

WordPress en multisite, Apache et redirection infinie

Si vous avez un site WordPress tournant sous Apache et équipé d'un multisite, vous avez sans doute déjà vu ce genre d'erreur dans le fichier `/var/log/apache2/error.log` :

```
Request exceeded the limit of 10 internal redirects due to
probable configuration error. Use 'LimitInternalRecursion' to
increase the limit if necessary. Use 'LogLevel debug' to get a
backtrace., referer: http://....
```

Ce problème semble lié à une redirection infinie générée par la configuration par défaut de WordPress.

Il faut remplacer les lignes automatiques générées dans le fichier `.htaccess` par celles-ci :

```
# BEGIN WordPress

RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]

# uploaded files
RewriteRule ^([_0-9a-zA-Z-]+)/?files/(.+) wp-includes/ms-
files.php?file=$2 [L]

# add a trailing slash to /wp-admin
RewriteRule ^([_0-9a-zA-Z-]+)/?wp-admin$ $1wp-admin/ [R=301,L]

RewriteCond %{REQUEST_FILENAME} -f [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^ - [L]
RewriteRule ^([_0-9a-zA-Z-]+)/?(wp-(content|admin|includes).*)
$2 [L]
RewriteRule ^([_0-9a-zA-Z-]+)/?(.*\.php)$ $2 [L]
RewriteRule . index.php [L]
# END WordPress
```

Source

: <https://gist.github.com/JustThomas/141ebe0764d43188d4f2>

Non ce blog n'est pas mort

...et pourtant l'envie de l'abandonner m'a plusieurs fois traversé l'esprit.

Une année chargée en émotions, en déceptions et en petits bonheurs.

Inverser la vapeur, essayer de me poser franchement, et de comprendre ce qui sous-tend mon comportement actuel : l'objectif de 2014.

Espérons que ce blog survivra (sans doute, j'ai plein d'idées d'articles en tête, juste une violente fracture de la motivation).

Gérer le drag&drop et le téléchargement de fichiers en Javascript

Depuis quelques temps, il existe au sein de nos chers navigateurs, l'API FileReader, qui permet, comme le résume bien cet article de [MDN](#) :

- de gérer les fichiers sélectionnés via les champs de

formulaire "fichier"

- de supporter le glisser/déposer de fichiers dans le navigateur

Les extraits de code qui suivent sont fortement inspirés de [l'article de Maxime Chaillou sur le drag&drop](#).

On crée tout d'abord une zone qui servira de réceptable à nos fichiers :

```
[html]<div id="dropfile">Drop your file here to deduplicate SMS</div>[/html]
```

On gère ensuite les évènements sur cet élément :

```
[javascript]
```

```
//Quand la souris, maintenue enfoncée, entre dans la zone, on la colore, pour indiquer l'interaction
```

```
$(document).on('dragenter', '#dropfile', function() {
```

```
$(this).css('border', '3px dashed red');
```

```
return false;
```

```
});
```

```
//Une fois que la souris est rentrée dans la zone, on actualise régulièrement l'état de la zone
```

```
$(document).on('dragover', '#dropfile', function(e){
```

```
e.preventDefault();
```

```
e.stopPropagation();
```

```
$(this).css('border', '3px dashed red');
```

```
return false;
```

```
});
```

```
//Si la souris, toujours enfoncée, ressort, on indique la fin de l'interaction
```

```
$(document).on('dragleave', '#dropfile', function(e) {
```

```
e.preventDefault();
```

```
e.stopPropagation();
```

```
$(this).css('border', '3px dashed #BBBBBB');
```

```
return false;
```

```
});
```

```
[/javascript]
```

Il faut ensuite gérer l'upload à proprement parler :

```
[javascript]
$(document).on('drop', '#dropfile', function(e) {
//Si le transfert est possible
if(e.originalEvent.dataTransfer){
//S'il y au moins un fichier
if(e.originalEvent.dataTransfer.files.length) {
//On bloque l'ouverture du fichier dans le navigateur, ce qui
est l'action par défaut
e.preventDefault();
//On bloque ensuite la propagation de l'évènement "dépot de
fichiers" pour éviter de déclencher toute action sur les
éléments parents
e.stopPropagation();
$(this).css('border', '3px dashed green');
//On lance l'upload à proprement parler
upload(e.originalEvent.dataTransfer.files);
}
}
else {
$(this).css('border', '3px dashed #BBBBBB');
}
return false;
});
[/javascript]
```

