

How to Set up Rules and Redirects in .htaccess?

No comments



Knowing how to handle redirects is important if you want your site to perform well despite all those changes. Adding and removing content, changing domains, reorganizing, and more are some things that happen on a website. To achieve this, you can create redirect rules in a .htaccess file at the server configuration level.

What Is a .htaccess File?

Even if you don't have access to the main configuration files for Apache servers, .htaccess files allow you to make configuration changes. .htaccess files operate on an individual directory basis. They are useful if you need to create a rule that only applies to one directory and its subdirectories. They also prevent you from having to edit the main configuration files of your server.

The .htaccess file uses the same syntax as the root configuration file, allowing you to use any directives normally found in the server configuration files. According to Apache's official documentation, it is recommended that you only use .htaccess files when necessary, as they can sometimes slow down your server's performance. Consider using other

redirect methods, such as JavaScript, PHP, or HTML, if it makes sense for your site. Furthermore, plugins make it easy to redirect pages, such as Redirection on WordPress.

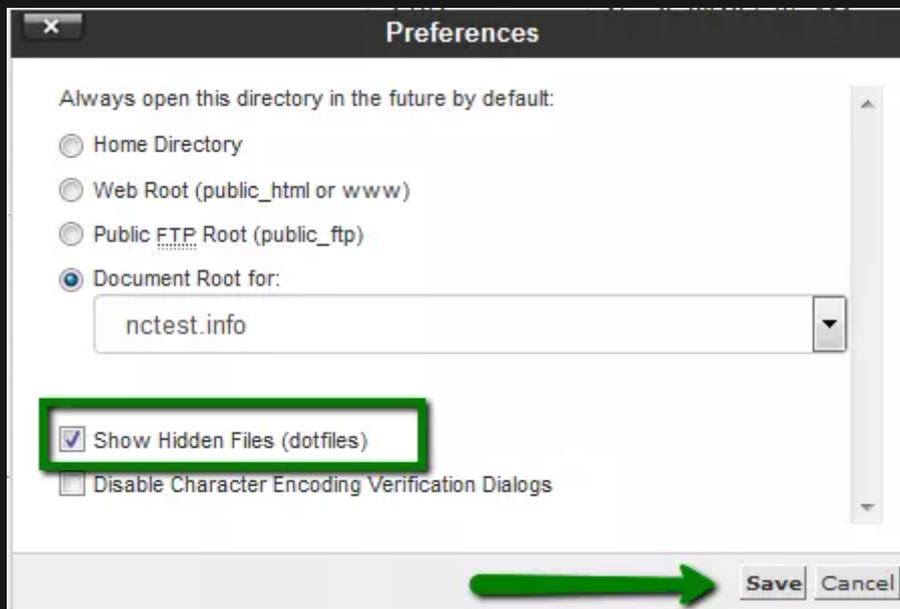
Set up Rules and Redirects in .htaccess

.htaccess is a configuration file that is supported by the Apache web server. It is used to alter the web server's configuration (enabling or disabling additional features) for a specific account without affecting the global configuration. If you place a .htaccess file in a directory containing files and subdirectories, the changes will be reflected instantly, and there is no need to restart the server.

