Le Web-scraping appliqué au SEO

Retour au sommaire de la lettre

Domaine :	**Recherche**	Référencement
Niveau :	Pour tous	**Avancé**

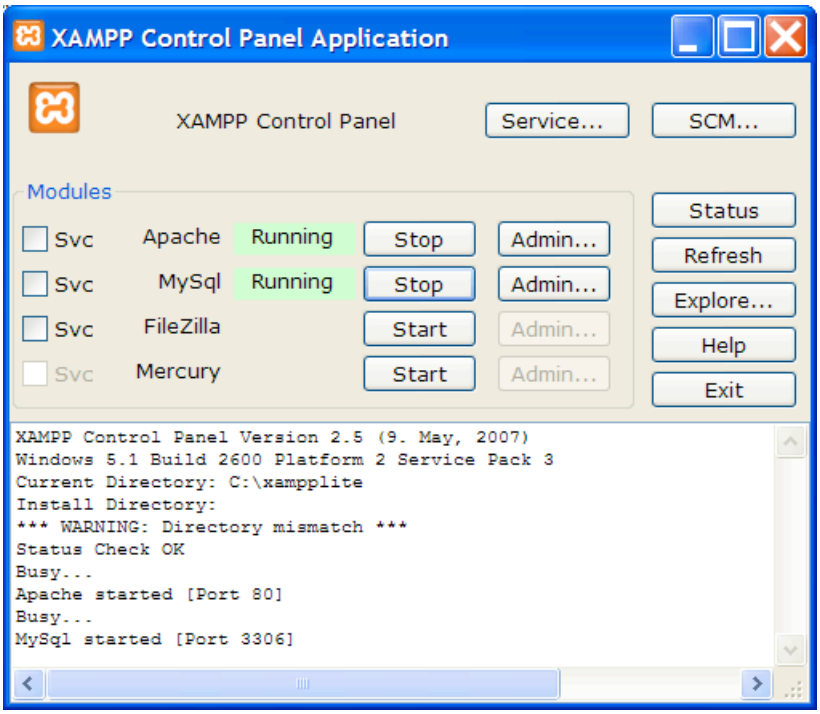
Nous en avons parlé le mois dernier, les techniques de Web-scraping permettent de récupérer, de "piocher", de traiter et d'archiver le contenu ou une partie d'une page web. S'ils sont des outils de veille indispensables, ils peuvent également être utilisés en référencement naturel pour apporter sur une page web du contenu venu d'autres sources, proposant ainsi aux moteurs de recherche du texte et un contenu éditorial souvent mis à jour. Si le système peut paraître complexe au départ pour les non-initiés, il s'avère rapidement beaucoup plus facile à mettre en œuvre qu'on ne l'imagine si vous suivez bien nos indications...

Nous en avons parlé le mois dernier : le principe du Web-scraping consiste à intégrer sur son site du contenu web et, par exemple, un flux RSS (interne ou externe) en utilisant un outil adapté. Imaginons que vous offriez à vos lecteurs une page « Actualités », ils apprécieront le fait de pouvoir lire une page régulièrement actualisée. Cela peut être un des flux RSS que propose votre site, des flux RSS provenant de sites tiers ou un mélange des deux. Vous pouvez alors utiliser Yahoo! Pipes afin de mixer et de filtrer les différentes sources et générer une sorte de « méga-flux ». La différence, en termes de référencement, est importante entre du code JavaScript, tel qu'il est utilisé lors de la création d'un Snippet, et du code PHP « pur et dur ». Si les internautes n'y verront que du feu, il n'en sera pas de même pour les moteurs de recherche qui ne lisent pas le code JavaScript mais se délecteront des liens "en clair" trouvés dans vos pages PHP. Et ces derniers seront d'autant plus intéressés par votre page puisqu'elle présentera un contenu sans cesse actualisé mais également « lisible ». Il sera ainsi possible de rendre quotidienne la mise à jour d'une page en s'appuyant sur du contenu externe...

Au final, nous terminerons notre exploration des possibilités offertes par le Web-scraping en analysant, maintenant, en quoi il constitue un élément indispensable au SEO.

