

Ecrire un client Twitter simple avec Zend Framework et la librairie OAuth

par Pádraic Brady (Auteur) AIT YAHIA Idir (Traducteur) ([Mon site perso](#)) ([Blog](#))

Date de publication : 17 octobre 2009

Dernière mise à jour :

Cet article est une traduction de l'original  **Writing A Simple Twitter Client Using the PHP Zend Framework's OAuth Library (Zend_Oauth)**. Qui a pour objectif de vous montrer comment créer un client Twitter simple avec Zend Framework

I - Introduction.....	3
II - Qu'est-ce que OAuth?.....	3
III - La Préparation est toujours inévitable !.....	3

I - Introduction

Hier, J'ai finalement eu le temps de finir et rafistoler Zend_Oauth, l'implémentation du consommateur d'OAuth Core 1.0 Revision A specification. Après l'avoir terminé, je l'ai utilisé pour écrire rapidement une interface assez simple pour publier quelques tweets sur Twitter pour mes tests. Avec un peu de documentation et quelques tests unitaires en extra, implémentation consommateur devra trouver sa place dans la version 1.10 de Zend Framework ainsi que l'implémentation du serveur, enfin je crois.

Dans cet article je vais scruter comment écrire rapidement un client Twitter que vous pourrez utiliser pour publier des tweets (ces petits messages de moins de 140 caractères) une fois que vous est autorisé via le protocole OAuth.

Vous pouvez télécharger tous les fichiers nécessaires (assurez vous de les modifier tel qu'il est décrit) ou les extraire à partir de git à l'adresse suivante:  <http://github.com/padraic/Tweet-Lite/tree/master>

II - Qu'est-ce que OAuth?

Si vous ne savez pas ce que c'est **OAuth**, la spécification d'OAuth le décrit ainsi :

Le protocole OAuth permet aux sites Web ou à des applications (consommateur) d'accéder à des ressources protégées d'un web service (fournisseur de service) via une API, sans contraindre les utilisateurs à divulguer leurs informations d'identification du fournisseur de services aux consommateurs. En général, OAuth crée une méthodologie librement implémentables et générique pour l'API d'authentification.

En d'autres termes, c'est un moyen de permettre à des sites Web d'accéder à vos données sur d'autres services via un service API, de même que l'API de Twitter ou Google Gdata, sans fournir à ces sites Web votre nom d'utilisateur et mot de passe. Au lieu de cela, OAuth vous permet d'autoriser de tels sites à accéder à vos données sans qu'ils aient besoin de votre nom d'utilisateur ou mot de passe, ils utilisent juste un jeton (token) d'accès fourni par votre fournisseur de service, ainsi vous pouvez à tout moment leurs interdire l'accès si vous le désirez. L'avantage est immédiatement évident votre nom d'utilisateur et mot de passe ne sont pas partagées ou distribuées à des sites potentiellement indignes de confiance. La surabondance de services utilisant Twitter sont un excellent exemple, jusqu'à récemment ils réclament tous votre nom d'utilisateur et mot de passe Twitter et honnêtement, comment pouvez-vous savoir qu'il ne s'agit pas d'un abus ? Parce qu'ils le disent ? OAuth élimine ce problème.

Le protocole fonctionne comme suit: le site web (consommateur) voulant avoir accès a vos données via un fournisseur de service, contact celui-ci en utilisant une requête http pour récupérer un jeton non autorisé. Le client vous redirige ensuite vers votre fournisseur de service pour que vous puissiez donner l'accès au client. URL de redirection va contenir le jeton non autorisé comme paramètre. Si vous approuvez la demande, vous serez redirigé vers le site qui est à l'origine de la demande avec un code de vérification lié à URL. Maintenant le site sait que vous lui avez accordé l'accès, alors il contact votre fournisseur de services en incluant le jeton fraîchement approuvé (une fois de plus) et le code de vérification dans l'URL. La réponse devrait être un jeton garantissant un accès complet (associé à l'utilisateur) que le client pourra utiliser dans les futures demandes d'accès a vos données (jusqu'à son expiration ou que vous lui retiriez son autorisation). À présent, vous pouvez vous débarrasser du jeton de la demande - dans le jargon d'OAuth vous avez échangé un jeton de demande d'accès non autorisé contre un jeton d'accès autorisé.

