**Пренаписване на интернет адреси с .htaccess**

Промяната на интернет адреси с mod\_rewrite се прави основно по следните seo причини:

* съкращаване на адресите: адресите стават за индексиране от търсачките (seo friendly) и лесни за ръчно изписване в браузър
* пренасочване на стари адреси към нови при преименуване на файлове на страници
* ако има страници, които се достъпват с повече от един адрес се пренасочват излишните адреси
* скриване на вида файлове на страниците на сайта

Пренасочванията се извършват от правила зададени във файла .htaccess, който трябва да се намира в главната уеб директория.

Правилата използват регулярни изрази (regular expressions) и могат да бъдат много сложни. В тази статия обаче ще разгледам само най-често ползваните правила от SEO специалистите.

За да работи този файл е нужно уеб сървъра да има инсталиран модул mod\_rewrite и в конфигурационния файл /etc/httpd/conf/httpd.conf  трябва да е зададена опцията **AllowOverride All**

На повечето съвременни хостинг сървъри тези опции са включени по подразбиране, ако това не е така ще трябва да се свържеш със системния администратор на хостинга.

**Проверка дали mod\_rewrite работи**

Постави следните редове в .htaccess файла и опитай да отвориш адреса http://domain.com/test  където domain.com е името на твоя сайт. Ако се отвори сайта ganbox.com това значи всичко е наред и mod\_rewrite работи.

Options +FollowSymLinks

RewriteEngine On

RewriteBase /

RewriteRule test http://ganbox.com/? [NC,R,L]

Обикновено файла .htaccess започва със следните няколко реда. За краткост ще бъдат пропускани в примерите долу.

Options +FollowSymLinks -Indexes -MultiViews

RewriteEngine On

RewriteBase /

**Просто пренасочване на преименувана страница**

Ако страница /about.html е преименувана на /contact.php

redirect 301 /about.html /contact.php

аналогично

RewriteRule ^about\.html$ /contact.php [R=permanent,L]

[L]  казва, че това е последното правило, което трябва да се обработи.

**Скриване на вида файлове**

Всеки адрес от вида domain.com/страница, като страница може да съдържа малки или главни букви, цифри и знаците “-” и “\_”.

RewriteRule ^([-\w0-9\_]+)$ /$1.php [L,NC,QSA]

Пример: при написване на адрес domain.com/форма-за-контакти2 ще бъде потърсена страница /форма-за-контакти2.php

**Внимание:** ако уеб съръвра е на Windows машина, горния регулярен израз вероятно няма да сработи,заради бъг в win32 mod\_rewrite апаче модула.

RewriteRule ^([-А-Яа-яA-Za-z0-9\_]+)$ /$1.asp [L,NC,QSA]

Вместо \w или А-Яа-яA-Za-z  използвай АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЬЮЯабвгдежзийклмнопрстуфхцчшщъьюяA-Za-z

В този пример при написване на адрес domain.com/Условия\_За\_Ползване ще бъде потърсена страница /Условия\_За\_Ползване.asp

като в полето за адреси на браузъра ще остане да се вижда http://domain.com/Условия\_За\_Ползване

Очевидно е, че по този начин няма как да се разбере какви скриптове управляват страниците на сайта. Дори може да се направи страниците да изглеждат като html файлове, а да се изпълняват едноименни php файлове с правилото:

RewriteRule ^([-А-Яа-яA-Za-z0-9\_]+)\.html$ /$1.php [QSA,L]

в този случай ако напишеш http://domain.com/clients.html ще бъде потърсен файла /clients.php. Този подход има двоен положителен ефект – от една страна търсачките и от друга сигурност.

**Съкращаване на адреси**

Много често при [SEO оптимизация](http://ganbox.com/) на динамични сайтове с CMS се налага съкращаване на адреси.

Дълги адреси като този

http://domain.com/index.php?sec=pro&cat=48&category=avto

да изглеждат като този

http://domain.com/sec/pro/cat/48/category/avto

RewriteRule

^sec/([^/]+)/cat/([0-9]+)/category/([^/]+)/?$

/index.php?sec=$1&cat=$2&category=$3 [QSA,L]

Разбира се горното правило се пише на един ред, като между отделните части има по един интервал.

След това в кода се правят промени на всички места, които образуват връзки, за да откриеш всички тези места може да търсиш във всички файлове за http или за href.

Хубавото е, че старите адреси продължават да работят и не е проблем за хората, които са направили Bookmark към стар адрес.

**Пренасочване на адреси към един файл**

Превод на адреси от вида www.domain.com/contacts към файл index.php, като подава по GET в променлива p името на страницата www.domain.com/index.php?p=contacts

# ако не бъде открит файл

RewriteCond %{SCRIPT\_FILENAME} !-f

# ако не бъде открита директория

RewriteCond %{SCRIPT\_FILENAME} !-d

# подава низа към файла index.php

RewriteRule ^([^/]+)/?$ /index.php?p=$1 [QSA,L]

По този начин всеки адрес ще се изпраща за обработка към един единствен файл index.php често наричан диспечер (dispatcher).