Publier le contenu d'un flux RSS sur une page web

MagpieRSS peut se télécharger à partir de cette adresse : http://magpierss.sourceforge.net.
Notre test a porté sur la version 0.61.
L'archive est au format tar.gz. Il faut la décompresser avec une application comme WinRAR (http://www.rarlab.com).
Vous pouvez procéder à des essais en installant un serveur local comme XAMPP.
1. Accédez à cette adresse : http://www.apachefriends.org/fr/xampp-windows.html#1361.
2. Cliquez sur le lien *XAMPP version allégée*.
3. Enregistrez l'archive auto extractible dans un dossier sur votre disque dur.
4. Double-cliquez sur ce fichier puis indiquez le chemin de votre disque dur : c:\.
Une fois le processus d'extraction finalisé, vous allez obtenir cette arborescence : c:\xampplite.
5. Double-cliquez sur un fichier nommé setup_xampp.bat afin de lancer le processus de configuration.
Cette mention va apparaître : « *Have fun with ApacheFriends XAMPP Lite!* ».
6. Appuyez sur une touche pour continuer.
7. Double-cliquez maintenant sur un fichier nommé xampp-control.exe.
Une fenêtre de configuration appelée *XAMPP Control Panel Application* va s'afficher.
8. Cliquez maintenant sur les boutons *Start* placés à droite des mentions *Apache* et *MySql*.
La mention *Running* doit apparaître à chaque fois.



Vous pouvez forcer les composants Apache et MySQL à se lancer en tant que service :
9. Cochez la case *SVC* devant les mentions *Apache* et *MySql*.
10. Cliquez à chaque fois sur *OK* afin de valider le processus d'installation.

Testons maintenant le bon fonctionnement de votre serveur :
1. Dans la barre d'adresses de votre navigateur, saisissez ceci : http://localhost.
Cette adresse va apparaître : http://localhost/xampp/splash.php.
2. Cliquez sur le lien *Français*.
Un message va vous signaler que « *Vous venez d'installer XAMPP avec succès!* ».
Afin de désinstaller XAMPP Lite, Il suffit d'arrêter les deux services et de supprimer le répertoire complet.
3. Dans votre répertoire site01, créez maintenant un dossier nommée Magpierss (ou ce que vous voulez).
4. Copiez les fichiers ss_fetch.inc, rss_parser.inc, rss_cache.inc et rss_utils.inc.
5. Copiez également le dossier extlib qui contient, pour seul fichier, snoopy.class.inc.
La commande pour appeler le script se résume à ceci :

```
require(' magpierss/rss_fetch.inc');
```

Voici celle permettant d'insérer le script :

```
$rss = fetch_rss($url);
```

En prenant l'exemple du flux RSS du site d'Abondance :

```
$rss = fetch_rss('http://actu.abondance.com/rss.xml');
```

Il ne nous reste plus qu'à spécifier les éléments du flux RSS que nous allons utiliser :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Titre</title>
</head>
<body>
<?php
```

