.htaccess Tutorial

Author:
Webhostingpad.com

Created On: 30 Nov 2007 10:37 PM

.htaccess Tutorial
Part 1 - Introduction
Introduction

In this tutorial you will find out about the .htaccess file and the power it has to improve your website. Although .htaccess is only a file, it can change settings on the servers and allow you to do many different things, the most popular being the ability to have your own custom 404 error pages. .htaccess isn't difficult to use and is made up of just a few simple instructions in a text file.

Does Webhostingpad Support It?

Yes, Webhostingpad supports .htaccess.

What Can I Do?

You may be wondering what .htaccess can do, or you may have read about some of its uses but don't realize how many things you can actually do with it.

There is a huge range of things .htaccess can do, including: password protecting folders, redirecting users automatically, custom error pages, changing your file extensions, banning users with certain IP addresses, only allowing users with certain IP addresses, stopping directory listings and using a different file as the index file.

Creating a .htaccess File

Creating a .htaccess file may cause you a few problems. Writing the file is easy - you just need enter the appropriate code into a text editor (like notepad). You may run into problems with saving the file. Because .htaccess is a strange file name (the file actually has no name but an 8-letter file extension), it may not be accepted on certain systems (e.g. Windows 3.1). With most operating systems, though, all you need to do is to save the file by entering the name as:

".htaccess"

including the quotes. If this doesn't work, you will need to name it something else (e.g. htaccess.txt) and then upload it to the server. Once you have uploaded the file, you can then rename it using an FTP program.

Warning

Before using .htaccess, I should give you one warning. Although using .htaccess on your server is extremely unlikely to cause any problems (if something is wrong it simply won't work), you should be wary if you are using the Microsoft FrontPage Extensions. The FrontPage extensions use the .htaccess file, so you should not really edit it to add your own information. If you do want to (this is not recommended, but possible) you should download the .htaccess file from your server first (if it exists) and then add your code to the beginning.

Custom Error Pages

The first use of the .htaccess file I will cover is custom error pages. These will allow you to have your own, personal error pages (for example, when a file is not found) instead of using your host's error pages or having no page. This will make your site seem much more professional in the unlikely event of an error. It will also allow you to create scripts to notify you if there is an error (for example, I use a PHP script on Free Webmaster Help to automatically e-mail me when a page is not found).

You can use custom error pages for any error as long as you know its number (like 404 for page not found) by adding the following to your .htaccess file:

ErrorDocument errornumber /file.html

For example, if I had the file notfound.html in the root directory of my site and I wanted to use it for a 404 error I would use:

ErrorDocument 404 /notfound.html

If the file is not in the root directory of your site, you just need to put the path to it:

ErrorDocument 500 /errorpages/500.html

These are some of the most common errors:

401 - Authorization Required
400 - Bad request
403 - Forbidden
500 - Internal Server Error
404 - Wrong page

Then, all you need to do is to create a file to display when the error happens, and upload it and the .htaccess file.

Part 2 - .htaccess Commands
Introduction

In the last part I introduced you to .htaccess and some of its useful features. Now I will show you how to use the .htaccess file to implement some of these.

Stop a Directory Index From Being Shown

Sometimes, for one reason or another, you will have no index file in your directory. This will, of course, mean that if someone types the directory name into their browser, a full listing of all the files in that directory will be shown. This could be a security risk for your site.

To prevent against this (without creating lots of new 'index' files, you can enter a command into your .htaccess file to stop the directory list from being shown:

Options -Indexes

Deny/Allow Certian IP Addresses

In some situations, you may want to only allow people with specific IP addresses to access your site (for example, only allowing people using a particular ISP to get into a certian directory) or you may want to ban certa IP addresses (for example, keeping disruptive members out of your message boards). Of course, this will only work if you know the IP addresses you want to ban and, as most people on the internet now have a dynamic IP address, this is not always the best way to limit usage.

You can block an IP address by using:

deny from 000.000.000.000

where 000.000.000.000 is the IP address. If you only specify 1 or 2 of the groups of numbers, you will block a whole range.

You can allow an IP address by using:

allow from 000.000.000.000

where 000.000.000.000 is the IP address. If you only specify 1 or 2 of the groups of numbers, you will allow a whole range.