La fonction upload() alors utilisée va nous permettre d'envoyer le contenu du fichier déposé à une autre fonction (par exemple pour l'envoyer à un serveur Web) :

```
[javascript]
function upload(files) {
var f = files[0] ;
var reader = new FileReader();

//La fonction handleReaderLoad est ici associée à l'évènement
"onload", pour qu'elle se déclenche une fois le chargement du
fichier terminé
```

```
reader.onload = handleReaderLoad;
```

```
//Pour les besoins de cette démonstration, on utilise ici  
readAsText() qui va lire le fichier sous forme de chaîne de  
caractères
```

```
//Mais on utilise plutôt readAsBinary, pour l'envoi à un  
serveur Web
```

```
reader.readAsText(f);
```

```
}
```

```
[/javascript]
```

La dernière fonction va quant à elle gérer le fichier à proprement parler (l'envoyer via Ajax, ou, comme ici, traiter la chaîne de caractères comme un XML et en dédoubler le contenu) :

```
[javascript]
```

```
function handleReaderLoad(evt) {
```

```
var xml = evt.target.result
```

```
deduplicateSMS($.parseXML(xml));
```

```
}
```

```
[/javascript]
```

Pour cet exemple (qui vise à supprimer les SMS en double dans une sauvegarde XML), on utilise également le plugin jQuery FileSaver.js, qui permet de lancer le téléchargement de fichiers dans un navigateur depuis Javascript :

```
[javascript]
```

```
function deduplicateSMS(xmlSource) {
```

```
var ids = new Array();
```

```
var duplicates= 0;
```

```
var total= 0;
```

```
//On explore le XML et on vérifie si la clé de chaque noeud  
existe déjà dans le tableau "ids"
```

```
$(xmlSource).find('allsms sms').each(function(index,e) {
```

```
if
```

```
(ids.indexOf($(this).attr('address')+$(this).attr('date'))==-1
```

```
) {
```

```

total+=1;
ids.push($(this).attr('address')+$(this).attr('date'));
} else {
//Si c'est le cas, on le supprime
duplicates+=1;
$(this).remove();
}
});
$(xmlSource).find('allsms').attr('count',total);
//On convertit ensuite ce XML en chaîne de caractères via
XMLToString, puis en "Blob" gérable par FileSaver.js
var blob = new Blob([XMLToString(xmlSource)], {type:
"text/xml;charset=utf-8"});
//On lance ensuite le téléchargement
saveAs(blob, "sms.xml");
}
[/javascript]

```

La méthode XMLToString utilisée ici est très simple et permet juste de convertir un arbre XML en chaîne :

```

[javascript]
/*
Source
:
http://www.dotnet-tricks.com/Tutorial/javascript/Y7Q9130612-Convert-string-to-xml-and-xml-to-string-using-javascript.html*/
function XMLToString(oXML)
{
//code for IE
if (window.ActiveXObject) {
var oString = oXML.xml; return oString;
}
// code for Chrome, Safari, Firefox, Opera, etc.
else {

return (new XMLSerializer()).serializeToString(oXML);
}
}
[/javascript]

```

Vous pouvez retrouver cet exemple sur Github :
<https://github.com/samy-r/sms-super-backup-deduplication>

Une démo est disponible sur mon espace de test :
<http://lahaut.info/demos/sms-super-backup-deduplication/>

Traduction française actualisée du thème “Adelle” (BluChic)

Ayant récemment eu besoin d'installer le thème Adelle (édité par BluChic), j'ai heureusement trouvé une [traduction française](#), mais celle-ci n'était pas totalement à jour : problème résolu désormais !

[Traduction française actualisée du thème Adelle](#)

(merci à [WPAddict](#) pour la traduction initiale)

A donner

Comme en 2011, je fais du ménage et j'en profite pour donner ce qui ne me sert plus (ou qui ne m'a jamais servi). J'enlèverais les éléments de cette page au fur et à mesure qu'ils auront été donnés.