В този файл се взема подадения низ и се слага в променлива $page. След това се проверява и ако файла със страницата не съществува се преминава към страница с грешка 404.php

$page= isset($\_GET['p']) ? $\_GET['p'] : 'home';

if(!file\_exists(ABS\_PATH.$page.'.php')) $page = '404';

include\_once(ABS\_PATH.$page.'.php');

От съображения за сигурност е много важно преди този код да е дефинирана константата ABS\_PATH като в нея има абсолютен път до главната уеб директория например:

define('ABS\_PATH', '/home/account/www/');

където account е потребителско име за cpanel на хостинга.

Ако се очаква в адреса на страницата да има само латински букви и тирета, тогава в index.php преди реда с file\_exists добави следния ред:

$page = preg\_replace('|[^A-Za-z\_-]|','',$page);

# премахва всичко различно от буква и знаците \_ и -

**Премахване на проблем с дублирани адреси**

Наскоро имах следния проблем при оптимизиране на динамичен сайт. В навигацията сайта има два бутона за превключване на езика с българското и английското знаме.

Ако текущия адрес е http://domain.com/page.php?id=7

то бутона с английското знаме сочи към http://domain.com/page.php?lang=en&id=7

а бутона с българското знаме към http://domain.com/page.php?lang=bg&id=7

От потребителска гледна точка е удобно защото при смяна на езика оставаш в текущата страница, но от гледна точка на търсачката всяка страница на български има по два адреса.

Дори началната страница има 3 адреса: http://domain.com,  http://domain.com/index.php  и  http://domain.com/index.php?lang=bg

Решение:

# премахване на всички низове index.\* от интернет адреса.

# Заб. Изтрий тези две правила ако искаш да инсталираш форум или блог

RewriteCond %{THE\_REQUEST} \ /(.+/)?index\.([^/]+)(\?.\*)?\  [NC]

RewriteRule ^(.+/)?index\.([^/]+)$ /%1 [NC,R=301,L]

# Правила за премахване на параметър lang=bg като запазва всичко останало в адреса

RewriteCond %{QUERY\_STRING} ^(([^&]\*&)\*)(lang=bg)($|&)(.\*)

RewriteRule .\* $0?%1%5 [N,E=REMOVED:true]

RewriteCond %{ENV:REMOVED} true

RewriteRule ^ %{REQUEST\_URI} [L,R=301]

**Внимание:** Ако добавиш първите две правила за премахване на index от адресите и в последствие решиш да инсталираш форум, блог или някакъв CMS в поддиректория в същия сайт, се очаква да имаш проблеми и  инсталацията няма да може да се извърши.

**Премахване на www.**

Пренасочване на  www.domain.com/страница  към domain.com/страница

RewriteCond %{HTTP\_HOST} ^www\.(domain\.com)$ [NC]

RewriteRule ^(.\*)$ http://%1%{REQUEST\_URI} [R=301,L]

**Други полезни трикове в .htaccess**

**Ограничаване на достъпа**

**по IP адрес**

За да забраниш достъпа до сайта от IP адрес 207.246.65.22 в началото на файла напиши

# ban web proxy

Deny from 207.246.65.22

За да забраниш цялата мрежа 64.4.0.0 – 64.4.63.255 на Microsoft и техния MSN Bot напиши

Deny from 64.4.0.0/18

За да забраниш няколко  мрежи в Киев от които в блога вали спам изпращан от различни IP адреси като: 95.133.140.7, 95.133.192.198 и др. напиши:

Deny from 91.124.0.0/16

Deny from 91.195.12.0/23

Deny from 94.178.0.0/15

Deny from 95.132.0.0/14

Има предвид, че така блокираш много хора от Киев. Прецени сам дали очакваш читатели от там.

**по тип на търсения файл**

Забрана за показване на файлове, които завършват на .txt

<Files ~ "\.txt$">

    Order allow,deny

    Deny from all

</Files>

**по вид браузър**

Ако имаш директория /old в която има стара версия на сайта, която поддържа архаичния браузър Internet Explorer 6, може да пренасочиш всички заявки от такъв браузър заедно с подадените параметри към тях.

