**Giới thiệu**

Thông thường để cấu hình cho Server Linux thì ta phải có kiến thức về quản trị mạng Linux, điều này đôi khi gây khó khăn cho các lập trình viên. Vậy có cách nào khắc phục vấn đề này mà không cần đăng nhập vào tài khoản phía Server?

Ở Server có một file đặc biệt tên là .htaccess, file này có nhiệm vụ khai báo cấu hình cho các folder kể từ folder mà nó đang đứng. Ví dụ bạn đang ở folder sources/.htaccess thì tất cả các folder con của folder sources cũng bị ảnh hưởng bởi file này. Bạn có thể khai báo cấm người dùng truy cập vào folder, đặt mật khẩu cho folder, và đặc biệt hơn nữa đó là quản lý đường dẫn như rewrite URL, redirect URL. Đó là đối với Linux, còn đối với Window thì file này sẽ có tên là web.config và tác dụng của nó cũng tương tự, tuy nhiên về cú pháp sử dụng thì hai file .htaccess và web.config sẽ khác nhau.

File .htaccess sử dụng cú pháp của Regular Expression nên để sử dụng thành thạo thì bắt buộc ban phải biết các quy tắc căn bản trong RegExp, bạn không cần phải biết sâu mà chỉ cần biết các cú pháp RegExp căn bản là được. Ngoài ra nó còn có cú pháp riêng mà ban phải học thuộc để biết mà điều hướng đúng cách.

Như vậy công dụng của file .htaccess trong một project PHP là giúp khai báo cấu hình cho project đó như viết lại đường dẫn, bảo mật file, đặt mật khẩu cho file, bật cache client để tối ưu hóa tốc độ cho website. Và trong series này sẽ tổng hợp tất cả các vấn đề này, mời các bạn theo dõi.

**Bài 01: File htaccess là gì?**

**1. Định Nghĩa File .htaccess là gì?**

File **.htaccess** (hay còn gọi là  là **Hypertext Access)** của Apache là một tập tin dùng để cấu hình máy chủ web　server chạy Apache. Nó được máy chủ chấp nhận như là một thành phần và cho phép chúng ta thực hiện điều hướng và bật các tính năng một cách linh hoạt hoặc bảo vệ một phần (folder) nào đó của trang web. Đồng thời .htaccess dùng để thiết lập các tùy chọn: thực thi hay loại bỏ các chức năng, tính năng của Apache. Vì vậy, 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 của website.

Trong cái tên .htaccess thì htaccess là phần đuôi và tập tin này là không có tên (noname), chính vì thế khi bạn đưa file này lên host không phải lúc nào nó cũng hiển thị ra.  Ví dụ bạn dùng Total Commander để xử lý file FTP thì bạn sẽ không thể thấy file .htaccess được.

Để**thao tác với file .htaccess** ta có thể sử dụng bất kì một editor nào hỗ trợ code text như Netbeans, PHP Designer, hay thậm chí là phần mềm notepadd có sẵn trong window.

File .Htaccess không được viết nên từ một ngôn ngữ lập trình nào, nó được viết bởi những quy tắc Regular Expression nên nếu bạn không nắm vững kiến thức này thì thao tác với file .htaccess rất khó khăn. Và nếu bạn chỉnh sửa file với một lỗi dù chỉ là nhỏ thì khi truy cập vào website của bạn sẽ **xuất hiện lỗi 500**, như vậy là bạn đã HACK được web mình rồi đấy. Vì thế hãy thận trọng khi chỉnh sửa file .htaccess và hãy luôn luôn có một file backup trước khi đụng đến.

**2. Một vài quy tắc trong file .htaccess**

Sau phần định nghĩa **file .htaccess là gì?**thì ta sẽ tìm hiểu một số **quy tắc và ứng dụng của file htaccess**.

