

Bloguer par SMS avec Wordpress - Plugin de micro-blogging avec l'API Orange

par François Marx

Date de publication : 08/04/2008

Dernière mise à jour : 30/05/2008

Ce tutorial a pour but de créer un plugin wordpress pour le micro-blogging en utilisant les API d'Orange.

I - Introduction.....	3
II - Structure d'un plugin Wordpress.....	4
III - Principe du plugin et configuration des API.....	5
IV - Code du plugin.....	6
IV-A - Principales fonctions du plugin.....	6
IV-B - Code.....	8
IV-C - Le résultat final.....	10
V - Conclusion.....	11

I - Introduction

Wordpress est un système de gestion de contenu écrit en PHP et l'un des moteurs de blog les plus populaires. Ce tutorial a pour but de créer un plugin wordpress pour le micro-blogging en utilisant les API d'Orange. Le micro-blogging est une pratique consistant à envoyer des messages courts (moins de 200 caractères) et à partager ce contenu. Twitter est l'application de micro-blogging la plus connue. Dans la suite, nous présenterons rapidement le système de plugin de Wordpress et l'API SMS d'Orange. La seconde partie sera consacrée à la description de la structure d'un plugin Wordpress. Puis, nous détaillerons les quelques fonctions essentielles en PHP du plugin

Comme indiqué précédemment, notre outil de micro-blogging est une extension ou plugin de Wordpress. Il existe une multitude de plugins wordpress, on trouve pas loin de 2000 plugins disponibles sur le site wordpress.org. Il faut souligner que le but de ce tutorial n'est pas de présenter Wordpress mais de combiner sur un exemple : plugin et web-services. Pour une description précise de Wordpress, on peut se référer au site wordpress.org

L'API SMS d'Orange sera la seconde brique de base de l'outil de micro-blogging. Les API d'Orange ont déjà fait l'objet d'un **article** auquel je vous invite à vous référer pour davantage de détails sur les API d'Orange (1) . Pour poster de nouveaux articles, le blogueur enverra simplement un SMS à un numéro court. Pour utiliser les API d'Orange, l'utilisateur devra s'enregistrer sur le site d'**Orange Partner** pour obtenir sa clé d'accès au service et **configurer** les différents paramètres des API.

Le fonctionnement du plugin est très simple et l'ergonomie minimaliste. Le blogueur envoie un SMS au 20345 (prix d'un SMS 'normal') dont le premier mot sera utilisé uniquement pour l'aiguillage du SMS vers votre boîte mail. Le titre sera délimité par un dièse (#) et le reste du SMS correspond au contenu proprement dit du SMS. Le blogueur enverra le SMS suivant au 20345 pour poster un article (on admet qu'il a configuré 'post' comme mot clé) : 'post nouveau tuto#Je suis en train de terminer un plugin de micro-blogging' ce qui donnera sous Wordpress un article avec comme titre 'nouveau tuto', le corps de l'article 'je suis en train de terminer un plugin de micro-blogging' et la date du post sera l'arrivée du SMS dans la boîte mail.

II - Structure d'un plugin Wordpress

Ce plugin a été testé avec la version 2.3.3 de Wordpress (mais devrait fonctionner avec toute version supérieure à 2.1 de Wordpress) et PHP 5.2.4. Les plugins Wordpress sont des scripts PHP situés dans le répertoire wordpress \wp-content\plugins. Vous pouvez au choix créer un nouveau répertoire ou mettre directement votre script dans le répertoire plugins. Chaque plugin (i.e. le fichier php) commence par un en-tête particulier (indispensable, autrement votre plugin ne sera pas référencé par Wordpress).

