



PHP et AJAX

Langage PHP avancé

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS

1



Séquence 6 : PHP et AJAX



Objectifs spécifiques : A la suite de cette séquence, l'étudiant doit être capable de :

1. Décrire la notion d'application RIA (Rich Internet application)
2. Les différentes technologies utilisées dans le développement de RIA
3. Décrire la technologie AJAX
4. Utiliser AJAX pour rendre plus rapide et plus interactif une application web dynamique

Séquence 6 : PHP et AJAX



Plan de la séquence :

Introduction

1. Notions d'application RIA
2. La technologie AJAX
3. Fonctionnement d'AJAX
4. L'objet XMLHttpRequest
5. Utilisation d'AJAX avec PHP

Conclusion

Notions d'application RIA

Développement d'Application RIA

- Une application RIA (Rich Internet application) ou application Internet riche, est une application Web qui offre des caractéristiques similaires aux logiciels traditionnels installés sur un ordinateur.
- La dimension interactive et la vitesse d'exécution sont particulièrement soignées dans ces applications Web.

Technologies utilisées

- Plusieurs technologies peuvent être utilisées pour développer une application RIA. Les plus connues sont Adobe Flex/AIR, Microsoft Silverlight et AJAX
- Adobe Flex & Microsoft Silverlight sont des solutions coûteuses mais offrent de larges possibilités, notamment aux designers
- Alors qu'il est difficile de faire de l'AJAX sans Framework. De plus la maintenance est difficile et le coût de développement élevé.

La technologie AJAX

- AJAX est l'acronyme de **A**synchronous **J**avaScript And **X**ML.
- AJAX n'est pas un langage de programmation.
- Mais utilise simplement la combinaison :
 - d'un objet XMLHttpRequest intégré au navigateur (pour demander des données à partir d'un serveur Web),
 - de JavaScript et du DOM HTML (pour afficher ou utiliser les données)
- Les applications AJAX peuvent utiliser XML pour transporter des données, mais aussi des données sous forme de texte brut ou de texte JSON.
- AJAX permet de mettre à jour des parties d'une page Web, sans recharger la page entière.
- Pour des raisons de sécurité, Ajax ne fonctionne que si la page Web et le fichier XML ou texte qu'il tente de charger sont situés sur le même serveur.

Fonctionnement d'AJAX

Navigateur

Un événement se produit

- Création d'un objet XMLHttpRequest
- Envoi d'une requête Http



Internet



Serveur

- Traitement de la requête
- Envoie d'une réponse à la page Web



Internet



Navigateur

- Traitement de la réponse avec JavaScript
- Mise à jour de la page

Fonctionnement d'AJAX

Exemple : (La page HTML)

```
<!DOCTYPE html>
<html>
<body>

<div id="demo">
  <h2>Test d'AJAX</h2>
  <button type="button" onclick="loadDoc()">Changer le contenu</button>
</div>

</body>
</html>
```

Fonctionnement d'AJAX

Exemple : (La fonction JavaScript loadDoc)

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

- Le fichier `ajax_info.txt` contient simplement du texte.

L'objet XMLHttpRequest

- La clé de voûte d'AJAX est l'objet XMLHttpRequest.
- Il est intégré dans tous les navigateurs pour accéder aux données d'un serveur.
- Mais il est aussi utilisé pour échanger des données avec un serveur.
- Il permet de :
 - mettre à jour une page web sans la recharger
 - demander des données au serveur après le chargement de la page
 - Recevoir des données du serveur après le chargement de la page
 - Envoyer des données au serveur en arrière plan

L'objet XMLHttpRequest

Créer un objet XMLHttpRequest

- Syntaxe pour créer un objet XMLHttpRequest:
variable = **new** XMLHttpRequest();
- Pour les anciennes versions d'Internet Explorer (IE5 et IE6) qui utilisent un objet ActiveX:
variable = **new** ActiveXObject("Microsoft.XMLHTTP");

Pour gérer tous les navigateurs, y compris IE5 et IE6, on vérifie si le navigateur prend en charge l'objet XMLHttpRequest. Si c'est le cas, on crée un objet XMLHttpRequest, sinon, on crée un ActiveXObject:

