eBook Gratuit

APPRENEZ

Hypertext Access file

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#.htaccess

Table des matières

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: hypertext-access-file

It is an unofficial and free Hypertext Access file ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Hypertext Access file.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec le fichier Hypertext Access

Remarques

Un fichier `.htaccess` contrôle la manière dont Apache interagit avec votre site. Lorsqu'un fichier `.htaccess` est placé dans le répertoire de votre domaine (généralement le répertoire racine), le fichier est détecté et exécuté par Apache.

Un fichier `.htaccess` est couramment utilisé pour les éléments suivants:

- Refuser des adresses IP spécifiques à votre site
- Mot de passe protégeant votre site
- Réécriture d'URL
- Pages d'erreur personnalisées
- Compression et mise en cache de fichiers
- Sécurité générale et prévention des piratages

Versions

Diverses versions d'Apache

Version	Version actuelle	Libération
1.3	1.3.42	1998-06-06
2.0	2.0.65	2002-04-06
2.2	2.2.31	2005-12-01
2.4	2.4.23	2012-02-21

Examples

Mise en place .htaccess

> `.htaccess` fichiers `.htaccess` (ou "fichiers de configuration distribués") permettent de modifier la configuration par répertoire. Un fichier contenant une ou plusieurs directives de configuration est placé dans un répertoire de document particulier et les directives s'appliquent à ce répertoire et à tous ses sous-répertoires.

Un fichier .htaccess contrôle la manière dont Apache interagit avec votre site. Il est utilisé pour modifier les requêtes et modifier le comportement par défaut sans avoir à modifier les fichiers de configuration du serveur principal.

La configuration de `.htaccess` est aussi simple que d'ouvrir un bloc-notes et de l'enregistrer en tant que `.htaccess` . Généralement, ce fichier sera placé dans le répertoire `root` des fichiers de votre site Web, mais vous pouvez l'utiliser sous plusieurs répertoires différents. Ceci est particulièrement utile si vous cherchez à protéger des répertoires spécifiques par mot de passe.

Activation de .htaccess

> Parfois, même une seule erreur dans votre fichier `httpd.conf` ou `.htacces` entraînera une fusion temporaire du serveur, et les utilisateurs verront la page *500 - Internal Server Error* . Donc, assurez-vous de toujours faire une sauvegarde de vos fichiers `httpd.conf` et `.htaccess` avant de procéder à une modification.

```
<Directory "/var/www">
    AllowOverride All
</Directory>
```

`.htaccess` fichiers `.htaccess` sont normalement activés par défaut. Ceci est contrôlé par la directive `AllowOverride` dans le fichier `httpd.conf` . Cette directive ne peut être placée qu'à l'intérieur d'une section `<Directory>` .

À côté de `All` il existe de nombreuses autres valeurs qui limitent la configuration de certains contextes uniquement. Certains d'entre eux sont:

- **Aucun** - Désactive complètement `.htaccess` .
- **AuthConfig** - Directives d'autorisation telles que celles concernant l'authentification de base.
- **FileInfo** - Directives traitant de la définition des en-têtes, des documents d'erreur, des cookies, de la réécriture d'URL, etc.
- **Index** - Personnalisations par défaut des listes de répertoires.
- **Limite** - Contrôle l'accès aux pages de différentes manières.
- **Options** - Accès similaire aux index, mais inclut encore plus de valeurs telles que ExecCGI, FollowSymLinks, Includes et plus.

```
# Only allow .htaccess files to override Authorization and Indexes
AllowOverride AuthConfig Indexes
```

Pages d'erreur personnalisées

`.htaccess` peut être utilisé pour définir une page d'erreur personnalisée correspondant au thème de votre site Web au lieu de voir une page d'erreur blanche avec un techno-babble noir lorsque les utilisateurs se retrouvent sur une page avec un code de réponse du serveur d'erreur. La page d'erreur peut être n'importe quel fichier analysable par le navigateur, y compris (mais sans s'y limiter) .html, .php, .asp, .txt, .xml.