* Một lỗi cú pháp nhỏ (thậm chí là một khoản trắng) trong cấu hình .htaccess cũng khiến website của bạn hoạt động không được bình thường như trước.Do vậy trước khi điều chỉnh .htaccess, bạn nên backup cẩn thận dữ liệu, nếu có trục trặc xảy ra, việc khôi phục trở nên đơn giản. Nếu file .htaccess bị lỗi thì kết quả sẽ trả về client là lỗi 500.
* Bạn phải 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 thông tin cấu hình này. Để làm được như vậy, bạn nên chmod file .htaccess là 644, về chmod bạn có thể tìm hiểu ở nhiều tài liệu ngoài Internet.
* Trường hợp bạn không có quyền cấu hình trên httpd.conf của Apache thì việc sử dụng .htaccess rất hữu ích cho nhu cầu của bạn.
* Có thể dùng ký tự # để đặt trước một dòng mà bạn muốn nó không chạy, tức là chuyển dòng đó sang chế độ chú thích.
* File .htaccess sẽ có tác dụng với thư mục hiện tại và các thư mục con của thư mục hiện tại.
* Có thể sử dụng file .htaccess để viết lại đường dẫn, bảo vệ thư mục và file, bật chức năng cache và gzip giúp tốc độ trang web được cải thiện hơn. Và còn nhiều điều nữa mà tôi không thể biết hết được.

**3. Công dụng phổ biến của .htaccess**

Một số công dụng phổ biến của file .htaccess

* Authorization, authentication
* Rewrite URLs
* Blocking
* SSI
* MIME types
* Tùy chỉnh Error Page
* Kiểm soát Cache

**Ưu nhược điểm của .htaccess**

**Ưu điểm**

* Thực thi ngay lập tức: Vì htaccess được đọc trên mọi yêu cầu , những thay đổi được thực hiện trong những tập tin này có hiệu lực ngay lập tức - như trái ngược với các tập tin cấu hình chính mà đòi hỏi các máy chủ được khởi động lại để các thiết lập mới có hiệu lực.
* Hỗ trợ người dùng không có đặc quyền: cho phép người dùng cá nhân có khả năng thay đổi cấu hình trang web của họ trong khi các tập tin cấu hình máy chủ chính không cần phải được thay đổi.

**Nhược điểm**

* Giảm hiệu suất: cụ thể, khi được cấu hình để sử dụng .htaccess, thì 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.
* Bảo mật: việc cho phép người dùng cá nhân thay đổi cấu hình của một máy chủ có thể gây ra vấn đề liên quan đến bảo mật nếu không được thiết lập đúng cách.

**Lời Kết**

Đây là bài giới thiệu**file .htaccess là gì**rất căn bản và chưa thực sự đầy đủ nên nếu bạn muồn tìm hiểu sâu hơn thì lên mạng có rất nhiều site tiếng anh nói vấn đề này.

**Bài 02: Kiểm soát và bảo mật website với htaccess**

Trong phần này ta sẽ tìm hiểu một số quy tắc giúp trang web được an toàn, kiểm soát được website và tìm hiểu một số tính năng căn bản tốt cho SEO.

***Lưu ý: Trong các ví dụ đoạn code nào được bôi xanh là đoạn code ta có thể chỉnh sửa theo ý mình***

**Bật chức năng Rewrite**

Đây là bước đầu tiên để ta thao tác với fiel .htaccess. Để bật chức năng rewrite này ta chỉ cần thêm một đoạn code nhỏ lên đầu file .htacess như sau:

|  |
| --- |
| Options +FollowSymlinks  RewriteEngine on |

**Xử lý Chuyển hướng lỗi**

Một trong những ứng dụng phổ biến của htaccess là xử lý các lỗi như lỗi không tìm thấy dữ liệu hoặc lỗi không theo được. Các lỗi này được thể hiện thông qua các con số mà máy chủ đáp trả. Lỗi thông dụng nhất là **lỗi không tìm thấy dữ liệu 404.**Lỗi này sẽ xuất hiện tùy theo trình duyệt nhìn rất là khó chịu, điều này đối với các bộ máy tìm kiếm như google rất ghét. Để giải quyết vấn đề này ta có thể sử dụng file .htaccess để chuyển hướng. Điều này rất quan trọng với SEO vì:

1. Nếu ta chuyển hướng tốt với báo cáo 301, tức là thông báo **không có gì ở link vừa vào**và chuyển hướng qua link khác, như vậy sẽ công cụ search engines sẽ hiểu đây không phải là url không tồn tại.
2. **Thân thiện với người dùng**vì nếu ta có một trang thông báo lỗi thì người dùng sẽ hiểu họ đang đi vào một đường dẫn không tồn tại, cộng với một số hướng dẫn quay trở lại ở trang thông báo lỗi sẽ giúp người dùng cảm thấy dễ chịu hơn.