```
        require_once('magpierss/rss_fetch.inc');
        $url = 'http://actu.abondance.com/rss.xml';
        $rss = fetch_rss( $url );
        echo "Channel Title: " . $rss->channel['title'] . "<p>";
        echo "<ul>";
        foreach ($rss->items as $item) {
                $href = $item['link'];
                $title = $item['title'];
                echo "<li><a href=$href>$title</a></li>";
        }
        echo "</ul>";
?>
</body>
</html>
```

La seule contrainte est que la page web doit posséder une extension PHP…
MagpieRSS analyse un flux RSS en inspectant ces quatre champs :
- channel : contient les métadonnées du flux RSS placées entre la balises racines <rdf:RDF> ou <rss> ;
- items : encadre les données correspondants à chacun des éléments qui composent le flux RSS ;
- image et textinput : fonctionnent comme des espaces associés qui vont extraire l'ensemble des données incluses entre les deux balises correspondantes.

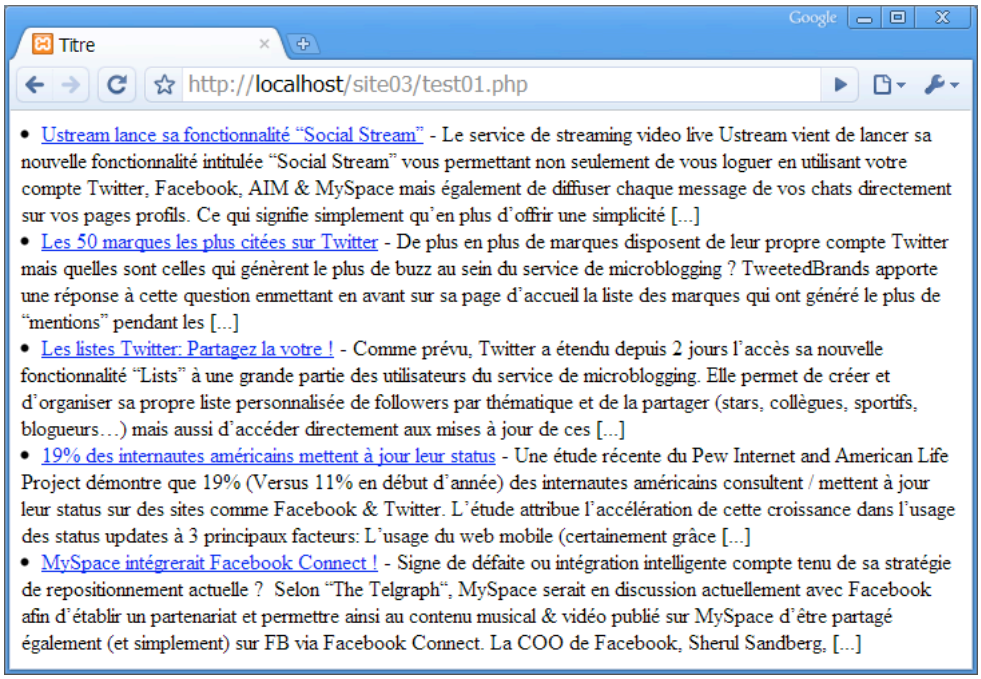
Bien entendu, il est possible de personnaliser la façon dont vont s'afficher les extraits web. Par exemple, voici comment afficher les 5 derniers articles du site Abondance :

```
<?php
require_once('magpierss/rss_fetch.inc');
$url = 'http://actu.abondance.com/rss.xml';
$rss = fetch_rss($url);
if ($rss) {
                $items = array_slice($rss->items, 0, 5);
        foreach ($items as $item) {
```

```
                echo "<p><a href=\"" .$item['link']. "\">"
                .$item['title']. "</a></p>";
                }
}
?>
```

Le principe de ce script est un peu différent du précédent puisqu'il utilise la fonction array(). Cette dernière permet de créer des tableaux dynamiques. Array_slice retourne une série d'éléments extraits du tableau array commençant à l'offset défini et représentant la longueur voulue des éléments.
On peut maintenant vouloir afficher le contenu complet d'un flux RSS. Cette fois-ci, nous utiliserons un autre flux RSS :

```
<?php
require_once 'magpierss/rss_fetch.inc';
$rss = fetch_rss('http://feeds.feedburner.com/Mashablefrance');
$items = array_slice($rss->items, 0, 5);
 foreach ($items as $item )
 {
        echo '<li><a href="'.$item['link'].'">'.$item['title'].'</a> -
'.$item['description'].'</li>';
        }
?>
```



En fonction de la structure des flux, vous pouvez aussi afficher les images incluses dans les éléments.

Rien ne vous empêche d'ajouter, par exemple, la date de chacun des éléments :

```
<?php
require_once 'magpierss/rss_fetch.inc';
$rss = fetch_rss('http://feeds.feedburner.com/Mashablefrance');
$items = array_slice($rss->items, 0, 5);
 foreach ($items as $item )
        {
                $titre = $item["title"];
$lien = $item["link"];
$date = date("d/m/y",strtotime($item["pubdate"]));
$result .= "$date : ";
$result .= "<a href=\"".$lien."\" title=\"".$titre."\" target=\"_blank\">".$titre."</a><br>\n";
}
echo $result;
?>
```

La date utilisée pour les flux RSS étant au format anglais, il faut formater cette information en utilisant une fonction PHP nommée strtotime().
Vous pouvez également utiliser une feuille de style externe :

