Related Articles

Apache

- How To Install and Secure phpMyAdmin on Ubuntu 12.04
- How To Install Linux, Apache, MySQL, PHP (LAMP) stack On CentOS 6
- How To Install Linux, Apache, MySQL, PHP (LAMP) stack On CentOS 6
- How To Set Up Apache Virtual Hosts on Ubuntu 12.04 LTS
- How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu

How To Use the .htaccess File

Published On: 12 Jul 18:10

Write an Article

Why Use an .htaccess Page

An .htaccess file is a way to configure the details of your website without needed to alter the server config files. The period that starts the file name will keep the file hidden within the folder.

You can create the .htaccess file in a text editor (make sure to name it only .htaccess without any other extension or name) and then upload it to your site through an ftp client.

Additionally the placement of the .htaccess file is important. The configurations in that file will affect everything in its directory and the directories under it.

Things to be Aware of

Although an .htaccess page can be immensely useful and can be used to make marked improvement to a site, there are 2 things that it can influence.

One: Speed—the .htaccess page *may* slow down your server somewhat; for most servers this will probably be an imperceptible change. This is because of the location of the page: the .htaccess file affects the pages in its directory and all of the directories under it. Each time a page loads, the server scans its directory, and any above it until it reaches the highest directory or an .htaccess file. This process will occur as long as the AllowOverride allows the use of .htaccess files, whether or not the file the .htaccess files actually exists.

Two: Security—the .htaccess file is much more accessible than standard apache configuration and the changes are made live instantly (without the need to restart the server). Granting users permission to make alterations in the .htaccess file gives them a lot of control over the server itself. Any directive placed in the .htaccess file, has the same effect as it would in the apache configuration itself.

Generally speaking, Apache discourages the use of the .htaccess if the user can easily reach the apache configuration files themselves.

With that out of the way, let's proceed with the .htaccess info.

How to Activate an .htaccess file

If you have access to the server settings you can edit the configuration to allow the .htaccess file to override standard website configs. Open the apache2 default host configuration file. NB: You will need sudo privileges for this step.

```
sudo nano /etc/apache2/sites-available/default
```

Once inside that file, find the following section, and change the line that says AllowOverride from None to All. The section should now look like this:

```
 <Directory /var/www/>
```

```
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
 </Directory>
```

After you save and exit that file, restart apache.

```
sudo service apache2 restart
```

Creating the .htaccess File:

You can create the .htaccess file in a text editor (make sure to name it only .htaccess without any other extension or name) and then upload it to your site through an ftp client.

Alternatively you can use this command, replacing the example.com with the name of your site, to create your .htaccess file in terminal.

```
sudo nano /var/www/example.com/.htaccess
```

Five Common Uses for an .htaccess Page

---

1. Mod_Rewrite: one of the most useful facets of the .htaccess file is mod_rewrite. You can use the space in the .htaccess file to designate and alter how URLs and web pages on your sites are displayed to your users. You can find the entire tutorial on how to do this here.

---

2. Authentication: Although using the .htaccess file does not require as many permissions as accessing the apache2.conf file would require, we can still make effective changes to a site. Once such change is to require a password to access certain sections of the webpage.

The .htaccess passwords are kept in a file called .htpasswd. Go ahead and create and save that file, being sure to store it somewhere other than the web directory, for security reasons.

You should use the space inside the .htpasswd file to write in the name and passwords of all the users that you want to have access to the protected part of the site.

You can use this useful site to generate the username and encrypted password pair.

If the username of your authorized user is jsmith and password is “awesome”, the pair would look like this: jsmith:VtweQU73iyETM. You can paste as many lines as needed into the .htpasswd file, but be sure that every user gets their own line.