Rappel : je suis sur **Lyon**, donc remise en main propre ou envoi par la Poste ☐

e-Tecely : reportage Tribune de Lyon

L'hebdomadaire "Tribune de Lyon" a réalisé un petit reportage vidéo sur le test de e-Tecely auquel j'ai participé :

[TCL. Samy Rabih : "e-Tecely est une avancée mais je reste sur mes gardes"](#)

(par [Tribune de Lyon](#) sur [Vimeo](#))

Passer de BlogSpirit à WordPress : nouvelle version de l'importeur !

Suite à un [commentaire de Capripot](#), j'ai découvert son travail sur l'importeur BlogSpirit vers WordPress que j'avais moi-même repris à l'époque. Il a notamment corrigé les problèmes suivants :

- * importation des article se basant sur la page complète de la liste des articles par catégorie
- * « / » en trop à la fin de l'adresse ce qui empêchait l'import des articles d'une catégorie
- * bug de version lors de l'import dans wordpress)
- * script un peu plus bavard
- * choix du nom au départ ou écriture du nom du blog dans NOM_SITE.txt

- * dialog.rb renommé en import.rb pour plus de clareté
- * non blocage du script à la fin
- * correction de l'import des commentaires
- * ajout d'un fichier LISEZ-MOI

[Télécharger nouvelle version importeur BlogSpirit vers WordPress](#)

Son site : <http://www.capripot.com/>

Et vous pouvez retrouver l'outil pour passer d'OverBlog à WordPress sur la [page dédiée](#)

Installation de WordPress en multi domaines

Dès le début de l'année, [Automattic](#) (éditeur de WordPress), avait annoncé que la prochaine version de WordPress (3.0) générerait de manière native le multi-domaines (anciennement assuré par WordPress MU, qui serait donc inclut directement dans WordPress 3).

La dernière version disponible (au 20 juin 2010) propose effectivement ce genre de fonctionnalités.

Cependant, de base, WordPress 3 ne propose du multi-sites (terme générique pour désigner ces fonctions) que selon les schémas suivants :

- X sous-domaines d'un nom de domaine : toto.domaine.com, tata.domaine.com, etc
- X sous répertoires d'un nom de domaine : domaine.com/toto, domaine.com/tata, etc.

Cela répond à certaines problématiques, notamment pour de la mutualisation à grande échelle, mais dans des cas bien précis comme le mien (hébergement d'un nombre limité de blogs sur un même serveur dédié), il peut être intéressant de **pouvoir faire tourner plusieurs blogs ayant des noms de domaine séparés sur la même base de code**, notamment pour faciliter les mises à jour.

WordPress 3 ne le gère pas de manière "intuitive" mais nous allons voir ensemble qu'il est possible tout de même, au prix de quelques manipulations, de le faire fonctionner parfaitement dans cette configuration.

Pré-requis :

- [les fichiers d'installation de WordPress 3](#)
- les fichiers pour la mutualisation : [domain_mapping.php](#) et [sunrise.php](#)
- un nom de domaine "principal" pointant vers votre hébergement (dans notre exemple, [www.geekcherchegeekette.fr](#))
- un ou des noms de domaine pointant vers le même hébergement

1/ Installation de WordPress 3

Celle-ci est identique à celle des versions précédentes, mis à part que vous pouvez maintenant *choisir le mot de passe de l'utilisateur principal dès l'installation*.

2/ Activation du mode multi-sites

Il vous suffit de rajouter les lignes :

```
define('WP_ALLOW_MULTISITE', true);  
define('MULTISITE', true);
```

juste après

```
define('WP_DEBUG', false);
```

dans votre fichier wp-config.php (à la racine de votre

installation de WordPress)

Si vous vous reconnectez sur votre blog "principal", vous verrez alors apparaître un menu **Outils / Réseau** sur lequel il vous faudra cliquer :



Cette page permet de créer un réseau de sites (en d'autres termes, de mutualiser les fichiers de WordPress entre plusieurs sites).