```
<title>Titre de la page </title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<?php
...
?>
```

Par défaut, la liste des derniers articles est actualisée toute les heures. MagpieRSS créé automatiquement un répertoire nommé Cache au niveau supérieur de là où sont exécutés les scripts. Vous pouvez le constater en ouvrant l'arborescence de votre site local. Si vous souhaitez modifier ce délai, ouvrez le fichier rss_cache.inc puis éditez cette ligne :

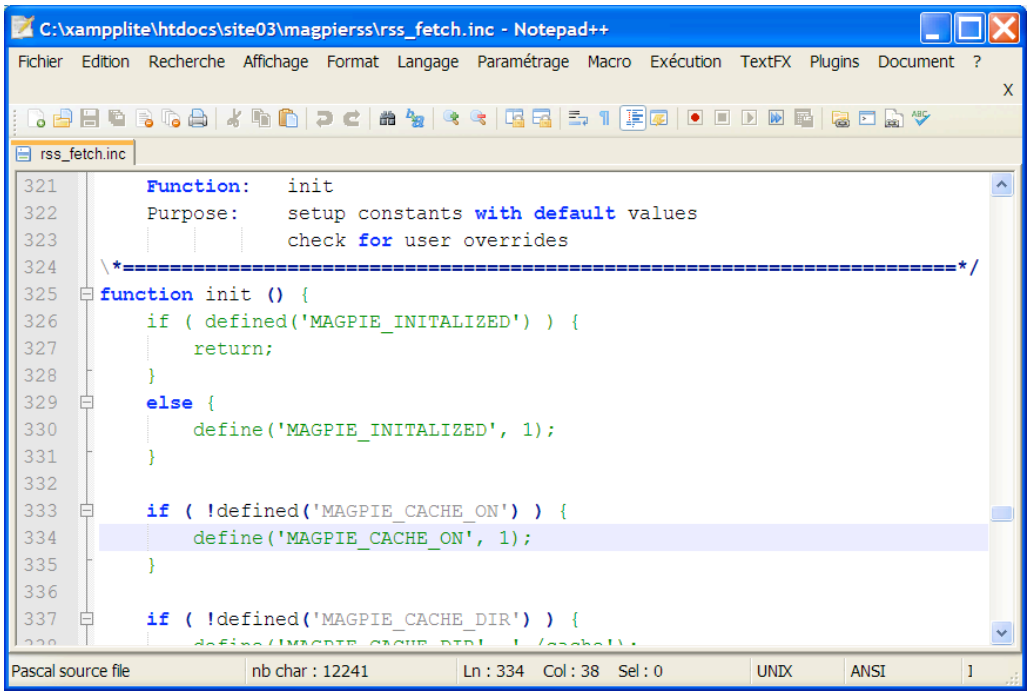
```
var $MAX_AGE = 3600;  // when are files stale, default one hour
```

Afin de modifier facilement ce type de fichier, utilisez un utilitaire comme Notepad++ (http://notepad-plus.sourceforge.net/fr/site.htm).
Il est aussi possible de désactiver le cache de MagpieRSS en utilisant cette procédure :

1. Ouvrez le fichier rss_fetch.inc.
2. Accédez à cette ligne :

```
if ( !defined('MAGPIE_CACHE_ON') ) {define('MAGPIE_CACHE_ON', 1);
```

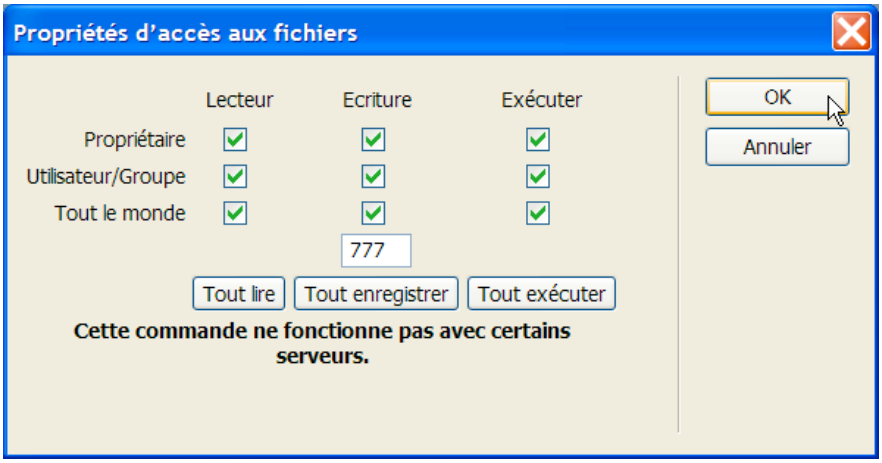
Modifiez le 1 par un 0.



Si vous voulez conserver cette fonctionnalité, vous devrez créer, sur le serveur distant, un répertoire nommé cache puis lui attribuez un Chmod 777.

Chmod (abréviation de change mode) est une commande permettant de changer les permissions d'accès sur un fichier ou un répertoire. Sa mise en œuvre est simple :

1. Connectez-vous à votre espace FTP.
2. Cliquez avec le bouton droit de la souris sur un des fichiers listés ou un des répertoires.
3. Cliquez sur le sous-menu *Attributes (CHMOD)* ou *Lire les informations* ou encore *Permissions de fichier*.
4. Changez, si nécessaire, les propriétés.



L'autre souci qui peut se poser est que, pour fonctionner, MagpieRSS nécessite que la page soit en PHP. Oui ! Mais alors comment faire si l'ensemble des pages de votre site sont en HTML ? Une solution possible consiste éditer le fichier .htaccess de votre site et d'utiliser une redirection 301.
Voici un exemple de déclaration :

