htaccess note

*1.Những thủ thuật .htaccess bạn nên biết*

Dòng lệnh bắt đầu .htaccess

Dòng lệnh .htaccess của bạn nên bắt đầu như sau :

*RewriteEngine On  
RewriteBase /*

Dòng lệnh cơ bản redirect 301

Dòng lệnh redirect 301 về cơ bản có dạng như sau trong tệp tin .htaccess :

redirect 301 /old/old.htm http://www.domain.com/new.htm

Dòng lệnh trên thông báo chuyển rời tệp tin old.htm trong thư mục old đến vị trí mới tạihttp://www.domain.com/new.htm.

Chú ý : Để bắt đầu thì bạn nên redirect 301 các trang trên cũng host, bạn chỉ nên chuyển các trang cũ đến thư mục gốc tương đối. Có thể thực hiện việc này bằng việc loại bỏ “http://www.domain.com/” mà chỉ thêm đường dẫn tương đối đến thư mục gốc.

Điểm mạnh nhât của việc thay đổi hàng loạt đường dẫn URL chính là mod\_rewrite của Apache, đặc biệt là khi thay đổi tên miền hay thư mục  
hay số lượng lớn các tập tin. Chúng ta hãy tìm hiểu phần tiếp theo.

Chuyển toàn bộ đến tên miền mới

Nếu bạn quyết định thay đổi tên miền thì các đường dẫn trước đây từ các website bên ngoài, từ máy tìm kiếm và ngay cả các đường dẫn tuyệt đối trên website cũ đều bị thay đổi và khi bạn truy cập, các địa chỉ trên sẽ trỏ bạn đến trang báo lỗi 404 : trang không tìm thấy.

Vì thế nếu bạn thay đổi tên miền thì đừng để mất các liên kết quí báu và người dùng tiềm năng từ máy tìm kiếm, hãy redirect các liên kết cũ tới tên miền mới :

*Options +FollowSymLinks*

*RewriteEngine on*

*RewriteRule (.\*) http://www.domain.com/$1 [R=301,L]*

Hãy thay đổi www.domain.com bằng tên miền mới của bạn.

Redirect toàn bộ trang trong một thư mục đến mọt trang mới

Giả sử như bạn không còn sử dụng thư mục “thuthuatseo” nữa và muốn chuyển tất cả các trang trong thư mục này đến trang “seoblog.php” thì hãy thêm dòng lệnh sau vào tệp tin .htaccess nằm tại thư mục gốc :

*RewriteRule ^thuthuatseo(.\*)$ /seoblog.php [L,R=301]*

Chuyển các trang động tới một trang mới

Để chuyến trang web động page.php?id=n (với n là giá trị biến) tới một trang tĩnh mới new-page.htmlthì bạn có thể làm như sau :

*RewriteRule ^page.php?id=(.\*)$ /new-page.htm [L,R=301]*

URL với www hay không www

Đôi khi các bạn bắt gặp việc website sử dụng “www” trong URL, trong khi có những website lại không dùng “www” này. Nên nhớ rằng “tương ứng” với dịch vụ Web. Các bạn có thể lựa chọn riêng cho mình. Nhưng không thể chọn cả hai vì sẽ gây ra trùng lặp nội dung. sau đây là hai ví dụ, hướng dẫn bạn xử lý triệt để vấn đề này.

Trường hợp sử dụng www

Options +FollowSymLinks

*RewriteEngine On*

*RewriteBase /*

*RewriteCond %{HTTP\_HOST} !^www\.gocit.\.net$ [NC]*

*RewriteRule ^(.\*)$ http://www.domain.com/$1 [R=301,L]*

Ngoài ra còn có cách viết sau có thể áp dụng cho tất cả các host và domain, không phải edit lại :

*Options +FollowSymLinks*

*RewriteEngine On*

*RewriteBase /*

*RewriteCond %{HTTP\_HOST} !^www\.[a-z-]+\.[a-z]{2,6} [NC]*

*RewriteCond %{HTTP\_HOST} ([a-z-]+\.[a-z]{2,6})$     [NC]*

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

Trường hợp không sử dụng www

*Options +FollowSymLinks*

*RewriteEngine On*

*RewriteBase /*

*RewriteCond %{HTTP\_HOST} !^aevn\.fr$ [NC]*

*RewriteRule ^(.\*)$ http://aevn.fr/$1 [R=301,L]*

Bạn có thể thử lại với các tên miền ví dụ trên mà tớ áp dụng thành công đúng như trích dẫn.

