

TP 12: Envoi de données avec AJAX

Objectifs:

- Envoyer des informations (ex: un tweet) à une API du Web
 - Effectuer une requête POST avec XMLHttpRequest
-

Effectuer une requête POST avec XMLHttpRequest

Dans la partie précédente, nous avons utilisé la classe XMLHttpRequest pour envoyer des requêtes de type HTTP GET. Pour rappel, ce type de requête permet de récupérer des informations depuis un serveur.

Dans cette partie, nous allons voir comment utiliser cette même classe pour envoyer des requêtes de type HTTP POST. Même s'il permet aussi de recevoir une réponse du serveur à ces requêtes, ce type de requête permet d'**envoyer** des informations.

Envoi d'une chaîne de caractères

Voici un exemple de requête POST effectuée en JavaScript/AJAX:

```
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://httpbin.org/post');
xhr.onreadystatechange = function() {
  if (xhr.readyState === 4) {
    alert(xhr.responseText);
  }
};
// envoi d'une chaîne de caractères:
xhr.send('ceci est un exemple de données envoyées');
```

Le principe de fonctionnement est exactement le même que celui d'une requête GET, sauf que nous avons cette fois-ci **envoyé une chaîne de caractères** via notre requête à l'adresse serveur `https://httpbin.org/post`.

Pour cela, nous avons:

- remplacé le paramètre GET par POST, dans l'appel à la méthode `open()`,
- et passé une chaîne de caractères en paramètre de l'appel à la méthode `send()`.

Envoi d'un objet JavaScript / JSON

Pour envoyer un objet JavaScript / JSON dans une requête POST, il faut d'abord sérialiser l'objet (c'est à dire: le convertir) sous forme de chaîne de caractères, à l'aide de la fonction `JSON.stringify()`.

Il suffit donc de modifier le paramètre passé à la méthode `send()`, tel que dans l'exemple suivant:

```
// ... ou envoi d'un objet JSON:
xhr.send(JSON.stringify({ message: 'bonjour!' }));
```

En effet, `JSON.stringify()` est la fonction inverse de `JSON.parse()`:

- alors que `JSON.parse()` permet de convertir une chaîne de caractères en objet,
- `JSON.stringify()` permet de convertir un objet en chaîne de caractères.

Attention: veillez à ne pas appeler la méthode `send()` d'une même instance (ex: `xhr`, dans notre exemple) plus d'une seule fois !

Conseils pratiques pour diagnostiquer le fonctionnement de vos requêtes

- Comme d'habitude, n'oubliez pas de consulter la console de votre navigateur pour vérifier la présence éventuelle d'erreurs dans votre code et/ou dans vos requêtes AJAX. (ex: accès non autorisé à une API)
- Utilisez l'onglet "Réseau" (ou *Network*, en anglais) de Chrome Dev Tools pour suivre l'exécution de vos requêtes, et consulter à la fois leur contenu et celui de la réponse du serveur.
- Utilisez l'API <https://httpbin.org/post> pour tester le bon fonctionnement de vos requêtes POST. Cette API vous envoie en réponse le contenu que le serveur a reçu suite à votre requête.