```
redirect 301 /index.html http://www.exemple.fr/index.php
```

Une seconde solution est suggérée sur le site de Webmonkey (http://www.webmonkey.com) : Créez un nouveau fichier nommé news.php et placez-le dans le répertoire de MagpieRSS.
Il peut ressembler à ceci :

```
<?php
header("Content-type:text/javascript");
include('./rss_fetch.inc');
$url = "http://actu.abondance.com/rss.xml";
$rss = fetch_rss($url);
if ($rss) {
        $items = array_slice($rss->items, 0, 5);
        $news_string = "";
        foreach ($items as $item) {
                $news_string = $news_string. "<p><a href='"
                .$item['link']. "'>" .$item['title'].
                "</a></p>";
        }
}
echo "document.write(\"$news_string\")";
?>
```

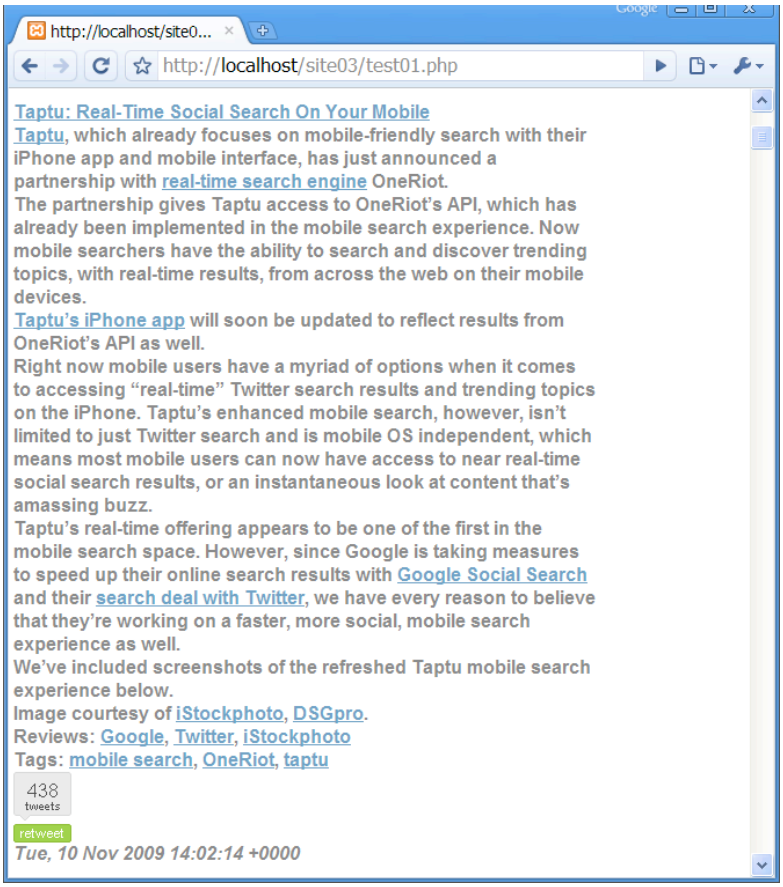
Créez ou modifiez maintenant vos pages HTML puis insérez, à l'endroit voulu, cette ligne de code :

