

Les fichiers .htaccess (sous apache)

Nous avons vu en cours (et en TP) l'utilisation du fichier httpd.conf afin de personnaliser le fonctionnement du serveur apache.

La configuration basée sur ce fichier permet une gestion centralisée de la configuration. Cette configuration peut se faire à 2 niveaux, soit globalement, soit pour une partie du serveur (section <Directory>, <Files>, <Location> ...).

Cette gestion centralisée de la configuration peut ne pas être souhaitée, ou aussi offrir un manque de souplesse. Pour palier à ce problème le serveur apache autorise l'utilisation de directives placées dans des fichiers particuliers se trouvant dans l'arborescence même du site en question.

Voici un copier coller du manuel d'apache concernant ce fichier:

Apache permet une gestion décentralisée de la configuration par l'utilisation de fichiers spéciaux à l'intérieur de l'arborescence web. Ces fichiers spéciaux sont généralement appelés .htaccess, mais n'importe quel nom peut être défini par la directive AccessFileName. Les directives placées dans les fichiers .htaccess s'appliquent au répertoire où se trouve le fichier ainsi qu'aux sous répertoires. Les fichiers .htaccess respectent la même syntaxe que les fichiers principaux de configuration. Comme les fichiers .htaccess sont lus à chaque requête, Les modifications effectuées dans ces fichiers prennent effet immédiatement.

Pour savoir si une directive peut être placée dans les fichiers .htaccess, vérifiez le contexte de la directive. L'administrateur peut contrôler quelles directives peuvent être placées dans les fichiers .htaccess en modifiant la directive AllowOverride du fichier principal de configuration.

C'est bien beau tout cela, mais concrètement à quoi ça sert ?

Après avoir effleuré les concepts des fichiers .htaccess nous allons aborder quelques cas d'utilisation, illustrés à chaque fois par des exemples.

Il faut préciser que la plupart des fonctions permises par les fichiers .htaccess sont réalisable avec des scripts côté serveur (php, cgi), mais nécessite alors un effort supplémentaire de développement, et offre moins de souplesse.

Il y a plusieurs cas classique d'utilisation de ce fichier « magique ».

- **Restriction de l'accès a une page pour certaines adresses IP:**

Voici un exemple de fichier .htaccess refusant la connection à tout le monde sauf aux personnes abonné chez une fournisseur d'accès « français ».

```
order allow,deny
deny from all
allow from *.fr
```

- **Restriction de l'accès à une page par un mot de passe:**

La protection d'un répertoire par un mot de passe se fait souvent en php (ou autre langage de script) alors qu'il est bien plus simple de la faire avec un .htaccess comme le montre l'exemple ci dessous:

Fichier .htaccess:

```
AuthName "private"
```

```
AuthType Basic
AuthorUserFile .htpasswd
```

Fichier .htpasswd:

```
toto:grrefgregfd
titi:fgfbnbnjnn
```

Avec ces 2 fichiers il existe 2 « utilisateurs » pouvant accéder au répertoire dans lesquels ils se trouvent: toto et titi.

Les mots de passes sont comme vous le voyez crypté (même méthode quel /etc/passwd sous unix)

- **Personnaliser les erreurs http:**

```
ErrorDocument 401 /error/401.php
ErrorDocument 403 /error/403.php
ErrorDocument 404 /error/404.php
```

Dans l'exemple ci dessus en cas d'erreur http, l'utilisateur n'est pas redirigé vers la page par défaut pour l'erreur, mais vers la page web correspondant à l'erreur.

Ainsi, il est possible de concevoir des pages d'erreurs, conservant l'aspect graphique du site.

Il y d'autres utilisations possibles des fichiers .htaccess(et moins documentées), parmi lesquels 2 ont retenu mon attention:

- **Autoriser le listing d'un répertoire:**

Il est courant de désactiver le listage des répertoires pour des raisons de sécurité, cependant il existe des situations ou l'on souhaite activer cette option (page de téléchargement pas exemple). Il est certes possible de parvenir à se résultat en utilisant un script (php, cgi), mais à l'aide d'un fichier .htaccess, ceci se fait en une seule instruction.

Directive à inclure dans le .htaccess pour autoriser le listage d'un répertoire:

```
Options indexes
```

- **Personnaliser le listing d'un répertoire:**

Le listage du répertoire présenté ci dessus, utilise la configuration par défaut d'apache en ce qui concerne les icônes correspondant aux différents types de fichiers. Il est cependant possible de personnalisé un peu l'aspect de la page généré.

Exemple de personnalisation d'un listage:

```
Options indexes
AddIcon (DIR,/icons/dir.png) ^^DIRECTORY^^
AddIcon (IMG,/icons/img.png) images/*
AddIcon (UP,/icons/up.png) .
DefaultIcon /icons/unknown.png
```

Nous avons personnalisé les icônes utilisé lors du listage, mais l'apparence de la page n'a pas vraiment changé. Cependant il est possible de le personnalisé encore plus, à l'aide de deux directives:

```
HeaderName /inc/header.htm
ReadmeName readme.html
```


Des informations supplémentaires sur les fichiers .htaccess:

- **Les fichiers .htaccess sont hiérarchisé**, ainsi prenons l'arborescence suivante:

```
/
+-- .htaccess
+-- error/
| +-- 401.php
| +-- 403.php
| +-- 404.php
+-- icons/
| +-- dir.png
| +-- img.png
| +-- unknown.png
| +-- up.png
+-- inc/
| +-- .htaccess
| +-- header.html
+-- private
| +-- .htaccess
| +-- secret
|   +-- .htaccess
+-- readme.html
```

Ci dessous quelques cas pratique de la hiérarchie:

- Pour rediriger les pages d'erreurs pour tout le site il suffit de modifier le .htaccess à la racine du site.
 - Pour interdire le listage des répertoires sauf dans le dossiers private (et ses enfants) :
 - ◆ Dans le .htaccess à la racine on ajoute la directive « Options -Indexes »
 - ◆ Dans le .htaccess du répertoire private on ajoute la directive « Options Indexes »
-
- **Les fichiers .htaccess et les performances d'apache:**

Comme décrit dans le manuel d'apache, l'utilisation des fichiers.htaccess, n'est pas anodine en terme de performances du serveur. En effet pour chaque requête le serveur examine le fichier .htaccess du répertoire courant, mais du à la hiérarchie, il doit aussi examiner les .htaccess de tout les dossiers d'un niveau supérieur.

Il faut donc utiliser ces fichiers avec précaution sur un serveur en très forte charge.