

# Effectuer une requête

Les méthodes utilisées pour toutes les requêtes HTTP sur le serveur sont `open()` et `send()` :

- `open()` configure la requête en spécifiant divers paramètres.
- `send()` effectue réellement la requête et accède ainsi au serveur.

Revenons plus en détail sur la méthode `open()`. Celle-ci se note :

```
open(méthode, url du fichier, mode)
```

- `méthode`. Spécifie le type de requête : "GET" ou "POST".
- `url du fichier`. Est l'adresse absolue ou relative du fichier sur le serveur. Depuis la prise en charge du XMLHttpRequest2, celui-ci ne doit plus appartenir au même domaine que le fichier d'appel mais peut être situé sur un autre domaine.
- `mode`. Valeur `true` pour une requête en mode asynchrone ou `false` pour une requête synchrone ou normale. En mode asynchrone, la requête s'effectue sans bloquer le processus de la page. Ce qui permet une navigation plus fluide et plus confortable pour l'utilisateur. Ce mode asynchrone est un élément crucial de la technique AJAX. En mode synchrone, tant que la réponse à la requête n'est pas parvenue, le processus du script ne se poursuit pas et bloque. Ce qui n'est jamais très convivial pour l'utilisateur.

Détaillons également la méthode `send()`.


Avec la méthode POST, il est possible de joindre des données à la requête. Elle se note alors `send(données)`. Ces données doivent alors être sous la forme d'une chaîne de caractères. En cas d'utilisation de la méthode POST, il faut absolument modifier le type MIME, sinon la requête sera ignorée par le serveur. On ajoutera ainsi après la ligne contenant la méthode `open()` : `xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");`

Avec la méthode GET, plus par habitude que par nécessité, on notera `send(null)` pour spécifier qu'il n'y a pas de données associées.

Le code devient :

```
if(window.XMLHttpRequest){
var xhr = new XMLHttpRequest();
}
else if(window.ActiveXObject){
var xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
else {
alert("Votre navigateur n'est pas compatible avec AJAX...");
}
xhr_object.open("GET", "test.txt", true);
xhr_object.send(null);
```

où `test.txt` est l'adresse relative du fichier.

 Cet ouvrage étant centré sur l'implémentation d'AJAX côté client, nous utiliserons surtout la méthode GET. La méthode POST envoie les données vers des programmes serveur PHP ou ASP.NET par exemple. Le traitement de ces données dépasse le cadre de cet ouvrage.

