

URL Rewriting ou réécriture d'url avec Apache

Le tutoriel que nous allons découvrir ensemble introduit les bases de la réécriture d' grâce au mode Rewrite du serveur web Apache. Il n'est pas exhaustif mais constitue une base suffisamment solide pour mettre en place un tel système qui fonctionne et qui soit efficace.

L'URL Rewriting, c'est quoi ?

La réécriture d'URL est, comme son nom l'indique, un système permettant de changer l'apparence d'une URL en vue d'une utilisation optimale pour les moteurs de recherches ou bien tout simplement pour des raisons esthétiques.

Par exemple, au lieu de présenter une URL originale aux visiteurs du type :

```
http://www.notresite.com/index.php?rub=lireArticle&id=8
```

Nous lui présenterons une URL réécrite qui pourrait avoir cette forme là :

```
http://www.notresite.com/articles/lire-8-titre-de-l-article.html
```

Nous constatons ici combien notre URL gagne en esthétisme et en optimisation pour le référencement. On a transformé notre URL d'origine dynamique (construite avec des pages PHP et des paramètres supplémentaires) en une « URL statique » (fichier HTML) contenant plusieurs mots-clés.

Mise en place du mode Rewrite

Pour mettre en place ce système, votre serveur web (d'un point de vue matériel) doit tout d'abord être équipé du logiciel [Apache](#). Apache est aujourd'hui devenu le processus serveur web le plus utilisé. Il est simple à mettre en place et il est gratuit. C'est d'ailleurs pourquoi on le retrouve dans les applications telles que EasyPHP, WAMP Server, MAMP, MovAMP, XAMP...

Nous devons tout d'abord commencer par activer le *mod_rewrite* d'Apache. Deux cas de figure s'offrent à nous :

- Nous avons la main mise totale sur notre serveur (serveur dédié par exemple).
- Nous dépendons d'un prestataire de services pour l'hébergement qui nous interdit l'accès à la configuration de ses serveurs (hébergement mutualisé).

Dans le premier cas, il nous suffit de vérifier dans le fichier *httpd.conf* d'Apache que le *mod_rewrite* est bien actif. Dans le cas contraire, nous devons juste décommenter la ligne. La ligne en question est la suivante :

```
Chargement du module de réécriture d'url  
LoadModule rewrite_module modules/mod_rewrite.so
```

Une ligne commentée est précédée d'un signe dièse. Si celle-ci est précédée d'un dièse, alors il nous suffit de le supprimer pour rendre le *mod_rewrite* actif.

Dans le cas où nous n'avons pas la possibilité de toucher au fichier de configuration d'Apache, nous

n'aurons d'autre choix que de demander directement à l'hébergeur si le `mod_rewrite` est actif sur leurs serveurs.

Écriture de règles simples

Une fois l'étape précédente effectuée, nous devons spécifier dans le fichier `.htaccess` (situé à la racine) de notre site que nous allons réécrire des URL. Pour cela, nous ajoutons la ligne de code suivante en respectant scrupuleusement la casse (majuscules / minuscules).

Rappel : les dièses sont des commentaires.

```
Configuration du module Rewrite
# On suit les liens symboliques
Options +FollowSymlinks
# Activation du mod rewrite d'Apache
RewriteEngine On
```

Maintenant, nous allons écrire toutes les règles permettant de réécrire nos URL. La première est simple, nous redirigeons `index.html` vers `index.php`.

```
Ecriture de la première règle de redirection
# Réécriture de index.html vers index.php
RewriteRule ^index\.html$ /index.php [L]
```

Des petites explications s'imposent. Nous venons ici d'écrire une **règle** grâce au mot-clé `RewriteRule`. Puis nous allons rechercher la chaîne `index.html` qu'il faudra rediriger vers `index.php` du répertoire racine (`/`). L'accent circonflexe signifie *début* et le dollar *fin*. Le marqueur `[L]` final est appelé *flag* (drapeau). *L* signifie ici *Last* (dernier). Cela veut dire que c'est la dernière règle de réécriture à tester. Le drapeau `[L]` est toujours à placer à la fin de votre dernière règle du fichier `.htaccess`. Notez qu'il faut échapper le point avec un antislash (`\`) car c'est un caractère particulier pour Apache.

Nous allons à présent reprendre notre tout premier exemple (celui avec les articles) afin d'écrire la règle générale de réécriture.

```
Exemple concret avec un module d'articles
# On suit les liens symboliques
Options +FollowSymlinks
# Activation du mod rewrite d'Apache
RewriteEngine On
# Réécriture de index.html vers index.php
RewriteRule ^index\.html$ /index.php
# Réécriture des URL des articles
RewriteRule ^/articles/lire-([0-9]+)-([a-z0-9\-\_]+)\.html$ /index.php?
rub=lireArticle&id=$1 [L]
```

Nous venons de créer une règle générale qui s'applique à tous les articles. Nous avons ajouté deux expressions régulières (entre parenthèses). La première indique que nous attendons une série de 1 ou plusieurs chiffres (le plusieurs est codé grâce au signe `+`). La seconde expression régulière recherche une chaîne de caractères composée de 1 ou plusieurs chiffres, lettres et tirets. La variable `$1` contient la suite de chiffres. Nous remplaçons donc cette valeur au bon endroit dans l'url originale. Cette variable fait référence à la première expression régulière. Si nous avions voulu faire passer le

titre de l'article dans l'url originale, nous aurions bien entendu utilisé la variable \$2.

Note : il est possible d'écrire autant de règles que nous le souhaitons.

Conclusion

Ce tutoriel est loin d'être exhaustif mais il donne les bases fondamentales pour réécrire des URL. Le site Apprendre-PHP.com fonctionne uniquement sur ce type de règles de réécriture. Nous vous invitons tout de même à lire le tutoriel de l'équipe de [Webmaster-Hub](#) qui est à la fois très didactique et facile à comprendre. Il s'agit de [Mod rewrite, ou la réécriture des URL "à la volée"](#).