```
<script type="text/javascript" src="./magpierss/news.php"></script>
```

Le principe consiste à envoyer l'intégralité du contenu généré dans une variable chaîne nommée $news_string. Cette dernière retourne, sous la forme d'une chaîne concaténée, le contenu de document.write().
Notez que cette solution ne fonctionne qu'avec les flux RSS au format XML.

Il faut signaler qu'une fonction native de PHP 5.0 vous permet de « parser » un flux RSS : SimpleXML. Voici un exemple de code :

```
<?
$rss_fichier = "http://feeds.feedburner.com/mashable";
$rss_flux = simplexml_load_file( $rss_fichier );
?>
<html>
<body>
<h3><?= $rss_flux->channel->title?></h3>
<?
foreach( $rss_flux->channel->item as $item ) {
      print "<b><a href=$item->link>$item->title</a></b><br>";
      print "$item->description<br>";
      print "<i>$item->pubDate</i><br><br>";
}
?>
```



Le manuel complet de SimpleXML est visible à cette adresse : http://php.net/manual/fr/book.simplexml.php. De notre point de vue, ces deux outils sont très similaires mais on peut préférer la souplesse d'utilisation de MagpieRSS.

Publier du contenu web sur votre site

DOM (*Document Object Model*) est un schéma de langage de programmation du W3C qui permet aux programmes et aux scripts d'accéder et de modifier dynamiquement le contenu, la structure et le style de documents XML ou HTML.

Simple HTML DOM est un « Parser » écrit en PHP 5 servant à manipuler du contenu HTML (même non valide) en utilisant les spécifications DOM. Il offre une simplicité très appréciable puisque vous n'avez pas à vous servir des expressions rationnelles pour extraire du contenu web.

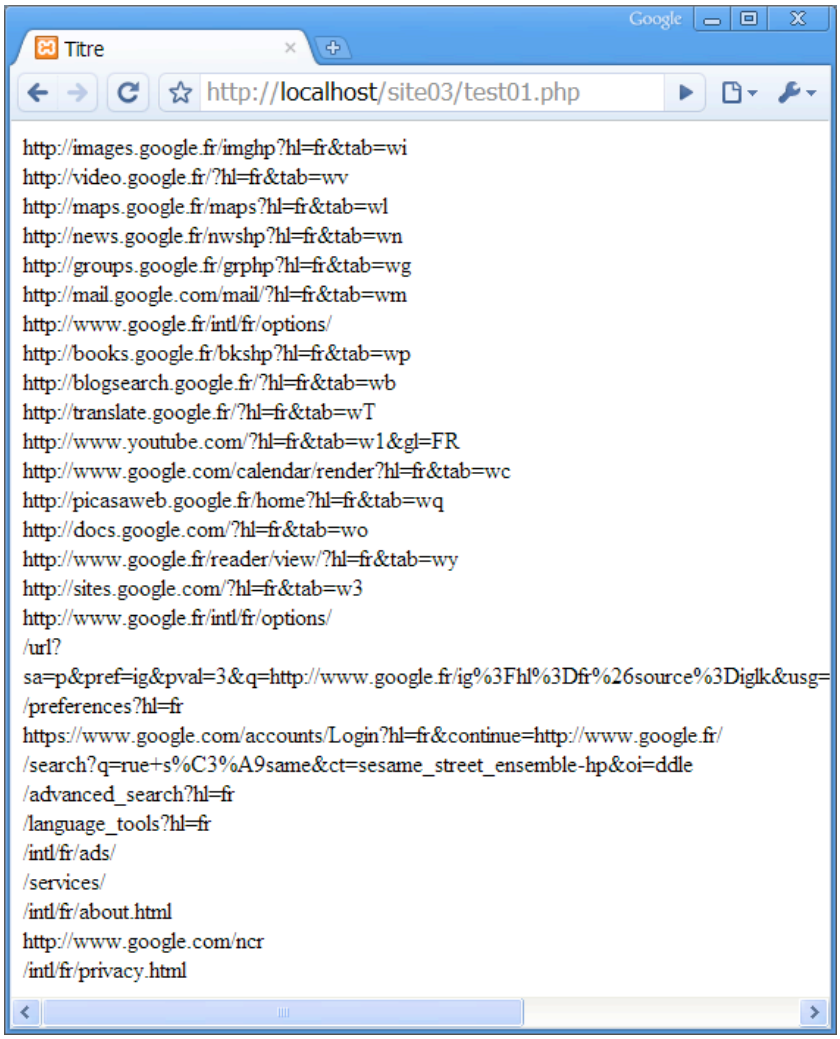
La différence avec l'outil précédent est que l'on n'utilise plus des flux RSS mais, bel et bien, des pages web dont nous allons simplement extraire les morceaux qui nous intéressent.
En cas de problème sur un serveur distant, notez que, dans le fichier PHP.ini, la commande allow_url_fopen doit être sur la valeur *TRUE*.

1. Téléchargez cette librairie à partir de cette adresse : http://sourceforge.net/projects/simplehtmldom/files/.
2. Décompressez l'archive ZIP puis copiez sur votre serveur un fichier nommé simple_html_dom.php.
3. Créez un nouveau document PHP dans lequel vous allez copier ces lignes :