Exemples de presque tous les codes de réponse d'erreur courants:

```
#Client Errors
```

```
ErrorDocument 400 /mycool400page.html    # Bad Request
ErrorDocument 401 /mycool401page.html    # Unauthorized
ErrorDocument 402 /mycool402page.html    # Payment Required
ErrorDocument 403 /mycool403page.html    # Forbidden
ErrorDocument 404 /mycool404page.html    # Page Not Found

#Server Errors

ErrorDocument 500 /mycool500page.html    # Internal Server Error
ErrorDocument 501 /mycool501page.html    # Not Implemented
ErrorDocument 502 /mycool502page.html    # Bad Gateway
ErrorDocument 503 /mycool503page.html    # Service Unavailable
ErrorDocument 504 /mycool504page.html    # Gateway Timeout
ErrorDocument 505 /mycool505page.html    # Internal Server Error
```

Il est toujours recommandé d'inclure les documents d'erreur pour les réponses d'erreur les plus courantes, 400, 403, 404 et 500, car ces erreurs peuvent se produire sur tous les navigateurs.

l'erreur 500 est l'une des erreurs les plus notoires car elle se produit si quelque chose échoue lors du chargement de la page à envoyer, le plus souvent des erreurs de prétraitement de HTML comme PHP, ASP et autres préprocesseurs HTML. Lors des tests, il est recommandé de configurer la page 500 pour afficher l'erreur qui s'est produite, plutôt qu'une page d'erreur 500 non spécifique.

Pour permettre à la page d'erreur 500 d'écrire une erreur spécifique, consultez l'un des éléments suivants, basé sur le préprocesseur HTML que vous utilisez: php asp

Définition du fuseau horaire du serveur

Il existe de nombreux fuseaux horaires dans le monde, il est important de vous assurer que votre serveur est configuré sur le bon. Ceci est fait dans `.htaccess` en utilisant:

```
SetEnv TZ America/Indianapolis
```

Quelques exemples d'autres fuseaux horaires possibles:

```
America/Los_Angeles
America/Los_Angeles - Pacific Time
Pacific/Honolulu - Hawaii
```

Assurez-vous simplement d'utiliser `SetEnv` devant le fuseau horaire sélectionné.

Lire Démarrer avec le fichier Hypertext Access en ligne: https://riptutorial.com/fr/dot-htaccess/topic/1023/demarrer-avec-le-fichier-hypertext-access

Chapitre 2: Gestion des types de fichiers

Examples

Permettre à PHP d'être analysé en HTML

Si vous souhaitez inclure du code PHP dans votre fichier HTML et que vous ne souhaitez pas renommer le type de fichier de `.html` ou `.htm` en `.php` , le `.html` ci-dessous permet à votre fichier HTML d'analyser correctement votre code PHP.

```
AddHandler application/x-httpd-php .html .htm
```

Lire Gestion des types de fichiers en ligne: https://riptutorial.com/fr/dot-htaccess/topic/1690/gestion-des-types-de-fichiers

Chapitre 3: Optimisation de la vitesse

Examples

Activer la compression (Apache 2.0+)

> L'activation de la compression gzip peut réduire la taille de la réponse transférée jusqu'à 90%, ce qui peut réduire considérablement le temps de téléchargement de la ressource, réduire l'utilisation des données pour le client et améliorer le délai de rendu de vos pages. - *Insights PageSpeed*

La compression peut être activée avec ceci:

```
AddOutputFilterByType DEFLATE "text/html"/
                              "text/plain"/
                              "text/xml"/
                              "text/css"/
                              "text/javascript"/
                              "application/javascript"
```

Apache Docs

Exploitation de la mise en cache du navigateur (Apache 2.0+)

> La récupération des ressources sur le réseau est à la fois lente et coûteuse: le téléchargement peut nécessiter plusieurs allers-retours entre le client et le serveur, ce qui retarde le traitement et bloque le rendu du contenu de la page. Toutes les réponses du serveur doivent spécifier une stratégie de mise en cache pour aider le client à déterminer si et quand il peut réutiliser une réponse récupérée précédemment. - *Insights PageSpeed*

Vous pouvez exploiter la mise en cache du navigateur comme suit:

```
# Enable browser caching
ExpiresActive On

# Set the default caching duration
ExpiresDefault "access plus 1 week"

# Change the caching duration by file type
ExpiresByType text/html "access plus 2 weeks"
```

Apache Docs

Activer KeepAlive (Apache 2.0+)

> L'extension Keep-Alive de HTTP / 1.0 et la fonctionnalité de connexion persistante de HTTP / 1.1 fournissent des sessions HTTP de longue durée permettant d'envoyer

> plusieurs requêtes via la même connexion TCP. Dans certains cas, il a été démontré que cela entraîne une accélération de presque 50% des temps de latence pour les documents HTML contenant de nombreuses images. Pour activer les connexions persistantes, définissez KeepAlive On. - *Apache Docs*

```
# Enable KeepAlive
KeepAlive On

# OPTIONAL — limit the amount of requests per connection with 'MaxKeepAliveRequests'
# Example: MaxKeepAliveRequests 500

# OPTIONAL — limit the amount of time the server will wait before it closes
# the connection with 'KeepAliveTimeout'
# Example: KeepAliveTimeout 500
```

Apache Docs

Lire Optimisation de la vitesse en ligne: https://riptutorial.com/fr/dot-htaccess/topic/3893/optimisation-de-la-vitesse

Chapitre 4: Réécriture et redirection

Remarques

Avant de pouvoir réécrire les URL, un module appelé `mod_rewrite.c` doit être activé. Habituellement, il est désactivé dans la configuration par défaut.