En-tête plugin

```
/*
Plugin Name: Post by SMS
Plugin URI: http://www.developpez.com
Description: take the SMS forwarded in your mailbox and post it on your wordpress blog
Author: Francois Marx
Version: 0.3
Author URI: http://www.orangepartner.com
*/
```

Vous vous demandez peut-être maintenant comment votre script va être capable d'interagir avec Wordpress. La solution est d'utiliser un hook qui permet d'exécuter une fonction (définie par l'utilisateur) lorsque Wordpress effectue certaines actions (ajout d'un post, enregistrement d'un utilisateur...). Le site de [wordpress](#) fournit une liste de tous les hooks. Par exemple ce bout de code (extrait du plugin exemple Hello.php disponible dans le répertoire wordpress \wp-content\plugins) affiche les paroles de la chanson 'Hello, Dolly'. Chaque fois que Wordpress charge la page d'administration, Wordpress exécute la fonction hello_dolly. Plus précisément, la fonction est exécutée juste avant la fermeture du tag </body> dans la page d'administration. L'assignation du hook à la fonction hello_dolly se fait grâce la fonction add_action avec comme paramètre le nom du hook et la fonction à exécuter.

Plugin hook

```
// This just echoes the chosen line, we'll position it later
function hello_dolly() {
    global $chosen;
    echo "<p id='dolly'>$chosen</p>";
}

// Now we set that function up to execute when the admin_footer action is called
add_action('admin_footer', 'hello_dolly');
```

Il est souvent nécessaire de configurer des options dans les plugins. L'ajout d'option de configuration se fait par l'intermédiaire de la fonction add_options_page qui prend comme argument le titre de la page d'options, le titre du menu, le niveau d'accès minimum de l'utilisateur, le fichier et la fonction qui affiche le contenu de la page d'options.

Définition page d'administration

```
function sms_add_option_pages() {
    if (function_exists('add_options_page')) {
        add_options_page('Post by SMS', 'Post by SMS', 8, __FILE__, 'sms_options_page');
    }
}

function sms_options_page() {
    // code if needed
    // html form if needed
}
```

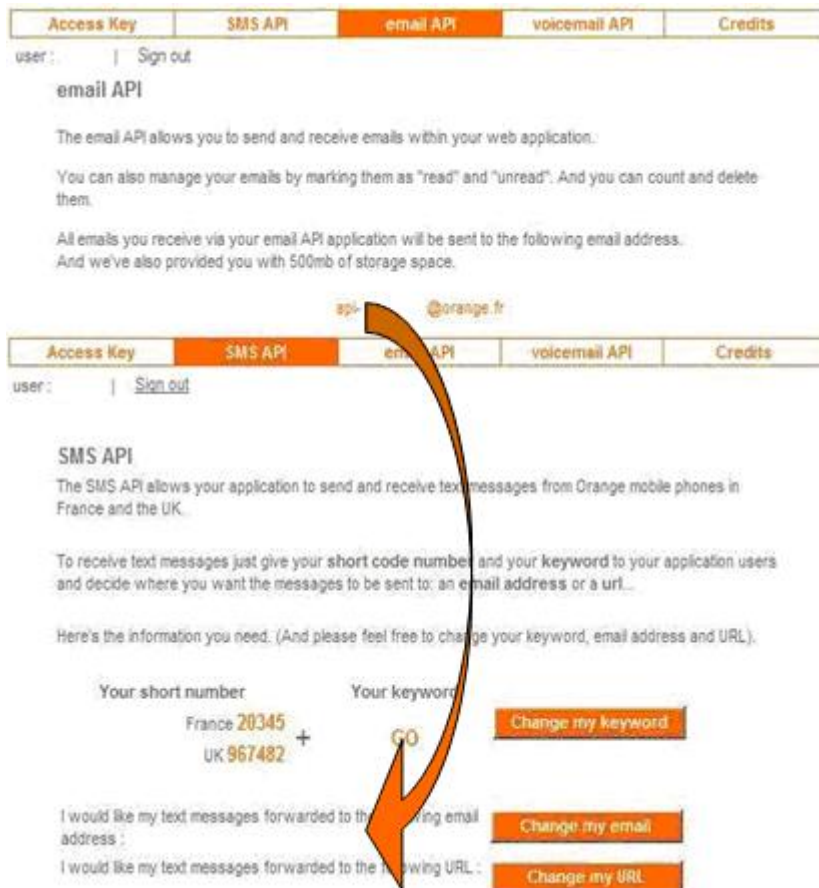
Après cette brève présentation de l'architecture d'un plugin Wordpress, nous allons entrer dans le vif du sujet en décrivant l'utilisation du plugin et la configuration des API.