Loại bỏ Query\_String

Đôi khi một trang web của bạn có thể hiển thị hai đường dẫn url khác nhưng ( cùng một nội dung) ví dụ seo-tools.php và seo-tools.php?v=mobile. Tương tự như phần trên, điều này sẽ tạp ra nội dung trùng lặp trên website của bạn. Vì vậy bạn có thể loại bỏ các tham biến như sau :

*RewriteCond %{THE\_REQUEST} ^GET\ /.\*\;.\*\ HTTP/*

*RewriteCond %{QUERY\_STRING} !^$*

*RewriteRule .\* http://www.domain.com%{REQUEST\_URI}? [R=301,L]*

Ngoài ra nếu như tham biến QUERY\_STRING không được gấn giá trị nào hết và URl kết thúc bởi biến rỗng “?”, thì bạn nên loại bỏ chúng, ví dụ “index.php?” :

*RewriteEngine On*

*RewriteBase /*

*RewriteCond %{QUERY\_STRING} .*

*RewriteRule ^index.php /index.php? [L]*

Chuyển phần đuôi mở rộng từ .php sang .html

Việc chuyển .php sang .html có tác dụng khi bạn cần viết lại đường dẫn thân thiện với máy tìm kiếm (Friendly URL for SEO).

*RewriteRule ^(.\*)\.html$ $1.php [R=301,L]*

Chuyển gạch dưới (\_) thành gạch ngang(-)

Dù [Google](http://www.gocit.vn/bai-viet/tag/google/) mới chấp nhận việc sử dụng gạch dưới “\_” như là ký tự ngăn cách, nhưng trong thực tế người dùng vẫn quen thuộc với dấu gạch ngang “-”. Bản thân vietprotocol cũng ủng hộ dấu gạch ngang vì nó dễ nhìn hơn. Bạn có thể chuyển đổi toàn bộ dấu “\_” sang “-” như sau :

*Options +FollowSymLinks*

*RewriteEngine On*

*RewriteBase /*

*RewriteRule !\.(html|php)$ - [S=4]*

*RewriteRule ^([^\_]\*)\_([^\_]\*)\_([^\_]\*)\_([^\_]\*)\_(.\*)$ $1-$2-$3-$4-$5 [E=uscor:Yes]*

*RewriteRule ^([^\_]\*)\_([^\_]\*)\_([^\_]\*)\_(.\*)$ $1-$2-$3-$4 [E=uscor:Yes]*

*RewriteRule ^([^\_]\*)\_([^\_]\*)\_(.\*)$ $1-$2-$3 [E=uscor:Yes]*

*RewriteRule ^([^\_]\*)\_(.\*)$ $1-$2 [E=uscor:Yes]*

*RewriteCond %{ENV:uscor} ^Yes$*

*RewriteRule (.\*) http://www.domain.com/$1 [R=301,L]*

Redirect [*WordPress*](http://blog-xtraffic.pep.vn/plugin-tang-toc-wordpress/) Feeds tới Feedburner

Trong bài viết sử dụng Feedbuner, các bạn có thể sử dụng plugin để quản lý Feeds RSS trên Blog WordPress. Nếu không các bạn có thể sử dụng code htaccess sau :

*RewriteEngine On*

*RewriteBase /*

*RewriteCond %{REQUEST\_URI} ^/feed\.gif$*

*RewriteRule .\* - [L]*

*RewriteCond %{HTTP\_USER\_AGENT} !^.\*(FeedBurner|FeedValidator) [NC]*

*RewriteRule ^feed/?.\*$ http://feeds.domain.com/support [L,R=302]*

*RewriteCond %{REQUEST\_FILENAME} !-f*

*RewriteCond %{REQUEST\_FILENAME} !-d*

*RewriteRule . /index.php [L]*

Đối với người dùng WordPress, các bạn có thể sử dụng plug-in Redirection Permanent Link để chuyến hướng các trang.