`mod_rewrite` peut être activé en exécutant la commande

```
$ sudo a2enmod mod_rewrite
$ sudo service apache2 restart
```

ou en commentant les lignes

```
#LoadModule rewrite_module modules/mod_rewrite.so
#AddModule mod_rewrite.c
```

dans le fichier `httpd.conf` .

Examples

Drapeaux de réécriture populaires

F | interdit

Semblable à `Deny` , cet indicateur force le serveur à renvoyer immédiatement un code de statut *403 Forbidden* au navigateur ou au client demandeur pour la demande.

Exemple: refuser l'accès aux requêtes se `exe` par `exe` :

```
RewriteRule .exe$ - [F]
```

G | parti

Si une ressource demandée était disponible dans le passé, mais qu'elle n'est plus disponible, vous pouvez utiliser cet indicateur pour forcer le serveur à renvoyer immédiatement un code de statut *410 Gone* au navigateur ou au client demandeur pour la demande.

Exemple: Dites à un visiteur qu'un ancien produit n'existe plus:

```
RewriteRule ^old-product.html$ - [G]
```

L | dernier

Dans la plupart des contextes, à l'exception de `.htaccess` , cet indicateur demande à `mod_rewrite` d'arrêter de traiter le jeu de conditions / règles en cours, de la même manière que `last` et `break` (Perl et C, respectivement).

Cependant, dans le contexte `.htaccess` ou `<Directory>` , une requête réécrite à l'aide d'un `RewriteRule` avec cet indicateur sera renvoyée au moteur d'analyse URL pour un traitement ultérieur. En tant que tel, il est possible que l'URI réécrit soit traité par le même contexte et éventuellement modifié.

Une recommandation générale consiste à utiliser l'indicateur `END` non seulement pour arrêter le traitement de l'ensemble de conditions / règles en cours, mais également pour empêcher toute réécriture ultérieure dans ces contextes.

Remarque: Les indicateurs `F` et `G` , décrits ci-dessus, utilisent tous les deux implicitement `L` , vous n'avez donc pas besoin de les spécifier séparément.

N | suivant

Cet indicateur réexécutera le processus de réécriture depuis le début, en commençant par le premier ensemble de conditions / règles. Cette fois, l'URL à faire correspondre n'est plus l'URI d'origine, mais l'URI réécrit renvoyé par le dernier jeu de règles. Utilisez cet indicateur pour redémarrer le processus de réécriture.

Un mot d'avertissement: utilisez ce drapeau avec précaution, car il pourrait en résulter une boucle infinie!

NC | nocase

Cela demande à `mod_rewrite` faire correspondre le `Pattern` d'un `RewriteRule` sans être sensible à la casse. Pour clarifier, `MyIndex.html` et `myindex.html` seraient considérés par le module comme la même chose. De plus, cet indicateur vous permet d'utiliser `az` au lieu de `A-Za-z` dans une expression régulière.

R | rediriger

Cet indicateur est utilisé pour envoyer une réponse de redirection HTTP au navigateur / client demandeur.

Par défaut, si aucun code n'est donné, une réponse de redirection avec le code d'état *302 Found* (similaire à une redirection temporaire) sera renvoyée. Si vous souhaitez utiliser une redirection plus permanente, vous devez utiliser le code d'état `302` (*301 Moved Permanently*).

En règle générale, seuls les codes d'état compris entre 300 et 399 doivent être utilisés avec ce drapeau. Si des codes de statut en dehors de cette plage sont utilisés (ce qui est parfaitement acceptable), alors la chaîne de substitution est ignorée et la réécriture est arrêtée comme si le drapeau `L` était utilisé. Dans certains cas, il s'agit d'un moyen pratique de forcer les réponses *404 Not Found* , même si la demande pointe vers une ressource existante.

