# Passer ses applications Twitter à OAuth

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 17 juillet 2010

https://www.bortzmeyer.org/twitter-oauth.html

---

On le sait, Twitter a annoncé l'arrêt du mécanisme d'authentification simple à partir du 16 août 2010 <http://www.supertweet.net/countdown>. Le mécanisme à utiliser désormais est OAuth (RFC 5849 [1]). Voici un exemple de passage à OAuth pour un petit programmeur Python.

L'ancien mécanisme d'authentification était le "*Basic Authentication*" décrit dans le RFC 7617. Son gros avantage est la simplicité pour le programmeur : il suffit de concaténer un identificateur et le mot de passe, de les encoder en Base64 et de les mettre dans l'en-tête HTTP `Authorization:` et c'est tout. D'une manière générale, la simplicité de l'API a d'ailleurs beaucoup fait pour le succès de Twitter.

Mais ce mécanisme très simple est aussi très dangereux. En effet, tout repose sur la connaissance du mot de passe. Celui-ci doit être connu des applications, voire des applications tierces, et il ne reste donc pas secret longtemps. D'où le passage à OAuth, qui permet une meilleure sécurité, au prix d'une plus grande complexité (et qu'il n'est pas entièrement possible de cacher dans la bibliothèque logicielle).

Heureusement, il y a quand même des bibliothèques qui aident et des articles qui expliquent. Question bibliothèque, comme je programme en Python, j'avais abandonné python-twitter <http://code.google.com/p/python-twitter/> qui, à l'époque, (en juillet 2010, dans sa version 0.6) n'avait pas de gestion de OAuth (c'est désormais fait, dans la version 0.8 et vous pouvez lire un article d'exemple en français <http://linux.leunen.com/?p=1038>). Pour moi, je suis passé à l'excellent Tweepy <http://joshthecoder.github.com/tweepy/> qui dispose d'une très bonne documentation <http://joshthecoder.github.com/tweepy/docs/index.html>. Question article, j'ai simplement suivi les instructions dans « "*Twitter from the command line in Python using OAuth*" <http://jmillerinc.com/2010/05/31/twitter-from-the-command-line-in-python-using-oauth> ». Voici les étapes parcourues pour le petit script qui me sert à informer des mises à jour de mon blog. Je ne fais que paraphraser en français l'excellent article ci-dessus, qui inclus en outre d'utiles copies d'écran.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5849.txt

1) Enregistrer l'application. Attention au nom choisi, car il sera affiché par Twitter (j'ai choisi `Blog Bortzmeyer`). Cela se fait en `http://twitter.com/oauth_clients`. Choisir Client (et pas "*browser*") pour un logiciel en ligne de commande. Choisir "*Read & Write*" pour une application qui permettra de faire des mises à jour. Choisir Twitter pour l'authentification.

2) Lancer l'application à usage unique, qui permettra d'obtenir les éléments d'authentification. Voici celle que j'ai utilisé :

```
#!/usr/bin/env python

# First connection to Twitter, to get the credentials

import tweepy

CONSUMER_KEY = 'OBTENUE EN ÉTAPE 1'
CONSUMER_SECRET = 'OBTENU EN ÉTAPE 1'

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth_url = auth.get_authorization_url()
print 'Please authorize: ' + auth_url
verifier = raw_input('PIN: ').strip()
auth.get_access_token(verifier)
print "ACCESS_KEY = '%s'" % auth.access_token.key
print "ACCESS_SECRET = '%s'" % auth.access_token.secret
```

Une fois qu'elle est lancée, elle affiche l'URL qu'il faut visiter avec son navigateur Web favori. On doit alors s'authentifier auprès de Twitter par le mécanisme habituel, autoriser l'application et, en échange, Twitter vous envoie un nombre, le PIN, qu'on peut alors saisir dans son terminal :

```
% python one-off-connection.py
Please authorize: http://twitter.com/oauth/authorize?oauth_token=YukdWz2zW8nf8HnLUtt67rftNiXcjbUvojrWN5dKRs
PIN: 185867
ACCESS_KEY = 'MA CLÉ'
ACCESS_SECRET = 'MON SECRET'
```

3) Il ne reste plus qu'à utiliser ces éléments d'authentification dans son programme. Voici un exemple :

```
#!/usr/bin/python

import tweepy

message = "TEST"

twitteroauthfile = "/home/bortzmeyer/.twitter/oauth"

if not os.path.exists(twitteroauthfile):
    print >>sys.stderr, ("Cannot find %s" % twitteroauthfile)
    sys.exit(1)

twitteroauth = open(twitteroauthfile)
twitter_consumer_key = twitteroauth.readline()[:-1]
twitter_consumer_secret = twitteroauth.readline()[:-1]
twitter_access_key = twitteroauth.readline()[:-1]
twitter_access_secret = twitteroauth.readline()[:-1]

auth = tweepy.OAuthHandler(twitter_consumer_key, twitter_consumer_secret)
auth.set_access_token(twitter_access_key, twitter_access_secret)
api = tweepy.API(auth)
status = api.update_status(message)
# No method to display a status in Tweepy, it seems
print status
```

Une autre solution, après la fin de l'authentification de base dans quelques semaines, serait d'utiliser un relais, comme celui que propose Supertweet <http://www.supertweet.net/>, ce qui limiterait les modifications à faire aux applications.

---

https://www.bortzmeyer.org/twitter-oauth.html