

Ajax

L3 I - DWA

AJAX

2017-2018

marc.lemaire@u-cergy.fr



Ajax : « *Asynchronous JavaScript and XML* »

- permet de modifier une partie de la page Web via une requête complémentaire en arrière-plan (fragment d'information)
- Mise à jour sans rechargement complet de la page.
- Web 2.0
- Existe depuis 2005

Principaux composants Ajax

- DOM : Document Object Model (HTML & CSS)
- JavaScript (natif ou via bibliothèques / framework : ex. jquery)
- XML (et XSL), mais d'autres formats sont possibles : texte brut, HTML, javascript, JSON, (binaire)
- Base asynchrones (mais en réalité des requêtes synchrones sont possibles)
- L'objet XMLHttpRequest

Quelques exemples Ajax

- Google suggest, Google map, google docs, etc.
- Vérification de la disponibilité d'un login
- Adaptation d'une liste déroulante d'un formulaire en fonction de choix précédents.

Ajax : Avantages et inconvénients

- + : Ergonomie : « application » Web
- + / - : Plus d'échanges réseau mais volume réduit (diminution de la bande passante)
- + : bien adapté à la navigation sur smartphone.
- - : Pas de support de l'accessibilité
- - : Peut être inhibé (puisque lié à JavaScript)
- - : Favoris, référencement, historique (cf. html5 history api)

Ajax : XMLHttpRequest 2

- Timeout (délai) pour la réponse du serveur
- Téléchargement de données binaires possible (ex. audio / vidéo) : responseType = 'blob'
- Cross-site : CORS (Cross Origine Resource Sharing) mais « Access Across Domains » bloqué par les navigateurs (sécurité).
- L'objet FormData

Ajax : Les propriétés de XMLHttpRequest

Propriété	Description
readyState	Retourne l'état de la requête.
onreadystatechange	Gestionnaire d'événements qui prend en charge les changements d'état de la requête.
status	Renvoie le code numérique de la réponse du serveur HTTP.
statusText	Renvoie le message lié au code numérique de la réponse du serveur HTTP.
responseText	Réponse du serveur sous forme de chaîne de caractères.
responseXML	Réponse du serveur sous forme d'un document XML.
timeout	Temps d'attente maximum accordé par le navigateur pour obtenir une réponse du serveur.

Ajax : Les méthodes de XMLHttpRequest

Méthode	Description
open()	Initialise la requête selon une série de paramètres fournis en argument.
send()	Effectue la requête, avec éventuellement l'envoi de données.
getAllResponseHeaders()	Renvoie l'ensemble de l'en-tête HTTP de la réponse sous la forme d'une chaîne de caractères.
getResponseHeader()	Renvoie la valeur d'un seul champ de l'en-tête HTTP sous la forme d'une chaîne de caractères.
setRequestHeader()	Assigne une valeur à un champ d'en-tête HTTP éventuellement envoyé lors de la requête.
overrideMimeType()	Force un document à être traité selon un type de contenu (MIME Type) particulier.
abort()	Annule la requête.

Ajax : Éléments de syntaxe

- `var xhr = new XMLHttpRequest();`
 - `open(méthode, url de la resource, mode)`
 - `send()`
- `xhr.open("GET", "test.txt", true);`
- `xhr.send(null);`

Ajax : la propriété readyState

Valeur	Description
0 ou uninitialized	La requête n'est pas initialisée.
1 ou loading	Début du transfert des données.
2 ou loaded	Les données sont transférées.
3 ou interactive	Les données reçues sont partiellement accessibles.
4 ou complete	Les données sont complètement accessibles.

Ajax : exemple de syntaxe

```
xhr.onreadystatechange = function() {  
    if(xhr.readyState == 4) {  
        // instructions  
    }  
}
```

Les codes de retour HTTP

valeur	signification
1xx	Message informatif
2xx	Requête traitée
3xx	redirection
4xx	Erreur du client
5xx	Erreur du serveur

Ajax : la propriété status

Code	Description
200	OK
301	Moved Permanently
302	Found
307	Temporary Redirect
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Ajax : exemple de syntaxe

```
xhr.onreadystatechange = function() {  
    if(xhr.readyState == 4 &&  
        xhr.status == 200) {  
  
        // instructions  
    }  
}
```

Ajax : Autres propriétés indispensables

- **Propriété `responseText`**
 - Texte brut ou formaté HTML
- **Propriété `responseXML`**
 - Réponse sous la forme de données XML
- **Propriété `timeout`**
 - nombre de millisecondes que le navigateur peut attendre avant d'obtenir la réponse du serveur. La valeur par défaut est 0.
 - Au delà, la demande est annulée (cf. `ontimeout`)

Ajax : exemple de syntaxe

```
var xhr = new XMLHttpRequest();
xhr.onreadystatechange = function() {
    if (this.readyState == 4
        && this.status == 200) {
        document.getElementById("demo")
            .innerHTML = this.responseText;
    }
};
xhr.open("GET", "ajax_info.txt", true);
xhr.send();
```


Ajax : choix de mise en oeuvre

- url : fichier du (même) serveur : statique (txt, xml, ...) ou dynamique (php, ...)
- responseText vs responseXML
- GET vs POST
- Synchrone vs Asynchrone

Ajax : mise en œuvre

- Réponse statique en architecture 2 tiers (fichiers : txt, html, csv, xml, ...)
- Réponse dynamique en architecture 2 tiers (source de type fichier : ex. csv)
- Réponse dynamique en architecture 3 tiers (avec connexion BD)

Ajax et responseXML

Principaux attributs et méthodes :

- `childNodes[] : Node[]`
- `parentNode : Node`
- `nextSibling : Node`
- `previousSibling : Node`
- `firstChild : Node`
- `lastChild : Node`
- `nodeValue : String`
- `documentElement : Element`

Ajax avec jQuery (1/2)

Principales méthodes

- `load()`, `get()`, `post()`

Syntaxe

- `$(selector).load(URL, data, callback);` // la réponse ajax sera placé dans l'élément de la page
- `$.get(URL, callback);`
- `$.post(URL, data, callback);`

Ajax avec jQuery (2/2)

Exemples

- `$("#div1").load("demo_test.txt");`
- `$("#div1").load("test.php", { limit: 25 });`
- `$("#div1").load("demo_test.txt", function(responseTxt, statusTxt, xhr){...});`
- `$.get("demo_test.asp", function(data, status){...});`
- `$.post("demo_test_post.asp", { iso : "es", name : "Spain" },
function(data, status){
alert("Data: " + data + "\nStatus: " + status);
});`

Ajax avec jQuery : compléments

- \$.ajax() : <http://api.jquery.com/jquery.ajax/>
- \$.getJSON :
<http://api.jquery.com/jquery.getJSON/>
- \$.each()

Références

- <https://developer.mozilla.org/en-US/docs/AJAX>
- https://developer.mozilla.org/en-US/docs/AJAX/Getting_Started
- https://www.w3schools.com/js/js_ajax_intro.asp
- <https://learn.jquery.com/ajax/>
- https://www.w3schools.com/jquery/jquery_ajax_intro.asp