Exemple

```
var xhttp;  
if (window.XMLHttpRequest) {  
    xhttp = new XMLHttpRequest();  
} else { // code for IE6, IE5  
    xhttp = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

L'objet XMLHttpRequest

Méthodes d'objet XMLHttpRequest

Méthode	Description
<code>new XMLHttpRequest()</code>	Crée un nouvel objet XMLHttpRequest
<code>abort()</code>	Annule la requête en cours
<code>getAllResponseHeaders()</code>	Renvoie l'information d'en-tête
<code>getResponseHeader()</code>	Renvoie des informations d'en-tête spécifiques
<code>open(<i>method</i>,<i>url</i>,<i>async</i>, <i>user</i>,<i>psw</i>)</code>	Spécifie la requête <i>method</i> : le type de requête GET ou POST <i>url</i> : l'emplacement du fichier <i>async</i> : true (asynchrone) ou false (synchrone) <i>user</i> : nom d'utilisateur facultatif <i>psw</i> : mot de passe facultatif
<code>send()</code>	Envoie une requête au serveur. Utilisé pour les requêtes GET
<code>send(<i>string</i>)</code>	Envoie une requête au serveur. Utilisé pour les requêtes POST
<code>setRequestHeader()</code>	Ajoute une paire étiquette/valeur à l'en-tête à envoyer

L'objet XMLHttpRequest

Propriétés de l'objet XMLHttpRequest

Propriétés	Description
onreadystatechange	Définit une fonction à appeler lorsque la propriété readyState change
readyState	Définit l'état de l'objet XMLHttpRequest. 0: requête non initialisée 1: connexion au serveur établie 2: requête reçue 3: traitement de la requête en cours 4: requête terminée et la réponse est prête
responseText	Renvoie la réponse sous forme de chaîne
responseXML	Renvoie la réponse sous forme de données XML
Status	Renvoie le numéro d'état d'une requête 200: "OK" 403: "Interdit" 404: "Pas trouvé" Pour une liste complète, https://www.w3schools.com/tags/ref_httpmessages.asp
statusText	envoie le texte d'état (par exemple "OK" ou "Not found")

Utilisation avec PHP

Le fichier HTML (code JavaScript)

1. AJAX est utilisé pour créer des applications plus interactives.
2. L'exemple suivant montre comment une page Web peut communiquer avec un serveur Web alors que l'utilisateur tape des caractères dans un champ de saisie:

```
function showLivre(str) {  
    if (str.length == 0) {  
        document.getElementById("txtLivre").innerHTML = "";  
        return;  
    } else {  
        var xmlhttp = new XMLHttpRequest();  
        xmlhttp.onreadystatechange = function() {  
            if (this.readyState == 4 && this.status == 200) {  
                document.getElementById("txtLivre").innerHTML = this.responseText;  
            }  
        };  
        xmlhttp.open("GET", "getlivre.php?q=" + str, true);  
        xmlhttp.send();  
    }  
}
```

Utilisation avec PHP

Le fichier HTML (le formulaire)

```
<body>
  <p><b>Taper un nom dans le champ de saisie ci-dessous:</b></p>
  <form> Nom livre: <input type="text" onkeyup="showLivre(this.value)">
</form>
  <p>Suggestions: <span id="txtLivre"></span></p>
</body></html>
```

Explication de code:

- Tout d'abord, on vérifie si le champ de saisie est vide (`str.length == 0`).
- Si oui, on efface le contenu de l'élément `txtLivre` puis on quitte la fonction.
- Si le champ de saisie n'est pas vide, on procède comme suit:
 - On crée un objet `XMLHttpRequest`
 - On appelle la fonction à exécuter lorsque la réponse du serveur est prête
 - On envoie la demande à un fichier PHP (`getlivre.php`) dans le serveur
 - Notons que le paramètre `q` est ajouté à **"getlivre.php?q="+ str**
 - La variable `str` contient le contenu du champ de saisie

Utilisation avec PHP

Le fichier PHP - "getlivre.php"

- Il vérifie un ensemble de noms et renvoie les noms correspondants au navigateur:

```
<?php $txt = "";  
$tab = array ("Long walk to freedom", "Une si longue lettre",...); // Tableau des livres  
$q = $_REQUEST["q"]; // Récupère la paramètre  
  
if ($q !== "") { // Recherche toutes les propositions de array si $ q est différent de ""  
    $q = strtolower($q);  
    $len=strlen($q);  
    foreach($tab as $name) {  
        if (stristr($q, substr($name, 0, $len))) {  
            if ($txt=== "") {$txt= $name;}  
            else {$txt .= ", $name"; }  
        }  
    }  
}  
echo $txt === "" ? "pas de suggestion" : $txt;  
?>
```