Bào vệ ăn cắp tài nguyên và băng thông

Đôi khi các tài liệu, film hay hình ảnh mà bạn tải trên host của mình bị người khác dùng và đăng trên website khác sẽ gây tốn băng thông, bạn có thể hạn chế việc ăn cắp bằng cách chỉ cho phép các tài nguyên trên tiếp cận trên những website bạn cho phép như sau (Nếu không sẽ trả về thư mục /feed/) :

*RewriteEngine On*

*RewriteBase /*

*RewriteCond %{HTTP\_REFERER} !^$*

*RewriteCond %{HTTP\_REFERER} !^http://(www\.)?domain.com/.\*$ [NC]*

*RewriteRule \.(gif|jpg|swf|flv|png)$ /feed/ [R=302,L]*

Ứng dụng redirect 301 trên các máy chủ khác (ngoài Apache)

IIS redirect

Trong phần quản trị dịch vụ internet, nhấn phải chuột chọn tệp tin và thư mục mà bạn muốn áp dụng redirect;  
Chon nút “a redirection to a URL”;  
Chọn trang Redirection;  
Chon tiếp “The exact url entered above” và “A permanent redirection for this resource”;  
Chon “Apply”.

ColdFusion Redirect

*<.cfheader statuscode="301" statustext="Moved permanently">*

*<.cfheader name="Location" value="http://www.domain.com">*

PHP Redirect

ASP Redirec

*<%@ Language=VBScript %>*

*<% Response.Status="301 Moved Permanently" Response.AddHeader "Location","http://www.domain.com/" %>*

ASP .NET Redirect

JSP (Java) Redirect

*<% response.setStatus(301); response.setHeader( “Location”, “http://www.domain.com/” ); response.setHeader( “Connection”, “close” ); %>*

CGI PERL Redirect

*$q = new CGI;*

*print $q->redirect("http://www.domain.com/");*

Ruby on Rails Redirect

*def old\_action  
headers[“Status”] = “301 Moved Permanently”  
redirect\_to “http://www.domain.com/”*

*2.Tối ưu và bảo mật website bằng file .htaccess*

Sử dụng file .htaccess để tối ưu và bảo mật website của bạn hơn: bảo vệ các file quan trọng, giới hạn upload, chuyển hướng 301, trang lỗi tùy chỉnh, tắt liệt kê nội dung thư mục, nén nội dung…

*Chức năng:*  
1. Bảo vệ tệp tin  
2. Tắt chữ ký số của web server  
3. Giới hạn upload  
5. Giới hạn quyền truy cập  
6. Trang lỗi tùy chỉnh  
7. Tắt liệt kê nội dung thư mục  
8. Chuyển hướng 301  
9. Chống ăn cắp băng thông như hình ảnh, video (bandwidth)  
10. Bật nén PHP compression (bandwidth)  
11. Thiết lập canonical url cho website

1.Bước 1, Tạo file .htaccess.

Mở Notepad hoặc phần mềm text editor. Lưu và Đặt tên file htaccess.txt.  
2. Nhập nội dung vào file htaccess.txt.

# Bảo vệ file file htaccess

*<files .htaccess>  
order allow,deny  
deny from all  
</files>*

#Tắt chữ ký số của web server

*# disable the server signature  
ServerSignature Off*

*# limit file uploads to 10mb  
LimitRequestBody 10240000*

# Bảo vệ file cấu hình config.php

*<files config.php>  
order allow,deny  
deny from all  
</files>*

# Giới hạn truy cập, chặn IP

*order allow,deny  
#deny from 000.000.000.000  
allow from all*

# Tạo Trang lỗi tùy chỉnh

*# custom error docs  
ErrorDocument 404 /notfound.php  
ErrorDocument 403 /forbidden.php  
ErrorDocument 500 /error.php*

# Tắt liệt kê nội dung thư mục

*Options All -Indexes*

# Chuyển hướng 301

*Redirect 301 /old.php http://www.domain.com/new.php*

# Khóa tên miền tham chiếu

*RewriteEngine on  
RewriteCond %{HTTP\_REFERER} digg\.com [NC]  
RewriteRule .\* – [F]*

Chống ăn cắp băng thông

#disable hotlinking of images with forbidden or custom image option  
RewriteEngine on  
RewriteCond %{HTTP\_REFERER} !^$  
RewriteCond %{HTTP\_REFERER} !^http://(www\.)?domain.com/.\*$ [NC]  
#RewriteRule \.(gif|jpg)$ – [F]  
#RewriteRule \.(gif|jpg)$ [h](http://www.yourdomain.com/stealingisbad.gif)ttp://www.domain.com/images.gif [R,L]  
Nén file