III - Principe du plugin et configuration des API

Comme indiqué précédemment, le but est de pouvoir poster des messages sur un blog via des SMS. Le principe est le suivant:

- Envoi d'un SMS au 20345 commençant par un mot clé (choisi par l'utilisateur des API d'Orange)
- Le SMS est alors aiguillé vers la boîte mail en api-xxxx@orange.fr de l'utilisateur
- Un timer est activé pour sonder régulièrement la boîte mail et détecter l'arrivée de nouveaux SMS. Nous profitons de l'équivalent 'cron' de Wordpress. Note: il est déconseillé de vérifier l'arrivée de nouveaux messages trop fréquemment.
- Lorsque un SMS arrive, le contenu est extrait et 'posté' comme un article dans Wordpress

Nous allons maintenant configurer les API pour utiliser notre plugin. Tout d'abord, accédez à l'**IHM d'administration** pour configurer le routage du SMS. Je suppose que vous êtes déjà inscrit sur le site d'orangepartner dans le cas contraire, il faut d'abord s'enregistrer. Dans l'onglet email, notez l'adresse mail qui vous a été attribuée api-xxxx@orange.fr Dans l'onglet SMS, configurez le routage de vos SMS vers votre adresse mail API Orange (api-xxxx@orange.fr).



Redirection des SMS vers la boîte mail : configuration de l'API

Tout est maintenant prêt pour développer notre plugin. Bien entendu, je suppose que vous avez installé Wordpress. Si ce n'est pas fait, c'est le moment.

IV - Code du plugin

IV-A - Principales fonctions du plugin

Le squelette du plugin est le suivant:

```
<?php
add_action('admin_menu', 'sms_add_option_pages');
add_action('sms_check_post', 'sms_check_post');

function sms_add_option_pages();

function sms_options_page();

function sms_check_post();

function sms_more_reccurences();

add_filter('cron_schedules', 'sms_more_reccurences');
?>
```

Les fonctions sms_add_option_pages et sms_options_page permettent de configurer les options. Les options sont :

- L'intervalle de temps entre chaque interrogation de la boîte mail pour vérifier l'arrivée de nouveaux messages
- La clé d'accès aux APIs

Définir une page de configuration est aisé. Nous avons déjà indiqué que la fonction add_options_pages ajoute un onglet d'options pour votre plugin dans la page options de Wordpress. Le contenu de cet onglet est défini dans une fonction. Il s'agit le plus souvent d'un formulaire pour récupérer les paramètres de configuration de votre plugin. Le code ci-dessous illustre la page d'option du plugin de micro-blogging.

Code HTML pour l'affichage de la page d'options

```
function sms_options_page() {
    // config scheduler
    ?>
    <div class=wrap>
    <h2>Scheduled SMS</h2>
    <form method="post" action="<?php echo $_SERVER["REQUEST_URI"]; ?>">
        <input type="hidden" name="info_update" id="info_update" value="true" />
        Number of minutes between SMS checks<br>
        <input name="sms_delay" type="text" size="10" value="<?php echo get_option('sms_delay'); ?>" /><br>
        API access key<br>
        <input name="sms_api_access_key" type="text" size="20" value="<?
php echo get_option('sms_api_access_key'); ?>" />
        <div class="submit">
            <input type="submit" name="info_update" value="<?php _e('Update options'); ?> &raquo;" />
        </div>
    </form>
    </div>
    <?php
}
```



Copie d'écran de la page de configuration du plugin de micro-blogging

Dans la fonction sms_options_pages nous configurons aussi le planificateur avec le code suivant:

Initialisation planificateur code PHP

```
<?php
if (isset($_POST['info_update'])) {
    update_option('sms_api_access_key', (string)$_POST['sms_api_access_key']);
    update_option('sms_delay', (int)$_POST['sms_delay']);
    wp_clear_scheduled_hook('sms_check_post');
    wp_schedule_event(time(), 'sms_schedule', 'sms_check_post');
}
?>
```