Once you are finished with the .htpasswd file, you can type this code into the .htaccess file to begin using the password function:

```
AuthUserFile /usr/local/username/safedirectory/.htpasswd
AuthGroupFile /dev/null
AuthName Please Enter Password
AuthType Basic
Require valid-user
```

- **AuthUserFile:** This line designates the server path to the .htpasswd file.
- **AuthGroupFile:** This line can be used to convey the location of the .htgroup. As we have not created such a file, we can leave /dev/null in place.
- **AuthName:** This is text that will be displayed at the password prompt. You can put anything here.
- **AuthType:** This refers to the type of authentication that will be used to the check the passwords. The passwords are checked via HTTP and the keyword Basic should not be changed.
- **Require valid-user:** This line represents one of two possibilities. "Require valid-user" tells the .htaccess file that there are several people who should be able to log into the password protected area. The other option is to use the phrase "require user *username*" to indicate the specific permitted person.

---

3. Custom Error Pages: the .htaccess file additionally allows you to create custom error pages for your site. Some of the most common errors are:

- 400 Bad Request
- 401 Authorization Required
- 403 Forbidden Page
- 404 File not Found
- 500 Internal Error

To make a page look friendlier and to provide more information to the site visitor than the default server error page offers, you can use the .htaccess file to create custom error pages.

I'm going to create a 404 page in this tutorial. However, you can substitute that error for whatever you prefer:

Once you have created and uploaded desired error page, you can go ahead and designate its location in the .htaccess file.

```
ErrorDocument 404 /new404.html
```

Keep in mind that the Apache looks for the 404 page located within the site's root. If you placed the new error page in a deeper subdirectory, you need to include that in the line, making it look something like this:

```
ErrorDocument 404 /error_pages/new404.html
```

4. Mime Types: In cases where your site features some application files that your server was not set up to deliver, you can add MIME types to your Apache server in the .htaccess file with the following code.

```
AddType audio/mp4a-latm .m4a
```

Be sure to replace application and file extension with the Mime Type that you want to support.

---

5. SSI: Server Side Includes are a great time-saver on a website. One of the most common uses of SSI is to update a large number of pages with some specific data, without having to update each page individually (for example, if you want to change a quotation at the bottom of a page).

To enable SSI, type the following code into your .htaccess file.

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

These three lines have the effect of telling the .htaccess that .shtml files are valid, with the second line specifically making the server parse all files ending in .shtml for any SSI commands.

However, if you have many .html pages that you are not eager to rename with .shtml extensions, you can use another tactic to parse them for SSI commands, the XBitHack.

Adding this line to the .htaccess file makes Apache check all the html files with the appropriate permissions for Server Side Includes.

```
XBitHack on
```

To make a page eligible for the XBitHack, use this command:

```
chmod +x pagename.html
```

See More

The .htaccess page gives you a lot of flexibility to build up your site, and this was just a brief overview.

If you have any further questions about the specific capabilities of the .htaccess file, feel free to post your questions in our Q&A Forum and we'll be happy to answer them.

By Etel Sverdlov

Comments

ja.delcallar

I followed the tutorial but when i upload via filezilla (Alreay installed proftpd), it says 550 .htaccess: Permission denied

Posted January 14th, 2013 09:49

Moisey

Are you upload to other files to that directory as that user or is only .htaccess prevented from being uploaded?

Posted January 14th, 2013 14:29

ja.delcallar

I am uploading to var/www/ directory via filezilla, and filezilla wont allow me to upload .htaccess

Posted January 15th, 2013 01:15

Moisey

Can you upload any other files that aren't .htaccess, such as "test-file"?

Posted January 15th, 2013 01:39

benny.subarja

403 forbidden site after doing this ,

i have to change AllowOverride All

to

AllowOverride None

and its working again

Posted March 7th, 2013 08:27

admin

Hi everybody,
I have same and fixed this problems.
1. Install Mod_rewite
sudo a2enmod rewrite
2. Change AllowOverride from None to All
sudo nano /etc/apache2/sites-available/default
.....

Options Indexes FollowSymLinks MultiViews
AllowOverride All
Order allow,deny
allow from all

.....
3. Restart apache
sudo service apache2 restart
4. Create .htaccess file by command
sudo nano .htaccess
pate your htaccess code -> Save and Exit
My site working. I hope this is helpful :)

Posted August 10th, 2013 06:38

artisannm

admin, what do you mean by, "Paste your htaccess code"? paste what, where?
thanks.