III - La Préparation est toujours inévitable !

Avec cette compréhension entre les mains, passons à la réalisation de ce petit client Twitter comme exemple! Vous aurez besoin de télécharger Zend Framework (la dernière version ou par Subversion). Vous aurez besoin aussi de récupérer Zend_OAuth depuis l'incubateur de Zend Framework vu qu'il ne fait pas partie du dépôt principal. Une fois que vous avez tout enregistré quelque part, notez bien les chemins de ces librairies pour que vous puissiez les inclure par la suite dans votre PHP include_path.

Du côté de Twitter nous avons trois étapes :

Premièrement, obtenez un compte Twitter! J'espère que vous en avez déjà un le suivant est le mien @padraicb.

Deuxièmement, vous avez besoin de configurer votre système afin de définir un nouveau domaine local. Sous Linux vous y parviendrez en éditant /etc/hosts, vous aurez besoin des droits du superutilisateur root, vous pouvez utiliser la commande sudo comme ceci " sudo myeditor /etc/hosts ". Sous Windows le fichier équivalent peut être trouvé dans le dossier C:\Windows\System32\drivers\etc\hosts, vous aurez besoin des droits d'Administrateur pour pouvoir

l'éditer. Ajouter un domaine local est assez simple, il est nécessaire pour pouvoir utiliser l'API de Twitter en local, Twitter refuse toutes requêtes venant depuis localhost ou 127.0.0.1, mais il ne filtre pas un nom de domaine local vu que c'est infaisable et ça mettrait en rogne la plus part des développeurs. Modifiez le fichier pour inclure une nouvelle entrée comme suit:

127.0.0.1 mytwitterclient.tld

Une fois enregistré, votre navigateur doit immédiatement vous rediriger vers localhost/127.0.0.1 à n'importe quelle adresse sous la forme d'<http://mytwitterclient.tld>. Si votre serveur web est démarré et que vous avez déjà un site à la racine de votre dossier WEB, vous y serez redirigé. Si vous trouvez ça gênant et que vous ne pouvez pas jouir d'une telle simplicité, vous pouvez jouer sur la configuration du Virtualhosting pour définir un chemin différent.

Troisièmement, le Fournisseur OAuth de Twitter ne donne pas un jeton d'accès pour chaque Tom, Dick et Harry, enfin si, mais il aimerait bien savoir au préalable que vous allez vous appeler Tom, Dick ou Harry, à cet effet vous avez besoin d'enregistrer de toutes les applications pour récupérer la clef d'accès à son services OAuth. C'est à peut prêt la même chose pour les fournisseurs de service similaire. Visitez l'adresse http://twitter.com/oauth_clients et connectez-vous en utilisant votre compte Twitter, vous serez amené sur une page où vous pourrez enregistrer de nouvelles applications/client ou d'en supprimer. Enregistrez une nouvelle application nommé la bon il vous semble, mais assurez vous de :

- a Choisir navigateur comme type d'application
- b Définir Callback URL à <http://mytwitterclient.tld/callback.php> (modifier le domaine et le chemin de vos préférences)
- c Sélectionner l'accès en mode lecture/écriture vu le fait qu'on a besoin d'envoyer de nouveaux tweets donc d'un accès en écriture.

Nous n'allons pas utiliser twitter pour se connecter. Immédiatement après l'enregistrement vous serez redirigé sur une page qui contient URLs à des actions variées OAuth et le plus important, la clef et code secret du consommateur que vous utiliserez dans votre application. Soyez sans crainte vous pouvez revisiter cette page à tout moment.