Exemple: Émettez une réponse de redirection *302 trouvée* :

```
RewriteRule ^bus$ /train [R,L]
```

Exemple: Émettre une réponse de redirection de *301 déplacé de manière permanente* :

```
RewriteRule ^speed-train$ /hyperloop [R=301,L]
```

Exemple: forcer un *404 introuvable* :

```
RewriteRule ^blip$ - [R=404,L]
```

redirections www et non-www

Rediriger tout domaine nu vers `www.[your_domain].tld` :

```
# Start Apache Rewriting engine
RewriteEngine On
# Make sure you're not already using www subdomain
# and that the host string is not empty
RewriteCond %{HTTP_HOST} !^$
RewriteCond %{HTTP_HOST} !^www\.
# We check for http/https connection protocol
RewriteCond %{HTTPS}s ^on(s)|
# In case the previous conditions matches, redirect to www
RewriteRule ^(.*)$ http%1://www.%{HTTP_HOST}/$1 [R=301,L]
```

Rediriger `www.[your_domain].tld` vers `[your_domain].tld`

```
# Start Apache Rewriting engine
RewriteEngine On
# We check if we're on the www subdomain
RewriteCond %{HTTP_HOST} ^www\.([^\.]+\.[^\.]+)$
# In case the previous condition matches, redirect to non-www
RewriteRule ^(.*)$ http://%1/$1 [R=301,L]
```

Rediriger tous les niveaux de sous-domaines imbriqués vers votre domaine principal:

```
# Start Apache Rewriting engine
RewriteEngine On
# We check if there's a subdomain
RewriteCond %{HTTP_HOST} \.([^.]+\.[^.]+)$
# redirect to the main domain name
RewriteRule ^ http://%1%{REQUEST_URI} [R=301,L]
```

URL optimisées pour le référencement

Les moteurs de recherche n'indexeront pas vos produits si vous avez une URL comme celle-ci:

```
http://www.yourdomain.com/product.php?id=123
```

L'URL SEO friendly ressemblerait à `http://www.yourdomain.com/123/product-name/` . Le code suivant vous aide à y parvenir sans avoir à modifier le code `product.php` .

```
RewriteEngine On
RewriteRule ^product/([0-9]+)/product-name-slug/?$ product.php?id=$1
```

Ajout d'une barre oblique à la fin

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_URI} !(.*)/$
RewriteRule ^(.*)$ /$1/ [L,R=301]
```

Le premier `RewriteCond` aide à exclure les fichiers. Le deuxième `RewriteCond` vérifie s'il existe déjà une barre oblique. Si c'est le cas, `RewriteRule` n'est pas appliqué.

Si vous avez une URL qui ne devrait pas être réécrite, vous pouvez ajouter un autre `RewriteCond` .

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_URI} !(.*)/$
RewriteCond %{REQUEST_URI} !url/to/not/rewrite
RewriteRule ^(.*)$ /$1/ [L,R=301]
```

redirections http et https et configuration HSTS

Redirect générique *https:*

```
# Enable Rewrite engine
RewriteEngine on

# Check if URL does not contain https
RewriteCond %{HTTPS} off [NC]
# If condition is true, redirect to https
RewriteRule (.*) https://%{SERVER_NAME}/$1 [R=301,L]
```

Redirect générique *http:*

```
# Enable Rewrite engine
RewriteEngine on

# Check if URL does contain https
RewriteCond %{HTTPS} on [NC]
# If condition is true, redirect to http
RewriteRule (.*) http://%{SERVER_NAME}/$1 [R=301,L]
```

Connexion HTTPS forcée (HSTS):

```
<IfModule mod_headers.c>
    Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
</IfModule>
```

où l'option `includeSubDomains` peut être supprimée si HSTS doit être appliqué uniquement au domaine de base ou au domaine avec la configuration ci-dessus.

Redirection avec / sans paramètres de requête

Rediriger sans paramètres de requête:

```
RewriteRule ^route$ /new_route_without_query [L,R=301,QSD]
```

Rediriger avec les paramètres de requête:

```
RewriteCond %{QUERY_STRING} ^$
RewriteRule ^/?route$ %{REQUEST_URI}?query=param1&query2=param2 [NC,L,R=301]
```

Lire Réécriture et redirection en ligne: https://riptutorial.com/fr/dot-htaccess/topic/1550/reecriture-et-redirection

Chapitre 5: Refuser l'accès

Examples

Refuser les adresses IP

```
order allow,deny
deny from 255.0.0.0
allow from all
```

Cela refuse l'accès à l'adresse IP `255.0.0.0` .

```
order allow,deny
deny from 123.45.6.
allow from all
```

Cela interdit l'accès à toutes les adresses IP comprises entre `123.45.6.0` et `123.45.6.255` .