Để điều hướng các trang lỗi đương nhiên trước tiên ta phải tạo một số trang thông báo lỗi. Điều này đối với một số CMS như wordpress hay các Framwork như Codeigniter có sẵn cho chúng ta nhưng không đẹp lắm.

**Ví dụ file chuyển hướng lỗi 404**

|  |
| --- |
| Options +FollowSymlinks  RewriteEngine on  ErrorDocument 404 http://www.example.com/404.html |

Đoạn code trên từ khóa ErrorDocument có ý nghĩa là nếu url request bị lỗi  thì xử lý chuyển hướng. Ở đây lỗi có hiệu lực là 404 và link  chuyển hướng là  *http://www.example.com/404.html*.

Ví dụ trên tôi đã dùng một địa chỉ đầy đủ (*tức là địa chỉ có full url*) nhưng nếu ta để ở dạng local thì hay hơn. Ví dụ:

|  |
| --- |
| ErrorDocument 401 /error\_401.html  ErrorDocument 403 /error\_403.html  ErrorDocument 404 /error\_404.html  ErrorDocument 400 /error\_400.html |

Các con số lỗi trên có ý nghĩa như sau:

* 404 - Not Found
* 401/403 - Unauthorized/Forbidden
* 400 - Bad Request
* 500 - Internal Server Error

Và còn rất nhiêu lỗi khác tôi không thể liệt kê hết được. Nhưng chung quy thì lỗi 404 vẫn là lỗi thông dụng nhất nên các website hiện nay chủ yếu tập trung vào lỗi này.

**Hiển Thị Trang Index Giả Mạo**

Nếu bạn vào một folder nào đó trong website mà không có file index.html thì lúc này tất cả các file hay folder đề bị thấy hết, điều này không tốt lắm vì lộ hết các files và folders. Giải quyết vấn đề này ta dùng file .htaccess với dòng lệnh:

|  |
| --- |
| Options All -Indexes |

Dòng lệnh này sẽ hiển thị một thông báo  Forbidden thay vì hiển thị cấu trúc folder.

**Thiết lập trang index mặc định**

Nếu bạn muốn thiết lập file mặc định thay vì file index.html thì ban bạn thêm đoạn code sau vào file .htaccess:

|  |
| --- |
| DirectoryIndex newindex.html |

Trong đó newindex.html là tên file bạn muốn thiết lập.

**Bảo Vệ File .htaccess**

Như bạn biết file htaccess là một file đặc biệt dùng để cấu hình Server một cách dễ dàng, sẽ rất nguy hiểm nếu hacker tấn công vào file này của bạn. Để bảo vệ fiel này ta dùng đoạn code sau:

|  |
| --- |
| <Files .htaccess>  Order Allow,Deny  Deny from all  </Files> |

Trong đó file .htaccess là tên file muốn bảo vệ. Nội dung bên trong có nghĩa không cho người nào truy cập đến file này.

Không phải chỉ bảo vệ được file .htaccess mà ta có thể bảo vệ một file bất kì. Ví dụ dưới đây tôi**bảo vệ file wp-config.php trong Wordpress**.

<Files wp-config.php>

Order Allow,Deny

Deny from all

</Files>

**Đặt mật khẩu cho folder**

Đôi lúc  ta cần đặt mật khẩu bảo vệ một file nào đó, tức là muốn vào file đó thì phải nhập username và password. Để làm được điều này các bạn phải tạo một file với tên **.htpasswd** tại folder gốc của file bạn muốn bảo vệ sau đó điền nội dung sau:

MyUsername:MyPassword

Trong đó MyUsrename là tên username, MyPassword là mật khẩu, hai giá trị này bạn có thể thay đổi bất kì.

Tiếp theo ở file .htaccess bạn thêm đoạn code sau:

AuthName "Restricted Area"

AuthType Basic

AuthUserFile /demo/.htpasswd

AuthGroupFile /dev/null

<Files file1.php>

require valid-user

</Files>

Trong đó AuthName "Restricted Area"   là tên chương trình.  /demo/.htpasswd là đường dẫn đến file htpasswd tính từ thư mục chứa file .htaccess. file1.php là tên file muốn bảo vệ. Bạn có thể lên trang web <http://www.tools.dynamicdrive.com/password/> để generate thông tin và lấy code về dán vào file htaccess thay vì tự code.