```
<?php
include('simple_html_dom.php');
$html = file_get_html('http://www.google.fr/');
foreach($html->find('a') as $e)
```

```
   echo $e->href . '<br>';
?>
```

Attention au fait que le chemin d'appel vers le fichier PHP doit être éventuellement modifié.
4. Enregistrez cette page puis ouvrez-la dans votre navigateur.
L'ensemble des liens présents sur la page d'accueil de Google s'afficheront.



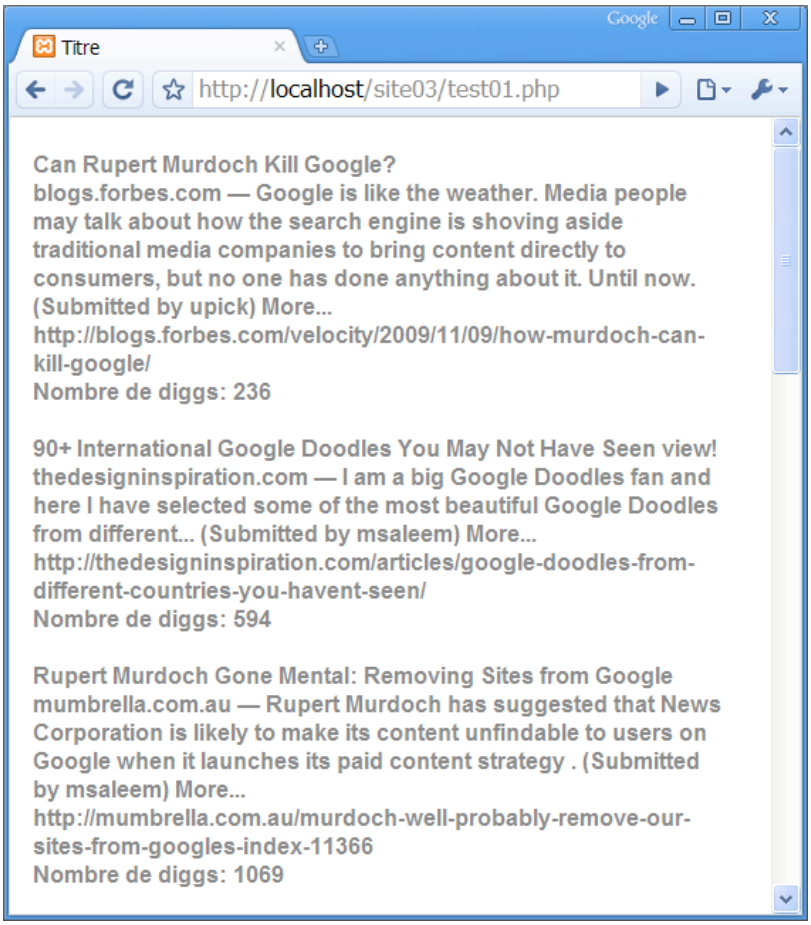
Voici un autre exemple vous permettant d'extraire le texte de n'importe quelle page web :

```
<?php
include('simple_html_dom.php');
$html = file_get_html('http://actu.abondance.com/2009/11/seoscope-nouveau-logiciel-de-reporting.html');
echo $html->plaintext;
?>
```

Dans l'archive qui est téléchargée, il existe un exemple intéressant de Web-scraping permettant de récupérer une page Digg. Voici le code légèrement remanié :

