

# Forum:[Hack Forums](#)

Thread: [\[The Empire\] Ultimate Beginners Guide To Htaccess](#)

• ???c??? • - March 26, 2014, 7:51 am

[img]http://imageshack.us/a/img90/6030/bannermo.gif[/img]

## Table of contents

- ? Intro
- ? Error documents
- ? Redirects
- ? Prevent viewing of htaccess
- ? Prevent directory listing
- ? Prevent hotlinking
- ? Password protection
- ? Blocking via IP

## Intro

Most of you here, if not all, will have heard of .htaccess at one point whilst working on your website. I can almost guarantee that you will have used this or read about using it specifically for custom error pages and password protection. Well, in this guide, I am going to cover both of those topics as well as a few others, to make sure you are fully utilizing your .htaccess file to the best of your ability. I will NOT be covering htaccess rewriting as that can get pretty complex and this is a beginners guide, but if you wish to know more about that then click [here](#).

First of all, what is a .htaccess file? Well to put it simply, here is a description of it.

Quote:A .htaccess (hypertext access) file is a directory-level plain text configuration file for web servers, which, in simple terms, controls access to a certain directory in your server.

.htaccess rules apply to a directory and all its subdirectories, unless there are more .htaccess files present within the sub-directories. The permissions of the .htaccess should be such that it allows universal read access but user only write access.

But wait, why do you always include a dot before the name of it? The answer is simple, .htaccess is the file extension. It is NOT file1.htaccess or thisfile.htaccess. It is always named simply, .htaccess.

To make one of these, just open up a new text file and when you are finished adding what you need, save it as .htaccess, it may change to htaccess.txt but once you upload it to your FTP you can change it to the proper name.

## Error Documents

Okay now there are a LOT of error codes that you may need to familiarize yourself with, but I will go over the main ones that you may encounter.

400 - Bad request  
403 - Forbidden  
404 - Not found  
500 - Internal server error

To use an error document, you must write it in the following format:

ErrorDocument code /directory/filename.ext

So that means if you were including a 404 page and you want to take it from your folder called 'errors', then you would write it in this format:

ErrorDocument 404 /errors/404.html

You can technically name the directories and page names anything you like, but I prefer to keep it simple, so as not to forget anything important or forget the locations of my files.

If you were going to include an Error Document for each of the errors I supplied, then it should look something like the following:

Quote:ErrorDocument 400 /errors/400.html  
ErrorDocument 401 /errors/401.html  
ErrorDocument 403 /errors/403.html  
ErrorDocument 404 /errors/404.html  
ErrorDocument 500 /errors/500.html

## Redirects

You can redirect using many methods, including Javascript, http-equiv or any of the server-side languages but .htaccess has to be the easiest due to the minimal work it actually requires. .htaccess will use redirect to look for any requests to a specific page that you want to direct somewhere else and once it finds it, it will then redirect to a new location which you specify.

Quote:Redirect /olddirectory/oldfile.html <http://site.com/newdirectory/newfile.html>

You may want to know that there are 3 elements to that command which are:

The Redirect command  
The location of the file/directory you want redirected  
URL of the destination you want to redirect to.

You can also redirect an entire directory by simple using

Quote:Redirect /olddirectory <http://yoursite.com/newdirectory/>

## Prevent Viewing Of .htaccess

If you use .htaccess for password protection (we will cover that later) then the location of your password info is in plain view through the .htaccess file itself. If you have set incorrect permissions or if your server is not as secure as it could be, then there is a possibility your site could become compromised as a browser thus has potential to view your file through a standard web interface.

To block direct access to the .htaccess file, simply add the following:

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

The first line specifies that the file named .htaccess is having this rule applied to it.

## Prevent Directory Listing

If you have a directory that you do not want users to be able to browse through, then you will want to prevent directory listing. In most cases you will not even need to do this, depending on your server, but in the case that you DO need to do this, then simply add this to your file.

```
Quote:IndexIgnore
```

The `*` used here is known as a wildcard, that by using will now block access to any file in that directory.

If you want to block everything apart from certain html pages, then use this:

```
Quote:IndexIgnore *.gif *.jpg
```

## Prevent Hotlinking

If you want to disallow hotlinking of your images etc (if you don't know the downfalls of hotlinking, then just google it as I am not going to cover it) then use this code:

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

This code creates a failed request when someone tries to hotlink your site, and shows a broken image in the case of image files.

If you want to show something different other than a broken image, then you can use this instead:

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

This will replace the before mentioned broken image with an image of your choice.

## Password Protection

The method we are going to use is known as .htaccess authentication or more commonly .htaccess password protection. It works by uploading two files to the directory you want to password protect, these files are called .htaccess and .htpasswd.

Your htaccess files should contain the following:

```
Quote:AuthType Basic
AuthName "Password Protected Area"
AuthUserFile /path/to/.htpasswd
Require valid-user
```

You only need to change “/path/to/.htpasswd” with the full path to your .htpasswd. Next you need to upload the .htpasswd file which contains the username and password to enter the password protected folder. The .htpasswd file should contain:

```
Quote:test:dGRkPurkuWmW2
```

The above code will allow the user “test” to access the password protected area with the password “test”. The text “dGRkPurkuWmW2” is a encrypted version of the password. You will need to use a htpasswd generator to create another password. Each line in the .htpasswd file contains a username and password combination, so feel free to add as many combinations as you like.

## Blocking Via IP

If you want to block a certain person via their IP, for any reason then you should use this code:

```
Quote:order allow,deny
deny from 123.45.6.7
deny from 012.34.5.
allow from all
```

You can obviously change the IPs to suit your needs, and remember to only put one command per line.

Nice man! Thanks for a bunch of clarification!

• **????c????** • - March 27, 2014, 9:00 am

(Mar 26, 2014 08:27 PM)OneAppleBoy Wrote: [\[CITE\]](#) Nice man! Thanks for a bunch of clarification! Not a problem man, glad I helped clear some stuff up for you.

**King of Hearts** - March 27, 2014, 9:03 am

This looks great. In the future, you should add modrewrite to change things like item.php?itemTitle=So This is a test item to /item/So This is a test item

• **????c????** • - March 27, 2014, 9:06 am

(Mar 27, 2014 09:03 AM)King of Hearts Wrote: [\[CITE\]](#) This looks great. In the future, you should add modrewrite to change things like item.php?itemTitle=So This is a test item to /item/So This is a test item

I am planning on doing a full rewrite tutorial soon, as I said in the intro, that can get quite tricky if you do not know what you are doing, and as this is a beginner tutorial I didn't want to delve too deep into that.

• **????c????** • - March 30, 2014, 6:17 am

I will be writing a tutorial on rewriting hopefully in the next 48 hours, so keep an eye out.