*# php compression – use with caution  
<ifmodule mod\_php4.c>  
php\_value zlib.output\_compression 16386  
</ifmodule>*

Tránh trùng lặp nội dung

*# set the canonical url  
RewriteEngine On  
RewriteCond %{HTTP\_HOST} ^yourdomain\.com$ [NC]  
RewriteRule ^(.\*)$ http://www.yourdomain.com/$1 [R=301,L]*

# Chống spam comments

*RewriteEngine On  
RewriteCond %{REQUEST\_METHOD} POST  
RewriteCond %{REQUEST\_URI} .comments-post\.php\*  
RewriteCond %{HTTP\_REFERER} !.\*yourdomain.com.\* [OR]  
RewriteCond %{HTTP\_USER\_AGENT} ^$  
RewriteRule (.\*) ^http://%{REMOTE\_ADDR}/$ [R=301,L]*

3. Upload htaccess.txt lên server.

Sử dụng ftp client (sử dụng ASCII mode) và đổi tên file là .htaccess.

CHMOD file .htaccess sang 644.

4. Kiểm tra kết quả.

Apache cung cấp khả năng cấu hình qua những files truy cập siêu văn bản .htaccess., cho phép thay đổi tinh chỉnh của Apache ([httpd](http://www.gocit.vn/bai-viet/tag/httpd/).conf). Đảm bảo file này được ấn định một mức độ bảo mật nhất định, tránh những truy cập bất hợp pháp từ bên ngoài (xem, sửa, xóa) những tinh chỉnh này.

1.Định nghĩa .htaccess:

Để làm được như vậy, trước tiên bạn cần chmod file .htaccess là 644, về chmod bạn có thể tìm hiểu ở nhiều tài liệu, hoặc google.  
.htaccess có tác dụng trên thư mục hiện hành (nơi chưa nó) và tất cả các thư mục con. Vì vậy, muốn sử dụng .htacess để quản lí truy cập vào website của mình, đơn giản bạn chỉ việc soạn thảo một file .htaccess, và lưu nó vào thư mục [root](http://www.gocit.vn/bai-viet/tag/root/) của website.

2.Comment trong .htaccess:

Nếu đã sử dụng và một lần sửa đổi cấu hình Apache thì chắc chắn bạn cũng biết, muốn sử dụng một module nào đó trong những available modules, bạn phải xóa dấu “#” (hash). Dấu “#” này dùng để định nghĩa một đoạn comment trong httpd.conf, và .htaccess cũng vậy. Một ví dụ:

CODE  
#Enable Rewrite mod << đây là comment, dòng này sẽ không được thực thi #trong .htaccess  
RewriteEngine on

3.Những chú ý quan trọng:.htaccess là một công cụ rất mạnh trong việc quản lí website. Một lỗi cú pháp nhỏ (thậm chí là một khoản trắng) cũng khiến website của bạn hoạt động không được bình thường như trước. Sau khi áp dụng .htaccess, bạn nên kiểm tra một lượt website của mình xem .htaccess có gây ra lỗi ở phần, trang con nào không? Do vậy trước khi làm việc với .htaccess, bạn nên backup cẩn thận dữ liệu của mình, nếu có trục trặc xảy ra, việc khôi phục trở nên đơn giản.

4.Vấn đề về performance:  
.htaccess cho phép bạn cấu hình mà không cần động chạm vào cấu hình chính của Apache. Tuy nhiên, về mặt hiệu năng và bảo mật, nếu có thể, bạn nên sử dụng httpd.conf hơn là sử dụng .htaccess. Cụ thể, khi được cấu hình để sử dụng .htaccess, thi Apache sẽ tìm kiếm tất cả những folder có chứa .htaccess để thực thi, và nó sẽ thực thi tất cả những file .htaccess tìm được. Do vậy, sẽ làm website của bạn trở nên ì ạch một cách không cần thiết. Nếu website của bạn là website cá nhân, hoặc ít người truy cập, thì đó là chuyện nhỏ. Nhưng nếu là một website đại chúng (báo điện tử, trang nhạc, film..) thì đây là một vấn đề rất lón. Do vậy .htaccess chỉ có tác dụng trong trường hợp bạn không có quyền cấu hình trên httpd.conf của Apache.

