

Pmwiki est conçu pour être utilisable comme un outil de maintenance de site web, et fréquemment les administrateurs voudront être capable de fournir un accès aux pages de [Pm Wiki](#) sans avoir "pmwiki.php" qui apparaît dans la barre d'adresse de la fenêtre du navigateur de l'utilisateur. Cette page explique les différentes manières de faire apparaître votre installation Pm Wiki sous une URL autre que le script pmwiki.php lui même. Pour les exemples suivants on considère qu'une URL telle que <http://www.example.com/wiki> sera configurée pour accéder à pmwiki.php sur le site www.example.com.

Il y a quatre façons principales d'y parvenir : l'aliasing, la fabrication d'un script "wiki", renommer le script pmwiki, ou l'utilisation du mod_rewrite.

Sommaire ([cacher](#))

1. [Aliasing](#)
2. [Faire un script "wiki"](#)
3. [Renommer le fichier pmwiki.php](#)
4. [Utiliser mod_rewrite](#)
5. [Avoir des URLs plus propre sans PmWiki](#)

1. Aliasing

Tout d'abord, si vous avez accès au fichier de configuration du serveur web, alors en ajoutant simplement la ligne

```
Alias /wiki /path/to/pmwiki.php
```

à ce fichier, l'alias sera effectué pour vous. Dans certain cas (par exemple Apache 2.0) vous serez contraint d'utiliser le chemin absolu jusqu'à pmwiki.php, par ex.

```
Alias /wiki /home/account/www/pmwiki.php
```

même si vous avez déjà défini le documentroot comme étant /home/account/www. Vous aurez alors également besoin de paramétrer les variables \$PubDirUrl et [\\$Uploads Url Fmt?](#) du fichier config.php afin qu'elles pointent sur les URL correctes pour les dossiers pub/ et uploads/.

2. Faire un script "wiki"

Si vous n'avez pas accès au fichier de configuration, alors un peu de redirection s'impose. D'abord créer un fichier (pas un répertoire) nommé "wiki" à l'endroit où le serveur web devrait normalement accéder à l'URL "/wiki". Souvent il s'agit du dossier DocumentRoot du serveur web mais cela dépend vraiment de la configuration de votre site. Placez le code PHP suivant dans le fichier wiki:

```
<?php chdir("/path/to/pmwikidir"); include("pmwiki.php"); ?>
```

Dans le script remplacez `/path/to/pmwikidir` avec le nom du dossier contenant `pmwiki.php`. Puis, ajoutez les lignes suivantes au fichier `.htaccess` dans le même dossier contenant le fichier `wiki` (créez le fichier `.htaccess` si il n'existe pas):

```
<Files wiki> SetHandler application/x-httpd-php </Files>
```

Cela indique au serveur de traiter le fichier `wiki` comme un script PHP même si celui-ci ne se termine pas par `.php`. Maintenant, les requêtes à votre fichier `wiki` provoqueront le déplacement du serveur au dossier contenant `pmwiki.php` et l'exécution de ce dernier depuis là. Assurez-vous que la variable `$ScriptUrl` de `config.php` pointe bien sur le fichier `wiki` et non pas sur le dossier contenant `pmwiki.php`.

3. Renommer le fichier `pmwiki.php`

Une troisième méthode est juste de renommer "`pmwiki.php`" en un nom plus approprié, par exemple "`do.php`". Si vous envisagez l'utilisation de l'analyseur de fichier log de serveur [webalizer](#), vous devriez éviter de renommer "`pmwiki.php`" en "`index.php`". Webalizer "tronque" les urls après la partie '`index.`', détruisant ainsi effectivement toutes les références aux pages. Par exemple: l'accès à '`/index.php/Main/Abc`' et '`/index.php/Main/Xyz`' comptent deux fois la même référence à '/.

Si vous voulez que `pmwiki` se charge automatiquement quand vous tapez une URL se terminant par un répertoire, vous pouvez faire une de ces deux choses. Si vous pouvez accéder aux directives d'alias du fichier de configuration de votre serveur web, alors utilisez l'alias décrit plus haut. Si vous ne pouvez pas, essayez en ajoutant la ligne suivante au fichier `.htaccess` du dossier de `pmwiki`.

```
DirectoryIndex pmwiki.php
```

Cela provoquera le chargement automatique de `pmwiki.php` à la place de `index.php` ou `index.html` et masquera le nom du fichier, au moins à la première visite du site.

4. Utiliser `mod_rewrite`

Le serveur Web Apache possède tout un tas de fonctionnalités délirantes, l'une d'entre elles est le [mod_rewrite](#). La bidouille suivante dans votre fichier `.htaccess` devrait rendre vos URL vraiment vraiment sympathiques:

```
RewriteEngine on RewriteBase / RewriteRule ^([A-Z].*)  
/path/to/pmwiki/pmwiki.php?pagename=$1 [L,qsappend]
```

Quand vous utilisez le `mod_rewrite`, assurez vous d'avoir les lignes suivantes dans votre fichier `local/config.php`:

```
$ScriptUrl = 'http://yourdomain.com/dir/pmwiki.php';  
$PageUrlFmt = 'http://yourdomain.com/$Group/$Title_';  
$PubDirUrl = 'http://yourdomain.com/dir/pub';  
$UploadUrlFmt = 'http://yourdomain.com/dir/uploads';
```

5. Avoir des URLs plus propre sans Pm Wiki

Vous devez changer votre configuration locale pour obtenir de Pm Wiki qu'il génère des jolies urls.
Paramétrer la variable \$EnablePathInfo à 1 indiquera à PmWiki de générer des noms de pages avec des slashes ("/") au lieu de points et supprimera tout l'aspect "pagename=". Puis paramétrer \$[Script Url?](#) avec le pseudo chemin que vous avez défini *sans* le slash final, devrait rendre les choses vraiment jolies. Par exemple, les commandes mod_rewrite ci-dessus combinées avec
\$ScriptUrl = "http://wiki.example.com";
dans le fichier config.php devrait produire une URL qui ressemble à ça
http://wiki.example.com/Main/HomePage/.

<< [Notification par courriel](#) | [Index doc admin](#) | [Configurer les liens InterMap](#) >>