

Utilisation de l'API de Twitter pour envoyer des sms groupés

<http://geekfg.blogspot.com> rev1

Par Ribreau François-Guillaume

Pré-requis :

- **2 comptes twitter**
 - Votre compte twitter personnel « **twittperso** » (configuré en notification « on » sur le deuxième compte, les notifications sur mobiles doivent être activées)
 - Et un compte twitter nommé par exemple « **twittbot** » qui permettra d'envoyer les sms à une partie de la liste d'amis qui le suit.
- Un serveur **Php & Mysql** avec l'extension **Curl d'activé** (nous utiliserons **Curl** dans notre exemple, mais les sockets font aussi très bien l'affaire)

Constitution de la base de données

Nous allons nous baser sur un cas concret, nous avons une table contenant une liste d'utilisateur, et nous souhaitons prendre que les utilisateurs qui ont accepté les notifications sms (champs « **notif_sms** » à 1).

Code sql pour générer la base de données :

```
CREATE TABLE `utilisateur` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `login` VARCHAR( 25 ) NOT NULL ,  
  `pwd` VARCHAR( 25 ) NOT NULL ,  
  `login_twitter` VARCHAR( 35 ) NOT NULL ,  
  `notif_sms` INT NOT NULL DEFAULT '0'  
) ENGINE = MYISAM ;
```

Code sql pour remplir la table pour notre exemple :

```
INSERT INTO `utilisateur` ( `id` , `login` , `pwd` , `login_twitter` ,  
  `notif_sms` )  
VALUES (NULL , 'admin', 'd8z4:!,dD', 'twittperso', '1'),  
  (NULL , 'lambda', 'd8z4Zs6!:', 't_lamb', '1'),  
  (NULL , 'bad_user', 'motdepasse', '', '0');
```

Nous avons donc une base de donnée, contenant 2 utilisateurs acceptant les notifications sms, et un autre non « **bad_user** ».

Maintenant pour récupérer le login twitter des personnes acceptant les notifications sms (**notif_sms** à 1) on effectue la requête :

Code sql pour récupérer la liste des logins twitter :

```
SELECT `login_twitter` FROM `utilisateur` WHERE `notif_sms` = 1
```

Utilisation de l'API de Twitter pour envoyer des sms groupés

<http://geekfg.blogspot.com> rev1

Par Ribreau François-Guillaume

Du côté de Twitter maintenant

Voici les différentes étapes qui doivent être faites par l'utilisateur pour pouvoir prétendre à recevoir des notifications sms depuis notre programme.

1. L'utilisateur doit créer un compte sur Twitter
2. L'utilisateur doit activer son téléphone portable pour le lier à son compte twitter, et passer les notifications sms à on (sur Twitter toujours)
3. L'utilisateur doit aller sur la page <http://twitter.com/twitbot> puis cliquer sur **follow** et mettre **notification à on**.
4. Le compte **twitbot** doit avoir accepté l'utilisateur en tant qu'amis sous Twitter (et inversement)
5. Pour vérifier que tout est bien fait, **connectez vous depuis votre compte twitbot**, et vérifié que vous pouvez **envoyer un message privée** à l'utilisateur

L'avantage de Twitter est qu'il offre un API simple d'utilisation (cf : [la page API de Twitter](#)).

Ce qui nous intéresse dans cet article est la fonction **d'envoi de message privé** à une personne précise via son **login twitter**.

L'url à utiliser pour envoyer un message privé à quelqu'un est la suivante :

http://twitter.com/direct_messages/new.format

Ainsi si on test dans le navigateur, à l'adresse :

http://twitter.com/direct_messages/new.xml

On observe une fenêtre d'authentification, remplissez là avec les identifiants de votre compte **twitbot**. Bien sur, Twitter retourne une erreur, car il faut que les paramètres **user** et **text** soit passé en **POST** dans la requête http.

- Le paramètre **user** peut être **l'id utilisateur** ou le **login** de l'utilisateur sur Twitter.
- Le paramètre **text** doit être encodé et ne peut être transmis en brut (sauf dans quelques cas particuliers), la fonction **urlencode() de Php** répondra à nos attentes sur ce point.

Le script PHP

Avant tout chose, nous devons nous connecter à la base de données (nommée **\$db_name**, avec les identifiants **\$db_user** et **\$db_pwd** sur le serveur sql **\$db_sql**) et récupérer la liste des logins Twitter acceptant nos notifications sms.

Script Php de connexion à la base de données et initialisation des variables :

```
<?php
$db_sql = '127.0.0.1';
$db_name = 'test';
$db_user = 'root';
$db_pwd = '';
```

Utilisation de l'API de Twitter pour envoyer des sms groupés

<http://geekfg.blogspot.com> rev1

Par Ribreau François-Guillaume

```
mysql_connect($db_sql,$db_user,$db_pwd); //connexion au serveur sql
$accessdb = mysql_select_db($db_name); //connexion à la base de donnée

//Compte du bot d'envoi des messages
$bot_login = 'twittbot';
$bot_pwd = 'twittbot';

$text = 'Hello world'; //Texte à envoyer

?>
```

Puis on récupère la liste des logins Twitter via notre requête sql énoncée plus haut.

Script Php pour récupérer la liste des logins Twitter

```
<?php
$sql = mysql_query(« SELECT `login_twitter` FROM `utilisateur` WHERE
  `notif_sms` = 1 »);
?>
```

Ensuite, dans une boucle, nous traitons utilisateurs par utilisateurs l'envoi d'un message privé.

Script Php permettant l'envoi de message privé via Twitter :

```
<?php

while($sql = mysql_fetch_object($sql))
{
    //Le login de l'utilisateur en cours est compris dans $sql-
    >login_twitter

    $twitter_API = 'http://twitter.com/direct_messages/new.xml';

    //Initialisation de Curl
    $curl_twitter = curl_init();

    $options = array(CURLOPT_URL => $twitter_API,
        CURLOPT_CONNECTTIMEOUT => 1,
        CURLOPT_HTTPAUTH => CURLAUTH_BASIC,
        CURLOPT_HEADER => false,
        CURLOPT_POST => true,
        CURLOPT_POSTFIELDS => 'user='.$sql-
        >login_twitter.'&text='.urlencode(stripslashes($text)),
        CURLOPT_USERPWD => $bot_login.':'.$bot_pwd,
        CURLOPT_RETURNTRANSFER => true);

    curl_setopt_array($curl_twitter, $options);
    $curl_result = curl_exec($curl_twitter);

    curl_close($curl_twitter);

    //On test si on a un résultat
    if(!empty($curl_result))
    {
        //La variable $curl_result contient le xml renvoyé par
        l'API
    }
}
```

Utilisation de l'API de Twitter pour envoyer des sms groupés

<http://geekfg.blogspot.com> rev1

Par Ribreau François-Guillaume

```
        echo 'Message envoyé avec succès à ' . $sql-
    >login_twitter.' <br />';
    }
    else
    {
        echo 'Erreur lors de l\'envoi du message à ' . $sql-
    >login_twitter.' <br />';
    }
}
?>
```

On initialise donc un objet **Curl** (appelé **\$curl_twitter**) on spécifie dans l'entête qu'il y a une authentification, et on précise le **login et le pwd** grâce à l'attribut **CURLOPT_USERPWD**. On spécifie aussi les paramètres à envoyer en mode **POST** grâce à **CURLPOST_POSTFIELDS**.

Mot de fin

Voilà c'est fini, il ne vous reste plus qu'à améliorer le script. J'ai donné une version « simple » pour que le programmeur lambda puisse comprendre le fonctionnement dans de bonne condition.

Si vous avez des remarques