**Lời Kết**

Trong bài này ta chỉ tìm hiểu những ứng dụng bảo vệ website căn bản của file .htaccess. Bài tiếp theo ta sẽ tìm hiểu nâng cao hơn đó là **sử dụng file .htaccess để rewrite url**.

**Bài 03: Những ký hiệu hay dùng của htaccess**

Như ở bài giới thiệu điều kiện sử dụng thành thạo file .htaccess là bạn phải biết kiến thức[Regular Expression](http://freetuts.net/tag/regular-expression). Nhưng không phải bạn nào cũng biết nên bài này mình sẽ đưa ra một số ký hiệu trong regular expression và ký hiệu của file .htaccess.

**Các Ký Hiệu regular Expression Căn Bản**

Vấn đề Regular Expression mình đã trình bày ở serie Regular Expression trong php rồi nên mình chỉ đưa ra những quy tắc đơn giản nhất và hay dùng nhất.

**Ký hiệu [list\_char]**

Danh sách các ký tự được phép chấp nhận

Ví dụ: [abc] => chỉ chấp nhận ký tự a hoặc b hoặc c

**Ký hiệu [begin- end]**

Dành sách các ký tự bắt đầu từ ký tự BEGIN và kết thúc là ký tự END

Ví dụ:

* [a-z] => chấp nhận các ký tự thường từ a đến z
* [A-Z] => chấp nhận các ký tự hoa từ A đến Z
* [0-9] => chấp nhận các ký tự từ 0 đến 9
* [a-c] => chấp nhận các ký tự từ a đến c
* [a-zA-Z0-9] chấp nhận các chữ cái thường, hoa và các ký tự số

**ký hiệu {min,max}**

Xác định chiều dài cho chuỗi tối thiếu mà MIN và tối đa là Max.

Ví dụ:

* [a-z]{5,10} => các ký tự từ a tới z dài từ 5 đến 10 ký tự
* [A-Z]{1,100} => các ký tự từ A tới Z dài từ 1 đến 100 ký tự
* [0-9]{10} => ký tự số dài 1 ký tự
* [a-zA-Z]{1,} => các ký tự chữ hoa hoặc chữ thường dài từ 1 ký tự trở lên

**Ký hiệu ^ và $**

^ là ký hiệu bắt đầu chuỗi, $ là ký hiệu kết thúc chuỗi.

Ví dụ:

* ^[a->z]{10}$ => chuỗi tự bắt  đầu là chữ cái thường, kết thúc cũng là chữ cái thường  và dài 10 ký tự.
* ^freetuts$ => chuỗi cố định freetuts
* ^freetuts[a-z]{2} chuỗi bắt đầu bằng freetuts và kết thúc là 2 ký tự thường, ví dụ freetutste, freetutssm ...

**Ký hiệu .**

Ký hiệu dấu chấm (.) đại diện cho ký tự bất kỳ.

Ví dụ:

* .{10,20} => ký tự bất kỳ dài từ 10 đến 20 ký tự

**ký hiệu (+) (?) (\*)**

* + : xác định chiều dài 1 hoặc nhiều ký tự
* \* : xác định chiều dài từ 0 hoặc nhiều ký tự
* ? : xác định chiều dàu 0 hoặc 1 ký tự

Ví dụ:

* [a-z]+ => chuỗi các ký tự chữ cái thường từ 1 ký tự trở lên
* [a-z]\* => chuỗi các ký tự chữ cái thường từ 0 ký tự trở lên
* [a-z]? => chuỗi các ký tự chữ cái thường từ 0 đến 1 ký tự

**Ký hiệu ()**

Dùng gom nhóm các group Regular Exprsesion, ví dụ:

* ([a-z])([A-Z]): Góm thành 2 nhóm
* ([a-z][A-Z]): Gom thanhf 1 nhoms

**Ký hiệu [^]**

Ngược lại với ký hiệu [], nghĩa là so khớp với những ký tự không nằm trong danh sách cặp [], ví dụ:

* [^a-z]{5,10} => chuỗi dài 5 đến 10 ký tự và không có ký tự chữ cái thường từ a đến z
* [^abc]{} => so khớp với chuỗi không chứa chữ cái a, b và c

**Ký hiệu |**

Đây là toán tử OR trong Regular Expression, có ý nghĩa Regex A hoặc Regex B

Ví dụ:

* (a|b) => là ký tự a hoặc b
* (freetuts|phonline) => chuỗi freetuts hoặc phponline

**Ký hiệu \**

Dùng để đặt trước các ký hiệu đặc biệt trong chuỗi RegEx. Các ký tự đặc biệt là các ký tự nằm trong danh sách ký hiệu của Regular Expression.