5.Những kí tự đặc biệt sử dụng khi cấu hình .htaccess

#  
Như đã nói ở trên, # đặt trước một dòng để định nghĩa dòng này là một đoạn comment, dòng này sẽ không được thực thi cho đến khi bạn xóa dấu “#” đi. Khi comment, bạn chỉ nên sử dụng các chữ cái, số, dấu gạch ngang hoặc gạch dưới để tránh những lỗi cú pháp không cần thiết.

[F]

Forbidden: Kí tự này dùng để chỉ định server sẽ trả về client trang lỗi 403 nếu truy cập vào những nơi không được phép

[L]  
Last rule: kí tự này để ấn định rằng bước trước đã xong thì ngừng và không tiếp tục thực thi lệnh rewrite tiếp theo nữa.

[N]

Next: chỉ thị cho server tiếp tục rewrite cho đến khi tất cả các mục chỉ định được rewrite hoàn toàn.

[G]

Gone: Chỉ định server trả về client trang báo lỗi không tồn tại (no longer exit)

[P]

Proxy: chỉ định server điểu kiển các yêu cầu được ấn định bởi mod\_proxy

[C]

Chain: Chỉ định server thực hiện rule hiện hành song song với rule trước đó

[R]

Redirect: chỉ định server đổi hướng request sang một trang khác trong trường hợp trình duyệt gởi yêu cầu duyệt một trang được sửa chữa đường dẫn (rewrite) trước đó.

[NC]

No-case: Chỉ định server match những kí tự tương tự nhau, ví dụ dòng lệnh sau:

CODE  
RewriteCond %{HTTP\_REFERER} !^http://domain.com/.\*$ [NC]  
#Thì referrer là domain.com || Domain.com || DoMain.com…đều được match.

[PT]

Pass Through: có nghĩa là dùng kí tự để buộc “rewrite engine” ấn định bảng giá trị của uri trở thành giá trị của tên files

[OR]

Đây có thể gọi là một toán tử, các biểu thức được nối tiếp nhau bằng toán tử OR, trong đó biểu thức nào đúng, sẽ là quy tắc để lệnh Rewrite áp dụng.  
Ví dụ:

*CODE  
RewriteBase /  
RewriteCond %{HTTP\_USER\_AGENT} ^Anarchie [OR]  
RewriteCond %{HTTP\_USER\_AGENT} ^ASPSeek [OR]  
RewriteCond %{HTTP\_USER\_AGENT} ^attach [OR]  
RewriteRule ^.\* – [F,L]*

Như vậy, mỗi dòng Rewritecond là một biểu thức chỉ định USER\_AGENT, nếu bất cứ USER\_AGENT nào match, thì RewriteRule sẽ được áp dụng, sau đó ngừng.  
Vì sao ngừng: Một request chỉ có một USER\_AGENT, vì vậy không cần thiết phải quay trở lại kiểm tra các biểu thức còn lại, sẽ làm giảm hiệu năng phục vụ.  
Lời bàn:như ở trên đã nói, bạn chỉ sử dụng những chức năng này của .htaccess nếu không có quyền cấu hình trên Apache và firewall. Những chức năng lọc và chặn User agents Mod\_[security](http://www.gocit.vn/bai-viet/tag/security/) và Iptables đều có thể thực hiện được, vì vậy nếu đã có rule cho iptables và Mod\_[security](http://www.gocit.vn/bai-viet/tag/security/), việc thêm rule trên cho .htaccess là dư thừa, và không cần thiết.

[NE]

No Escape: Chỉ định server xử lí các gói tin trả về mà không dùng kí tự thoát

[NS]

No Subrequest: Chỉ định server bỏ qua thư mục hiện hành nếu request nhắm vào thư mục con.

[QSA]

Append Query String: chỉ định server gắn chuỗi truy vấn vào cuối cùng của URL

[S=x]

Skip: Chỉ định server bỏ qua không xem xét các X rules tiếp theo nữa nếu một rule đã được thực thi.

[T=MIME-type]

MIME-type: Khai báo định dạng files của server

[]

Chỉ định các kí tự đặt đặt trong [] sẽ được match. Cụ thể [xyx] thì các request có mang theo các kí tự x, y hoặc z sẽ được match.