Prévention des liens chauds

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?yourdomain.com/.*$ [NC]
RewriteRule \.(gif|jpg|css)$ - [F]
```

Cela bloque tous les liens vers les fichiers ".gif", ".jpg" et ".css" qui ne proviennent pas du nom de domaine `http://www.yourdomain.com` .

Afficher le contenu alternatif:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?yourdomain.com/.*$ [NC]
RewriteRule \.(gif|jpg)$ http://www.yourdomain.com/angryman.jpg [R,L]
```

Cela bloque tous les liens vers les fichiers '.gif' et '.jpg' qui ne proviennent pas du nom de domaine ' http://www.votredomaine.com/ ' et affiche le fichier ' http://www.votredomaine.com/angryman .jpg 'à la place.

Refuser l'accès des adresses IP aux fichiers / répertoires

```
# Deny access to a directory from the IP 255.0.0.0
<Directory /path/to/directory>
    order allow,deny
    deny from 255.0.0.0
    allow from all
</Directory>
```

```
# Deny access to a file from the IP 255.0.0.0
<FilesMatch "^\.ht">
    order allow,deny
    deny from 255.0.0.0
    allow from all
</FilesMatch>
```

Lire Refuser l'accès en ligne: https://riptutorial.com/fr/dot-htaccess/topic/4741/refuser-l-acces

Chapitre 6: Sécurité générale et prévention des piratages

Remarques

La redirection .htaccess est un vecteur commun pour que les pirates malveillants exploitent et infectent les sites Web. Nous avons vu quels sont les fichiers .htaccess, comment ils sont utilisés par des pirates malveillants et comment protéger votre site Web.

Examples

Prévention de piratage

Empêcher l'accès à votre fichier `.htaccess`

```
<Files .htaccess>
order allow,deny
deny from all
</Files>

# Rename the file
AccessFileName thehtfile.ess
```

Prévenir les attaques par URL

```
# Enable rewrites
RewriteEngine On

# Block <script> tags from executing in the URL
RewriteCond %{QUERY_STRING} (<|%3C).*script.*(>|%3E) [NC,OR]

# Block scripts from setting a PHP Globals variable
RewriteCond %{QUERY_STRING} GLOBALS(=|[|\%[0-9A-Z]{0,2}) [OR]

# Block scripts from using base64_encode
RewriteCond %{QUERY_STRING} base64_encode.*(.*) [OR]

# Block scripts from using the a_REQUEST variable
RewriteCond %{QUERY_STRING} _REQUEST(=|[|\%[0-9A-Z]{0,2})
```

Désactiver l'utilisation de scripts sur vos répertoires.

```
AddHandler cgi-script .php .pl .py .jsp .asp .htm .shtml .sh .cgi
Options -ExecCGI
```

Désactiver l'index du répertoire

Si l'index de répertoire activé signifie que si quelqu'un accède à un dossier qui ne contient pas index.php, index.html, index.htm ou tout autre fichier par défaut défini dans DirectoryIndex dans la configuration apache, tous les fichiers de ce dossier seront répertoriés dans le navigateur. vous essayez de visiter cette page.

Souvent, l'index de répertoire est activé par défaut sur votre serveur Apache. Dans ce cas, une bonne pratique de sécurité consiste à désactiver l'index de répertoire avec la ligne suivante:

```
Options -Indexes
```

Lire Sécurité générale et prévention des piratages en ligne: https://riptutorial.com/fr/dot-htaccess/topic/2531/securite-generale-et-prevention-des-piratages

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec le fichier Hypertext Access	Community, Dilip Raj Baral, hjpotter92, James Oswald, Lag, Mike Rockétt, tbodt
2	Gestion des types de fichiers	Dilip Raj Baral, John R Perry, Jon Lin, Marvin, mauris, Mike Rockétt, Nicholas Qiao
3	Optimisation de la vitesse	John R Perry
4	Réécriture et redirection	Bogdan Alexandru Militaru, Dilip Raj Baral, Florian Lemaitre, hjpotter92, James, John R Perry, Mike Rockétt, shaN, Sven Reuter
5	Refuser l'accès	Dilip Raj Baral, John R Perry, tbodt
6	Sécurité générale et prévention des piratages	ban17, Dilip Raj Baral, John R Perry, Lag, Meysam, Mike Rockétt, OpenWebWar