Ví dụ:

* \. => ký tự dấu chấm chứ không phải ký hiệu dấu chấm trong RegEx
* \[ => ký tự mở ngoặc vuông chứ không phải ký hiệu mở ngoặc vuông trong RegEx

**Capturing GROUP**

Định nghĩa và các ví dụ về capturing group đã được trình bày trong serie Regular Expression trong php ở bài này, bạn vui lòng quay lại và đọc nhé.

Hiểu đơn giản Capturing Group là thứ tự các group gom nhóm trong chuỗi RegEx, toàn chuỗi là group 0, sau đó đếm theo tứ tự từ trái qua phải để tìm đến các group tiếp theo.

Ví dụ:

* ([a-z]{1,2})([0-9]{10,20}) => có 3 capturing group:
  + Thứ nhất là toàn chuỗi regex =>([a-z]{1,2})([0-9]{10,20})
  + Thứ 2 là theo thứ tự từ trái qua phải ta gặp group đầu tiên ([a-z]{1,2})
  + Thứ 3 là group tiếp theo ([0-9]{10,20})

Ta dùng những Capturing Group này để truyền vào URL khi viết lại đường dẫn (Rewrite Url).

Ngoài ra bạn cũng nên đọc bài None Capturing Group này đế tìm hiểu sâu hơn.

**Các ký hiệu riêng của file htaccess**

1. [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.
2. [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.
3. [N] - Next: chỉ thị cho server tiếp tục rewrite cho đến rule kế tiếp.
4. [G] - Gone: Chỉ định server trả về client trang báo lỗi không tồn tại (no longer exit)
5. [P] - Proxy: chỉ định server điểu kiển các yêu cầu được ấn định bởi mod\_proxy
6. [C] - Chain: Chỉ định server thực hiện rule hiện hành song song với rule trước đó.
7. [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.
8. [NC] - No-case: Chỉ định server match không phân biệt hoa thường.
9. [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.
10. [OR] - Toán tử kiểm tra từ trên xuống dưới, nếu điều kiện nào đúng thì dừng.
11. [NE] - No Escape: Chỉ định server xử lí các gói tin trả về mà không dùng kí tự thoát
12. [NS] - No Subrequest: Chỉ định server bỏ qua thư mục hiện hành nếu request nhắm vào thư mục con.
13. [QSA] - Append Query String: chỉ định server gắn chuỗi truy vấn vào cuối cùng của URL
14. [S=x] Skip: Chỉ định server bỏ qua không xem xét  x rules tiếp theo nữa nếu một rule đã được thực thi.
15. [T=MIME-type] - Khai báo định dạng files của server.
16. -d Kiểm tra thư mục có tồn tại hay không?
17. -f Kiểm tra file có tồn tại hay không
18. -s Kiểm tra giá trị của file có khác 0 hay không?

Bài này mang tính chất **giới thiệu các ký hiệu trong file htaccess** nên nếu các bạn không hiểu thì ở bài sau ta sẽ thực hành nó một cách trực quan hơn.

**Mã thông báo trả về từ server**

* 401 - Authorization Required : Lỗi chưa được xác thực user và 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.

**Lời kết**

Bài này chỉ mang tính chất **giới thiệu các ký hiệu trong file .htaccess** và **ký hiệu trong Regular Expression** chứ chưa thực hành nên các bạn cứ đọc nếu không hiểu thì bài tiếp theo chúng ta sẽ ứng dụng nó thực tế hơn. Nhưng ít nhất các bạn cũng phải đọc kỹ nó nhé

Tham khảo:

<http://freetuts.net/hoc-php/file-htaccess>

<http://phpjp.com/htaccess/>

<https://wiki.matbao.net/htaccess-can-ban.ashx>