If you want to deny everyone from accessing a directory, you can use:

deny from all

but this will still allow scripts to use the files in the directory.

Alternative Index Files

You may not always want to use index.htm or index.html as your index file for a directory. For example, if you are using PHP files in your site, you may want index.php to be the index file for a directory. You are not limited to 'index' files, though. Using .htaccess you can set foofoo.blah to be your index file if you want to!

Alternate index files are entered in a list. The server will work from left to right, checking to see if each file exists, and if none of them exist it will display a directory listing (unless, of course, you have turned this off).

DirectoryIndex index.php index.php3 messagebrd.pl index.html index.htm

Redirection

One of the most useful functions of the .htaccess file is to redirect requests to different files, either on the same server, or on a completely different web site. It can be extremely useful if you change the name of one of your files but allow users to still find it. Another use (which I find very useful) is to redirect to a longer URL. For example, in my newsletters, I can use a very short URL for my affiliate links. The following can be done to redirect a specific file:

Redirect /location/from/root/file.ext http://www.othersite.com/new/file/location.xyz

In this above example, a file in the root directory called oldfile.html would be entered as:

/oldfile.html

and a file in the old subdirectory would be entered as:

/old/oldfile.html

You can also redirect whole directoires of your site using the .htaccess file, for example if you had a directory called olddirectory on your site and you had set up the same files on a new site at: http://www.newsite.com/newdirectory/, you could redirect all the files in that directory without having to specify each one:

Redirect /olddirectory http://www.newsite.com/newdirectory

Then, any request to your site below /olddirectory will bee redirected to the new site, with the extra information in the URL added on. For example, if someone typed in:

http://www.youroldsite.com/olddirecotry/oldfiles/images/image.gif

They would be redirected to:

http://www.newsite.com/newdirectory/oldfiles/images/image.gif

This can prove to be extremely powerful if used correctly.

Part 3 - Password Protection
Introduction

Although there are many uses of the .htaccess file, by far the most popular, and probably most

useful, is being able to reliably password-protect directories on websites. Although JavaScript, etc. can also be used to do this, only .htaccess has total security (as someone must know the password to get into the directory, there are no 'back doors')

The .htaccess File

Adding password protection to a directory using .htaccess takes two stages. The first part is to add the appropriate lines to your .htaccess file in the directory you would like to protect. Everything below this directory will be password protected:

```
AuthName "Section Name"
AuthType Basic
AuthUserFile /full/path/to/.htpasswd
Require valid-user
```

There are a few parts of this which you will need to change for your site. You should replace "Section Name" with the name of the part of the site you are protecting, e.g. "Members Area".

The /full/parth/to/.htpasswd should be changed to reflect the full server path to the .htpasswd file (more on this later). If you do not know what the full path to your web space is, contact your system administrator for details.

The .htpasswd File

Password protecting a directory takes a little more work than any of the other .htaccess functions because you must also create a file to contain the usernames and passwords that are allowed to access the site. These should be placed in a file which (by default) should be called .htpasswd. Like the .htaccess file, this is a file with no name and an 8 letter extension. This can be placed anywhere within your website (as the passwords are encrypted), but it is advisable to store it outside the web root so that it is impossible to access it from the web.

Entering Usernames And Passwords

Once you have created your .htpasswd file (you can do this in a standard text editor) you must enter the usernames and passwords to access the site. They should be entered as follows:

```
username:password
```

where the password is the encrypted format of the password. To encrypt the password you will either need to use one of the pre-made scripts available on the web or write your own. There is a good username/password service at the KxS site (http://www.kxs.net/support/htaccess_pw.html) which will allow you to enter the user name and password and will output it in the correct format.

For multiple users, just add extra lines to your .htpasswd file in the same format as the first. There are even scripts available for free which will manage the .htpasswd file and will allow automatic adding/removing of users etc.

Accessing The Site

When you try to access a site which has been protected by .htaccess, your browser will pop up a standard username/password dialog box. If you don't like this, there are certain scripts available which allow you to embed a username/password box in a website to do the authentication. You can also send the username and password (unencrypted) in the URL as follows:

http://username:password@www.website.com/directory/

Summary

.htaccess is one of the most useful files a webmaster can use. There are a wide variety of different uses for it which can save time and increase security on your website.