Le planificateur a été introduit dans Wordpress 2.1. A chaque fois qu'une page est chargée, Wordpress compare l'heure courante à une liste de tâches planifiées et exécute les fonctions spécifiées. Il est possible de définir un évènement simple ou périodique. Dans l'exemple choisi, la fonction wp_schedule_event planifie l'exécution de la fonction sms_check_post avec la périodicité sms_schedule. sms_schedule est une périodicité définie par l'utilisateur. Wordpress définit deux options 'hourly' and 'daily'. Il est possible d'ajouter d'autres périodicités avec le bout de code suivant (la durée de l'intervalle est exprimé en secondes):

periodicity addition

```
function sms_more_reccurences() {
    return array(
        'sms_schedule' => array('interval' => 60*(int)get_option('sms_delay'), 'display' => 'SMS Schedule')
    );
}

add_filter('cron_schedules', 'sms_more_reccurences');
```

Le planificateur déclenche l'exécution de la fonction sms_check_post. La fonction sms_check_post récupère la liste des emails dans la boîte mail. Pour récupérer la liste des emails, nous utilisons la méthode getMailList des API d'Orange. Le code ci-dessous récupère la liste des emails et parcourt cette liste:

```
$url_list = "http://mail.alpha.orange-api.net/mail/getMailList.xml?id=$api_access_key";
$url_delete = "http://mail.alpha.orange-api.net/mail/deleteMail.xml?id=$api_access_key";
// get the list of all emails
$response = file_get_contents($url_list);
$xml_list = simplexml_load_string($response);

// loop on all emails
for ($n = count($xml_list->list->message); $n >= 1; $n--) {
    // TODO emails and SMS processing
}
```

Les mails dont le sujet commence par "Sms from" sont considérés comme des articles du blog. On récupère alors le contenu de ce mail à l'aide de la méthode getMail. Le contenu du message est alors 'parsé' pour supprimer le premier

mot qui correspond au mot clé nécessaire pour aiguiller le SMS dans votre boîte mail, extraire le titre de l'article et enfin son contenu. La chaîne de caractères comprise entre le premier espace et le premier dièse est considérée comme le titre, le reste du message correspond au contenu proprement dit de l'article. Le message est alors posté sur le blog à l'aide de la fonction `wp_insert_post`. Cette fonction prend comme paramètre le titre du 'post', le contenu du 'post', la date et le status. L'email est alors supprimé définitivement de la boîte mail avec la méthode `deleteMail`.

Post message

```
$response = file_get_contents($url_content."&number=".$n);
$xml_content = simplexml_load_string($response);
$body = $xml_content->message->body;
// remove the first words
$post_title = substr($body, strpos($body, ' '), strpos($body, '#')-strpos($body, ' '));
$post_title = $post_title.' (par SMS du '.substr($m->subject,9,11).')';
$post_content = substr($body, strpos($body, '#')+1);
$post_date = date('Y-m-d H:i:s', strtotime($xml_content->message->date)+get_option('gmt_offset')*3600);
$post_status = 'publish';
$post_data = compact('post_content', 'post_title', 'post_date', 'post_status');
$post_ID = wp_insert_post($post_data);
// delete emails
$response = file_get_contents($url_delete."&number=".$n);
```