```
<link href="style.css" rel="stylesheet" type="text/css">
<table width="50%">
  <tr class="content">
    <th class="block" scope="col">
          <p align="left">
          <?php
include_once('simple_html_dom.php');
```

```
function scraping_digg() {
   $html = file_get_html('http://digg.com/search?s=google');
   foreach($html->find('div.news-summary') as $article) {
      $item['title'] = trim($article->find('h3', 0)->plaintext);
      $item['content'] = trim($article->find('p', 0)->plaintext);
      $item['diggs'] = trim($article->find('li a strong', 0)->plaintext);
      $item['link'] = trim($article->find('a', 0)->href);
 $test [] = $item;
   }
   $html->clear();
   unset($html);
   return $test;
}
ini_set('user_agent', 'Mon-Application/2.5');
$test = scraping_digg();
foreach($test as $z) {
          echo '</br>';
         echo '<ul>';
         echo '<li>'.$z['title'].'</li>';
   echo '<li>'.$z['content'].'</li>';
   echo '<li>'.$z['link'].'</li>';
   echo '<li>Nombre de diggs: '.$z['diggs'].'</li>';
   echo '</ul>';
}
?>
   </p></th>
  </tr>
</table>
```



La commande include_once sert à évaluer la page web durant l'exécution du script. La fonction include_once est utilisée de préférence à la fonction include lorsque le fichier doit être évalué plusieurs fois au cours de l'exécution du script.

On définit ensuite une fonction appelée scraping_digg.
On utilise une variable nommée « html ». Une variable doit obligatoirement être précédée du caractère dollar ($).

La commande foreach passe en revue la page. À chaque itération, la valeur de l'élément courant est assignée à $article et le pointeur interne est avancé d'un élément.
On récupère chaque bloc d'actualité signalé par l'ID de classe « news-summary ».
Pour chaque bloc récupéré, on extrait le titre (title, encadré par des balises H3), son contenu (content, signalé par les balises <p>) et le texte d'introduction (diggs, signalé par la balise <strong id>) ainsi que le lien (link).

Vous pouvez mettre en évidence la structure de la page en affichant son code source.
La commande Trim permet de supprimer les espaces en début et en fin de chaîne.
Après avoir vidé le cache mémoire, la fonction Unset supprime la variable « html ».

La fonction ini_set sert à modifier la valeur de l'agent utilisateur afin de répondre aux spécifications édictées par l'API de Digg.

$test retourne la structure de l'arbre DOM dans une chaîne.

Il ne reste plus qu'à afficher chacun des éléments en les séparant par des puces (<ul>, <li>).
Pour une explication complète de l'API de Simple HTML Dom, consultez cette page : http://simplehtmldom.sourceforge.net/manual.htm.

Tel quel, cela paraît complexe mais, avec un peu de pratique, on arrive rapidement à récupérer toute sorte de contenu web. Et, en cas de difficulté, n'importe quel développeur PHP sera en mesure de vous donner un coup de main…

Le contenu est roi !

Il existe d'autres plateformes permettant de faire du Web-Scraping et on peut citer ces quelques applications :
- Zend_Dom_Query (http://framework.zend.com/manual/fr/zend.dom.query.html) ;
- phpQuery (http://code.google.com/p/phpquery/) ;
- **DomQuery** (http://www.extjs.com/learn/Tutorial:DomQuery_v1.1_Basics) ;
- **Curl** (http://curl.haxx.se/docs/comparison-table.html).

Cette liste n'est sûrement pas exhaustive… Mais attention de vous rappeler que, très souvent, les autres librairies qui sont à votre disposition ne génèrent pas un contenu « crawlable » par les moteurs de recherche. Au final, il est possible de créer, de toutes pièces, des pages web présentant des données constamment actualisées, de faciliter leur distribution à l'intérieur d'un site et de travailler sur les informations dynamiques qui seront, soit récupérées, soit automatiquement générées. Et c'est une excellente manière d'optimiser le référencement de son site auprès des moteurs !

Jean-Noël Anderruthy, ***webmaster spécialisé dans les technologies Google.***

Réagissez à cet article sur le blog des abonnés d'Abondance :
http://abonnes.abondance.com/blogpro/2009/11/le-web-scraping-applique-au-seo.html