[]+

Tương tự như trên, nhưng trong trường hợp này, nếu ta đặt [xyz]+ thì tất cả các request có mang theo kí tự x, y, z hoặc x’s, y’s. z’s hoặc sự kết hợp của x,y,z với bất kì kí tự nào đều được match.  
[^]

Ngược lại với [], các kí tự được đặt theo dạng [xyz^] sẽ không được match.

[a-z]

Dấu ‘-‘ đặt giữa 2 kí tự bên trong cặp dấu [] sẽ match tất cả các kí tự nằm trong khoảng từ kí tự đầu đến kí tự cuối. Ví dụ [a-bA-B] sẽ match tất cả các kí tự thường vào in hoa abcd và ABCD.

a{n}

Sẽ chỉ định chính xác n kí tự a được match. Ví dụ a{3} thì 3 kí tự a sẽ được match.

a{n,}

Tương tự như trên, nhưng trong trường hợp này sẽ match các kí tự a từ na trở lên.

a{n,m}

Chỉ định một khoảng chính xác các kí tự (m-n)a sẽ được match.

()

Nhóm các phương thức truyền dữ liệu (method). VD:

CODE  
RewriteCond %{REQUEST\_METHOD} ^(HEAD|TRACE|DELETE|TRACK) [NC,OR]  
Các request bắt đầu (^) HEAD, TRACE, DELETE hoặc TRACK sẽ được match.

^

Bắt đầu bằng (xem ví dụ ở trên).  
$

Kết thúc bằng ( .\*$: kết thúc bằng bất cứ gì)  
?

Thường dùng chung với (), như trong trường hợp test(ti)? thì test hoặc ti sẽ được match.

!

Phủ định. VD: “!string” sẽ match tất cả các kí tự, ngoại trừ string  
–

Loại đi một chức năng, hoặc một rule đang thực thi nào đó. VD:  
CODE  
# không cho xem chỉ mục trong một folder nào đó  
Options All –Indexes

+

Chỉ định 1 hoặc nhiều kí tự cho trước sẽ được match. Hoặc thêm một chức năng nào đó VD: G+ sẽ match tất cả các kí tự G, Gs, Gx, G..

\*

Tất cả đều được match. Ví dụ:  
CODE  
RewriteCond %{HTTP\_REFERER} !^http://domain.com/.\*$

Thì tất cả request files không mở đầu bằng \_http://domain.com đều được match.

|

Hoặc. Ví dụ x|y thì hoặc x hoặc y sẽ được match.

\

Kí tự thoát. VD: www\.domainname\.com Chỉ có www được match, chứ không có wwww hoặc wwwww, domainname chứ không có domainnames hoặc domainnamex…  
-d

Kiểm tra thư mục có tồn tại hay không?  
-f

Kiểm tra file có tồn tại hay không

-s

Kiểm tra giá trị của file có khác 0 hay không?

6. Mã chuyển hướng trang báo lỗi

– 401 – Authorization Required : Lỗi chưa được xác thực user và [password](http://www.gocit.vn/bai-viet/tag/password/)  
– 400 – Bad request : Lỗi truy cập không hợp lệ, do server không chấp nhận request, hoặc request thiếu một số yêu cầu nào đó.  
– 403 – Forbidden : Lỗi truy cập vào trang bị cấm  
– 500 – Internal Server Error : Lỗi do server  
– 404 – Wrong page : Lỗi truy cập vào trang không tồn tại  
-301 – Moved Permanently: 302 – Moved Temporarily: Lỗi truy cập vào trang đã bị di chuyển.

*3. Cách Rewrite URL trong PHP – Tạo đường dẫn thân thiện cho website*

Nhiều bạn thiết kế website hỏi làm sao tạo được các đường dẫn đẹp (URL Rewrite). Trước tiên chúng ta tìm hiểu cơ chế rewite: Rewrite Engine là một phần dịch vụ của Webserver được dùng để thay đổi URL(đường dẫn trên trình duyệt) sang một dạng khác với nhiều mục đích khác nhau.