IV-B - Code

Dans cette section, vous trouverez l'intégralité du code source du plugin

```
<?php
/*
Plugin Name: Post by SMS
Plugin URI: http://www.orangepartner.com
Description: take the SMS forwarded in your mailbox and post it
Author: Francois Marx
Version: 0.3
Author URI: http://www.orangepartner.com
*/

add_action('admin_menu', 'sms_add_option_pages');
add_action('sms_check_post', 'sms_check_post');

// Setup defaults if options do not exist
add_option('sms_api_access_key', 'xxxxxx');
add_option('sms_delay', 15);

function sms_add_option_pages() {
    if (function_exists('add_options_page')) {
        add_options_page('Post by SMS', 'Post by SMS', 8, __FILE__, 'sms_options_page');
    }
}

function sms_options_page() {
    if (isset($_POST['info_update'])) {
        update_option('sms_api_access_key', (string)$_POST['sms_api_access_key']);
        update_option('sms_delay', (int)$_POST['sms_delay']);
        wp_clear_scheduled_hook('sms_check_post');
        wp_schedule_event(time(), 'sms_schedule', 'sms_check_post');
    }
}
?>
<div class=wrap>
<h2>Scheduled SMS</h2>
<form method="post" action="<?php echo $_SERVER["REQUEST_URI"]; ?>">

    <input type="hidden" name="info_update" id="info_update" value="true" />

    Number of minutes between SMS checks<br>
    <input name="sms_delay" type="text" size="10" value="<?php echo get_option('sms_delay'); ?>" /><br>

    API access key<br>
```



```

<input name="sms_api_access_key" type="text" size="20" value="<?php echo get_option('sms_api_access_key'); ?>"/

    <div class="submit">
        <input type="submit" name="info_update" value="<?php _e('Update options'); ?> &raquo;" />
    </div>

</form>
</div>
<?php
}

// this function check in mailbox for new SMS Post
// post the SMS as a blog article
// delete the email
function sms_check_post() {

    $api_access_key = get_option('sms_api_access_key');
    // list of URL to get content, list and delete emails
    $url_content = "http://mail.alpha.orange-api.net/mail/getMail.xml?id=$api_access_key";
    $url_list = "http://mail.alpha.orange-api.net/mail/getMailList.xml?id=$api_access_key";
    $url_delete = "http://mail.alpha.orange-api.net/mail/deleteMail.xml?id=$api_access_key";

    // get the list of all emails
    $response = file_get_contents($url_list);
    $xml_list = simplexml_load_string($response);

    // loop on all emails
    for ($n = count($xml_list->list->message); $n >= 1; $n--) {
        $m = $xml_list->list->message[$n-1]; // email number n
        if (substr($m->subject,0,8) == "Sms from") {
            // when the subject start by Sms from then we consider this email as a SMS and post it
            // Notice: all SMS are considered as blog article
            // get SMS content
            $response = file_get_contents($url_content.'&number='.$n);
            $xml_content = simplexml_load_string($response);
            $body = $xml_content->message->body;
            // remove the first words
            $post_title = substr($body, strpos($body, ' '), strpos($body, '#') - strpos($body, ' '));
            $post_title = $post_title.' (par SMS du '.substr($m->subject,9,11).')';
            $post_content = substr($body, strpos($body, '#')+1);
            $post_date = date('Y-m-d H:i:s', strtotime($xml_content->message->
>date)+get_option('gmt_offset')*3600);
            $post_status = 'publish';
            $post_data = compact('post_content', 'post_title', 'post_date', 'post_status');
            $post_ID = wp_insert_post($post_data);
            // delete emails
            $response = file_get_contents($url_delete.'&number='.$n);
        }
    }
}

function sms_more_reccurences() {
    // convert sms_delay from minutes to seconds
    return array(
        'sms_schedule' => array('interval' => 60*(int)get_option('sms_delay'), 'display' => 'SMS Schedule')
    );
}

add_filter('cron_schedules', 'sms_more_reccurences');
?>

```

IV-C - Le résultat final

SMS Blogging

Un blog utilisant WordPress

Mon premier SMS (par SMS du 336)

30 mai 2008

Vive le micro blogging

Publié dans [Non classé](#) | [Modifier](#) | [Aucun commentaire »](#)

micro (par SMS du 336)

30 mai 2008

test

Publié dans [Non classé](#) | [Modifier](#) | [Aucun commentaire »](#)

Pages

» [A propos](#)

Archives

» [mai 2008](#)

Catégories

» [Non classé \(2\)](#)

Blogoliste

- » [Development Blog](#)
- » [Documentation](#)
- » [Plugins](#)
- » [Suggest Ideas](#)
- » [Support Forum](#)

V - Conclusion

Dans cet article, nous avons présenté la construction d'un plugin de micro-blogging pour Wordpress. Le planificateur de Wordpress interroge régulièrement l'API mail pour récupérer les SMS routés vers votre boîte mail. **Ce tutorial est un exemple et il n'est pas conseillé d'implémenter un sondage régulier de la boîte mail (pour éviter de surcharger les serveurs).** Un plugin plus exploitable utiliserait le routage vers un script sur le site du blogueur pour directement ajouter les articles dans la base Wordpress.

1 : L'auteur de ce tutorial travaille pour Orange et a contribué aux développements des API.