Posted September 13th, 2013 14:43

Kamal Nasser

@artisannm: Your htaccess code is the code you came up with after following this article.

Posted September 13th, 2013 15:43

artisannm

Hmm, nothing is helping. I am trying to install a community forum program called Elgg. its located in the top directory. I'm using filezilla for my ftp. The installer has a precheck:
"PHP
Your server's PHP satisfies all of Elgg's requirements."

"Web server
We think your server is running the Apache web server.

The rewrite test failed and the most likely cause is that AllowOverride is not set to All for Elgg's directory. This prevents Apache from processing the .htaccess file which contains the rewrite rules.

A less likely cause is Apache is configured with an alias for your Elgg directory and you need to set the RewriteBase in your .htaccess. There are further instructions in the .htaccess file in your Elgg directory."

I can't edit the .htaccess permissions inside the Elgg directory. I have changed Allow Override to All. I followed Admin's steps to no avail.
I didn't follow the steps for creating custom urls etc, as that wasn't applicable.

I did check inside the Elgg folder and copied the .htaccess-dist file over to the .htaccess and still can't change

permissions on .htaccess but can change permissions on .htaccess-dist

I also tried uncommenting Rewrite Base in .htaccess
I have restarted apache2 as guided.
I can't get past this .htaccess file in order to continue setting up the Elgg install.
I'm on ubuntu 12.04, interfacing through Terminal on an imac. I have mysql up and running along with apache2. I have a phpbb3 forum up and running as well, from the top directory along with a website.

thanks for your help in advance...

Posted September 14th, 2013 00:47

Kamal Nasser

@artisannm: Please pastebin apache's config files (and .htaccess).

Try enabling the rewrite module:

```
sudo a2enmod rewrite

sudo service apache2 restart
```

Posted September 15th, 2013 18:15

kenny

I had the same problem. Thanks to post my admin I got it working. His comment should be included in the main article.

"1. Install Mod_rewite
sudo a2enmod rewrite"

Posted October 8th, 2013 03:11

luis

Admin Thank you!

Posted October 24th, 2013 03:58

matteo.poile

Hi,
I've got some problem with htaccess in a virtualhost on Ubuntu 12.04.
First of all in my "/etc/apache2/sites-available/" folder there is 000-default.conf and not default.conf but this was not the problem.
Sudo a2enmod rewrite return Module rewrite already enabled and in the site.conf where I need to use htaccess, AllowOverride was set correctly to All but it didn't work because in my /ect/apache2/apache2.conf I had AllowOverride None.

Hope this help

Posted October 27th, 2013 21:11

jntslvdrt

Well, as said on tutorial, all htaccess do apache config can do as well too.
is there any tutorial teaching this?

Posted October 30th, 2013 19:20

fenixkim

Thanks a lot!!!

Posted December 9th, 2013 15:36

mario

AuthName Please Enter Password
Should be:
AuthName "Please Enter Password"

As per the Apache documentation: http://httpd.apache.org/docs/current/mod/mod_authn_core.html#authname

---

Posted January 31st, 2014 19:08

Leave a comment

Email

Password

Create your account or sign-in

Post my Comment

Company

- Pricing
- Comparison Chart
- Features
- Customers
- About
- FAQ
- Press
- We're Hiring!
- API
- Integrations
- Network Status
- Contact

Community

- Articles & Tutorials
- Get Paid to Write
- Suggest an Article
- Community Chat
- Q&A
- Blog
- Referral Program
- Events Calendar
- Feedback
- Badges & Logos
- The Shop

Getting Started

- One-Click Install Applications
- What is Cloud Hosting?
- Control Panel Overview
- Deploy a Virtual Server
- Set-Up SSH Keys
- Install Git on Ubuntu
- How to Install Ruby on Rails
- How to Install LAMP Stack
- Set-Up a Host Name