Kỹ thuật được nói đến là URL rewriting và nó đem lại những tiện lợi sau:- Làm cho URL trở nên thân thiện với người dùng cũng như dễ dàng với các Search Engine (bộ máy tìm kiếm)- Tránh lộ các đường link quan trọng- Tránh lộ các công việc bên trong trước người dùng

Bản chất của việc rewrite này các bạn có thể hình dung như sau: khi bạn gõ 1 đường link lên trình duyệt như sau: http://vidu.com/news/2007/13/01 thì đây là đường dẫn ảo, và khi request đến Webserver nó sẽ đổi lại thành đường dẫn thật như sau: http://vidu.com/index.php?mod=news&y…ay=13&month=01 << có thể là như thế này vì có nhiều cách rewrite.

Đối với Server Apache: để thực hiện được URL Rewriting bạn làm các bước sau:

1. Kích hoạt mod mod\_rewrite có sẵn trong Apache bẳng cách sửa trong file httpd.conf, bạn Find đến dòng chưa rewrite\_module rồi bỏ dấu # (nếu có) ở đầu đi. Sau đó restart Apache. Nếu bạn dùng các host trên internet thì có thể đã có sẵn rồi.

2. Tạo 1 file .htacces có nội dung như sau:  
Đầu tiên bật Rewrite Engine:

PHP Code:

*RewriteEngine on  
Options +FollowSymlinks  
RewriteBase /*

Tiếp đến là viết cú pháp RewriteRule:

PHP Code:

*RewriteRule ^Virtual\_Path Real\_Path*

Ví dụ:

PHP Code:

*RewriteRule ^topic\_([0-9]\*).html index.php?topic=$1*

ví dụ: topic\_1.html thành index.php?topic=1

PHP Code:

*RewriteRule ^article/([0-9]\*)/page/([0-9]\*) index.php?article=$1&page=$2*

ví dụ: article/1/page/2 thành index.php?article=1&page=2

Cú pháp để viết các Rule này giống như cách viết của biểu thức chính quy Regular Expression. ví dụ như sau:

(.\*) nhận tất cả các giá trị.  
([0-9]\*) chỉ nhận các giá trị là số, từ 0-9.  
([a-zA-Z]\*) chỉ nhận các giá trị là chữ cái, từ a-z và A-Z.

3. Sau khi tạo được file htaccess bạn copy nó vào thư mục gốc của web của bạn là có thể chạy được.

Đối với Server IIS: các bạn download gói cần thiết về để có thể thực hiện được việc rewrite, các bạn có thể download từ 1 số trang sau:

<http://www.micronovae.com/ModRewrite/ModRewrite.html>  
<http://www.qwerksoft.com/products/iisrewrite/>  
<http://www.isapirewrite.com/>  
[http://www.motobit.com/help/url-repl…od-rewrite.asp](http://www.motobit.com/help/url-replacer-rewriter/iis-mod-rewrite.asp)

Ngoài việc sử dụng mod url\_rewrite như trên thì nếu host của bạn ko có kích hoạt mod này hoặc bạn ko có quyền để cấu hình nó như trên. Để đơn giản các bạn có thể tạo ra những cách khác nhau để làm cho URL trở nên đẹp và thân thiện hơn. Dưới đây là 1 số cách khác nhau.

c1 dạng như sau: http://vidu.com/index.php/news/2007/13/01 << Trông rất giống với kiểu rewrite ở trên nhưng nó khác ở chỗ nó là đường dẫn thật còn url rewrite là đường dẫn ảo. Đường dân này hiểu như là 1 request đến file index.php với Query String (Xâu truy vấn) là /news/2007/13/01, các bạn tách xâu này ra rồi sử dụng theo mục đích của mình. Cách này đang được trang web 24h.com.vn sử dụng.

c2 dạng như sau: http://vidu.com/index.php/news\_2007\_13\_01.html << Gần giống cách 1 ở trên nhưng đường dẫn khác thân thiện với đuôi html, dễ dàng cho các search engine (máy tìm kiếm như Google, Yahoo) tìm kiếm nội dung.

c3 có thể viết như sau: http://vidu.com/index.php?q=news/2007/13/01 << cũng gần giống cách 1 nhưng nó rõ ràng hơn, nó được dùng trong FrameWork Drupal của PHP.

Và có thể có nhiều cách khác nhau các bạn có thể tự nghĩ ra. Hoặc có thể dùng cách mã hoá xâu truy vấn chẳng hạn.