

# XMLHTTPREQUEST (REQUETE XMLHTTP) SIMPLE

## OBJECTIF(S)

Mettre en œuvre et traiter une requête de type XMLHttpRequest.

Vérifier l'unicité d'un pseudonyme lors de la saisie (le pseudonyme est-il déjà référencé dans une application distante ? La vérification est faite avec Javascript, incluant une requête XMLHttpRequest).

## ÉNONCÉ

Vous devez construire un formulaire avec un unique champ de saisie : celui-ci permet à l'utilisateur de saisir un « pseudo » libre. Une requête est faite à un script placé sur un serveur pour chaque caractère saisi : le serveur répondra vrai (1) si le pseudo (en l'état de la saisie) est libre, faux (0) si le pseudo est déjà utilisé ou un ? si aucun pseudo n'est donné.

## ETAPES / BILAN

### a) Formulaire

Créez le formulaire avec un champ de saisie de type « text »

### b) Gestionnaire d'événements

Mettez en place un gestionnaire d'événements à chaque saisie de caractère (la syntaxe du gestionnaire d'évènement est onkeyup – lorsque l'on relâche une touche du clavier). Ce gestionnaire d'évènements invoque une fonction Javascript. Dans un premier temps, contentez vous d'afficher le contenu du champ de saisie dans une boîte de dialogue = fonction alert(string) en Javascript (affiche la valeur reçue en paramètre dans une boîte de dialogue).

Cette étape est validée lorsqu'à chaque saisie d'un nouveau caractère dans le champ du formulaire, la boîte de dialogue vous affiche le contenu du champ.

### c) Vérification de la saisie puis traitement

Vous devez rédiger une fonction Javascript et l'invoquer (ou alors compléter votre fonction existante) qui vérifie que le pseudo saisi contient au moins 4 caractères, sinon, l'utilisateur doit être averti par un message écrit à côté du champ de saisie (vous devez utiliser les fonctions d'interaction avec le DOM pour cela).

### d) Requête AJAX au serveur

Si le champ de saisie contient au moins 4 caractères, il vous faut soumettre une requête XMLHttpRequest au serveur pour l'interroger sur la disponibilité ou la non disponibilité du pseudo (pour l'exercice seuls les pseudos **toto** et **titi** sont déjà pris).

Pour établir la requête, inspirez vous de l'exemple donné en cours, en tenant compte des changements d'état de l'interaction avec le serveur (cf. onreadystatechange)

Si tout se passe bien, le serveur répondra au format texte (la réponse n'est pas un document XML). **Vous recevrez 0 si le pseudo est déjà utilisé, 1 si le pseudo est libre (disponible)** ou ? si le serveur n'a pas reçu de pseudo en paramètre.

Le serveur attend une requête à l'URL

<http://webu2.upmf-grenoble.fr/shs/jb/ajaxsimple/serveur.php>

avec un paramètre pseudo=valeursaisie

*Remarque : pour des raisons de sécurité, les requêtes XMLHttpRequest ne sont pas permises sur un URL différent du document faisant la requête dans certains navigateurs (Mozilla par exemple). Cela est largement connu et détourné par l'utilisation d'un script communément appelé « proxy » (programmé en PHP par exemple), qui est placé sur le même serveur que le document appelé et se contente de relayer la requête à un autre serveur (communément, il est possible de lire un fichier à un URL distant en PHP, ce « proxy » se contente donc de recevoir la requête AJAX puis de lire la cible de la requête et de retourner son contenu sur le flux courant).*

Vous pouvez charger un exemple de script PHP « proxy » à l'URL suivant :

<http://webu2.upmf-grenoble.fr/shs/jb/ajaxsimple/proxy.zip>

Votre requête XMLHttpRequest invoquera donc ce « proxy » (qui se charge de transmettre la requête au serveur tiers et retourne le résultat)

Le proxy attends un paramètre transmis avec la méthode GET : pseudo: la valeur saisie dans le champ de formulaire.

NB : dans la mesure où vous aurez besoin d'exécuter le proxy écrit en PHP, vous aurez besoin de travailler sur un serveur local à même d'exécuter du PHP – Si vous n'avez pas de serveur installé sur votre machine, vous pouvez utiliser des environnements d'installation « tout prêts » comme easyPHP, WAMP, Movamp... (et bien sûr solliciter mon aide sur ce plan)

### **e) Traitement de la réponse**

Une fois la réponse reçue (0, 1 ou ?), vous devez la traiter / l'interpréter et informer l'utilisateur sur la disponibilité du pseudo (vous devez utiliser les fonctions d'interaction avec le DOM pour afficher le message)

### **f) Question**

Je trouve que l'évènement onkeyup pour vérifier la saisie est « exagéré » (avis personnel).

Etes vous d'accord ?

Pourquoi ?

Si vous êtes d'accord, quelle alternative pourriez vous envisager ?