# за браузър IE6 или по-стар пренасочва към /old

RewriteCond %{HTTP\_USER\_AGENT} MSIE\ [1-6]

RewriteRule !^old(/.\*)?$ /old%{REQUEST\_URI} [QSA,L]

**ограничаване на ботове**

Ако забележиш в логовете, че твоя сайт е налазен от ботове може да използваш следния код:

RewriteCond %{HTTP\_USER\_AGENT} ^$  
RewriteRule ^.\* - [F]

Това ще блокира всички ботове, които не изпращат USER\_AGENT. Няма да се блокират ботовете на търсачките Google, Yahoo и др. Ако някой програмист иска да обхожда твоя сайт с бот, това няма да го спре, защото ще ти подава валиден USER\_AGENT. Също добра идея е да не се блокира с Forbbiden, а да се пренасочи към специална страница за ботове.

**забрана на достъп до сайт през прокси**

Напиши всичките тези правила и ще ограничиш достъпа до сайта от огромно количество свободни уеб прокси сайтове. Много е полезно за сайтове с форуми, където често потребителите си правят втори профил през прокси, за да се представят за друг човек.

RewriteCond %{HTTP:VIA}                 !^$ [OR]  
RewriteCond %{HTTP:FORWARDED}           !^$ [OR]  
RewriteCond %{HTTP:USERAGENT\_VIA}       !^$ [OR]  
RewriteCond %{HTTP:X\_FORWARDED\_FOR}     !^$ [OR]  
RewriteCond %{HTTP:PROXY\_CONNECTION}    !^$ [OR]  
RewriteCond %{HTTP:XPROXY\_CONNECTION}   !^$ [OR]  
RewriteCond %{HTTP:HTTP\_PC\_REMOTE\_ADDR} !^$ [OR]  
RewriteCond %{HTTP:XROXY\_CONNECTION}    !^$ [OR]  
RewriteCond %{HTTP:X-FORWARDED-FOR}     !^$ [OR]  
RewriteCond %{HTTP:FORWARDED-FOR}       !^$ [OR]  
RewriteCond %{HTTP:X-FORWARDED}         !^$ [OR]  
RewriteCond %{HTTP:HTTP\_CLIENT\_IP}      !^$  
RewriteRule ^(.\*)$ - [F]

**Сигурност**

**Много важно!** Ако ползваш на твоя сайт едно от следните: картинки против автоматични регистрации (captcha),  smarty темплейти или имаш директория, в която могат да се качват снимки или други файлове. Тогава вероятно тези директории имат повече права за писане в тях. Възможно е хакер да се възползва от тази слабост и да качи в такава директория root shell скрипт, който му дава достъп до файлове, база данни и изпълнение на команди. Пример за такива директории в WordPress са тези в  wp-content/uploads. Много добра идея е в такива директории да поставиш htaccess подобен на следния:

<Files ~ "^.\*\.(php|pl|log|bak)">

Order allow,deny

Deny from all

Satisfy All

</Files>

Това забранява достъпа до php и pl файлове качени в тази директория. Допълни в първия ред разделени с | разширенията на всички типове файлове, които не се очаква да присъстват в тази директория. Друг вариант е, ако директорията е само за картинки да изброиш типовете файлове, които се очакват да са в директорията със следния htaccess файл:

RewriteCond %{REQUEST\_URI} !(\.(gif|jpe?g|png)$)

RewriteRule .\* - [F,L]

Копирай този файл във всички директории за картинки. Сложи този файл в директорията upload на WordPress и той ще защити всички поддиректории, които се създават. В зависимост какви файлове качваш в блога, може да разшириш списъка. При мен е  !(\.(gif|jpe?g|png|zip|doc|xls|xml|html|txt)$) разбира се НЕ трябва да добавяш php иначе се губи смисъла.

Най-добрия вариант е ако може всички подобни директории да бъдат едно ниво нагоре извън главната уеб директория. Така няма да има нужда от htaccess в тях, защото дори хакер да качи файл няма да може да го достъпи през уеб. И освен това всеки път като архивираш проекта на сайта няма да копираш мегабайти излишна временна информация.

**Обработка на грешки**

Задаване на различни обработчици на възникнали грешки: служебна страница със съобщение , локална страница или файл на друг сървър.

ErrorDocument 403 "Sorry! Access denied!"

ErrorDocument 404 /404.html

ErrorDocument 500 http://drug.domain.com/error

в този пример, ако бъде написан адрес към несъществуваща страница, ще бъде зареден файла /404.html

Продължава в статията [Пренасочване на домейн](http://ganbox.com/blog/%d0%bf%d1%80%d0%b5%d0%bd%d0%b0%d1%81%d0%be%d1%87%d0%b2%d0%b0%d0%bd%d0%b5-%d0%bd%d0%b0-%d0%b4%d0%be%d0%bc%d0%b5%d0%b9%d0%bd/).

- See more at: http://ganbox.com/blog/%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%BD%D0%B0-%D0%BC%D0%B0%D0%B3%D0%B8%D1%8F-%D1%81-modrewrite/#sthash.9kVWuWkY.dpuf