Vous remarquez qu'il n'y a aucune mention du cas qui nous intéresse : ce n'est pas grave, nous allons contourner ce problème :

- Sélectionnez l'option **Sous-dossiers**
- Remplissez les cases **Nom du réseau** et **Adresse de contact de l'administrateur**
- Validez

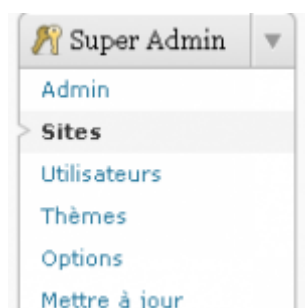
La page suivante vous indique les modifications à effectuer sur vos fichiers **wp-config.php** et **.htaccess** : pensez à en faire une sauvegarde avant !

Pensez également à créer un sous-dossier **blogs.dir** dans votre dossier **wp-content**.



Cliquez ensuite sur **Se connecter** en bas de la page et reconnectez-vous avec vos identifiants du blog qu'on vient de créer.

Un menu **Super Admin** est désormais présent en haut de la page :



3/ Paramétrage du multi domaines

Entrent maintenant en jeu le plugin `domain_mapping` : déposez le fichier **domain_mapping.php** récupéré au début dans un sous-dossier **mu-plugins** que vous aurez créé dans **/wp-content**.

Déposez ensuite le fichier **sunrise.php** directement dans le dossier **/wp-content**.

Rajoutez ensuite la ligne

```
define( 'SUNRISE', 'on' );
```

dans `wp-config.php` juste après les lignes que vous venez d'y rajouter.

Ces deux manipulations ont pour effet d'activer le menu **Domain Mapping** dans **Super Admin**. Dans cet écran, précisez l'IP du serveur hébergeant votre/vos site(s), cochez les 2e et 3e cases et validez :



4/ Création effective des domaines

Avant tout, rendez vous dans **Super Admin / Options** et choisissez **Français** comme langue par défaut.

Ensuite, on va se rendre dans **Super Admin / Sites** : comme indiqué plus haut, WordPress 3 ne gérant pas de manière native les multi domaines, nous allons devoir ruser.

Dans le formulaire **Ajouter un site**, saisissez le nom du site que vous voulez créer (sans les www et extension, en fait, peu importe ce que vous saisissez ici)

Complétez les champs **Titre du site** et **Adresse de contact de l'administrateur** (vous remarquerez que j'ai utilisé l'adresse principale, pour pouvoir me reconnecter facilement sur le nouveau blog).

Validez : notre nouveau blog est maintenant accessible depuis www.domaineprincipal.com/nomdusite.

Rendez-vous alors sur ce blog (via le bouton **Afficher** sous sa ligne dans la page des sites).

Il faut maintenant donner son URL propre à ce blog : allez sur la page d'administration du sous-blog (de la forme www.domaineprincipal.com/nomdusite/wp-admin).

Vous devriez normalement être connecté automatiquement (d'où l'intérêt d'utiliser le même mail pour l'utilisateur principal).


Dans **Super Admin / Domaines / New Domain**, saisissez le "Site

ID" (à récupérer dans la page **Sites** du blog de base) et son domaine associé : **votre sous-blog est désormais accessible par sa propre URL !**

Limitations :

- les thèmes doivent être activés par un utilisateur "Super Admin" et ne peuvent être modifiés de manière unilatérale : en clair, si je modifie un thème, il est modifié pour tous les blogs qui l'utilisent
- dans cette configuration, tous les blogs utilisent la même base de données (je prépare un second article plus spécifique pour une solution permettant de découpler les blogs à ce niveau là).

OMG ! 666 spams !



Brouillons
Spam (666)

(oui c'était l'article inutile du mois, ou même de l'année)

20 ans : comment faire une feuille de choux sans moyens

Loin de moi l'idée de juger le contenu d'un magazine tel que 20 ans nouvelle formule, mais vu les moyens investis (cf [l'article de 20 Minutes](#)), ca ne m'étonnerait pas que le

contenu soit proche du néant absolu ...

A noter que le PDG de FT Medias, qui a racheté 20 ans, est à l'origine de la série "Paris 16e"...

CQFD (sic)