur : Hayao Miyazaki  
Prix : 14,99 €

### Mon voisin Totoro

Réalisateur : Hayao Miyazaki  
Prix : 19,99 €

### Blood, The Last Vampire

Réalisateur : Hiroyuki Kitakubo  
Prix : 11,99 €

## 5 - Sécurité

Pour des questions de sécurité, il faut savoir que la technologie Ajax ne permet pas d'appeler des ressources présentes sur un autre serveur. La seule méthode qui existe pour pallier à ce problème est de mettre en place un système de certificat.

## 7 - Conclusion

Cet article vous a présenté une première approche à l'utilisation de la nouvelle technologie Ajax. Cette dernière, bien que très intéressante, doit être envisagée après réflexion car le référencement par les moteurs de recherches est à ce jour loin d'être optimal.

[Télécharger l'article au format PDF](#)

 *Article sous licence Creative Commons (Paternité - Partage des Conditions Initiales à l'Identique)*