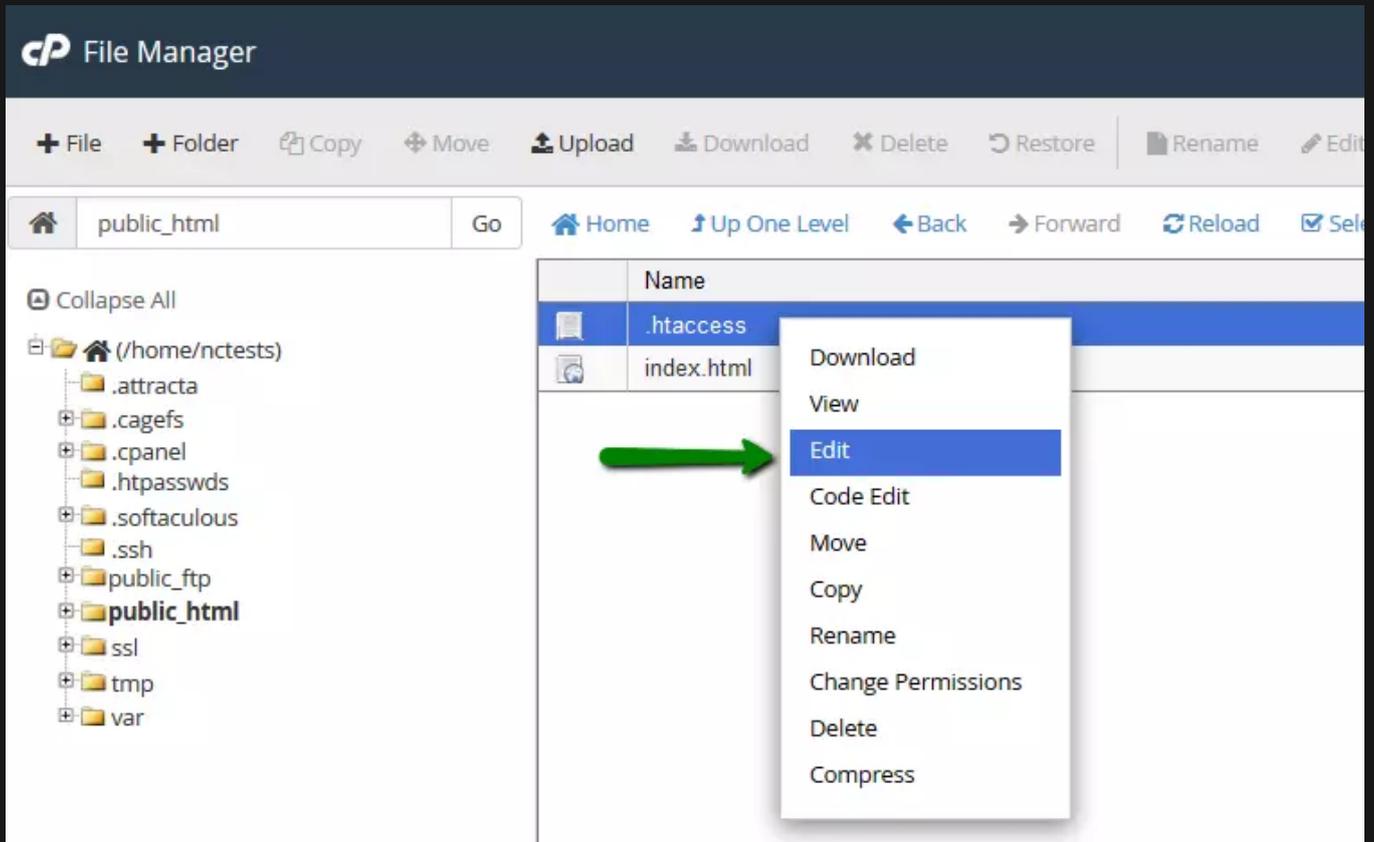
How to Locate .htaccess File

Follow these steps to access your hosting account's main .htaccess file:

1. Log into your cPanel account.
2. Select Files >> File Manager.
3. Navigate to the public_html folder if you wish to edit the .htaccess file for your main domain. To modify the .htaccess file for your add-on domain, move to the public_html/youraddondomain.com folder.
In the Settings menu, ensure that Show Hidden Files (dotfiles) is enabled:



4. Right-click the .htaccess file and select Edit:



5. If you do not find a .htaccess file in your File manager, create one using the File option.
Now you can add your own configuration rules and save them.

List of Commonly Used .htaccess Rules:

Below are the common rules for using a .htaccess file:

Authorization/authentication

Authentication/authorization: defines security restrictions for a directory. If a directory is password-protected, visitors must enter their username and password to access it. To set up such protection, you must:

1. Create the directory in which you want to protect.

/home/cpanel_user/.htpasswd/ folder (e.g., for public_html/test the path will be .htpasswd/public_html/test/).

2. Use the online generator to generate a hashed passwd file in this directory.

3. The following directives should be added to .htaccess:

```
AuthType Basic
AuthName "Directory Name"
AuthUserFile "/home/cpanel_user/.htpasswd/public_html/test/passwd" require valid-user
```

Blocking

The blocking feature allows users to be blocked based on their IP addresses or domains. It is very useful when blocking unwanted visitors or allowing the administrator to access certain website areas. The following text should be included in a .htaccess file to set up certain blocking rules:

1- Allowing everyone else access and blocking users with IP addresses X.X.X.X 2- Block all visitors except those with IP addresses X.X.X.X and Y.Y.Y.Y Mixing the deprecated directives Allow, Deny, and Order with the new directive Require is not recommended.

Custom Error Pages

The Custom Error Pages feature lets you create custom error pages for your website. This option is very useful since it allows you to display an error message that matches your website theme if a URL on your website fails. In this way, you can avoid the default '404 File Not Found' error and display a custom error that explains how to get to the content of your website rather than leaving visitors confused. The following text should be placed in a .htaccess file to set up a custom error document:

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

If a 404 (File Not Found) error occurs, this line instructs Apache to load an error page located in the domain's root directory. If any other error codes (403, 500, etc.) are encountered, substitute 404 with the corresponding error code and replace /404.html with the error file's path.

Mod_Rewrite (redirect rules)

Specifies how visitors will see web pages and URLs. Please take note of the use of Mod_Rewrite rules in the .htaccess file. Mod_Rewrite maps URLs to filesystem paths by default. However, it can also redirect URLs. To create a redirect, you should choose the type of redirect that is most appropriate for you:

Permanent Redirect

The status code of a permanent redirect is 301, cached in the browser memory, unlike a temporary redirect. It indicates the page has been relocated and asks all search engines and user agents to update their databases.

Temporary Redirect

A temporary redirect is when the page sends status code 302 to the browser. This code prevents the browser from caching the redirect into its saved data. The visitor or search engine will be redirected, but the search engine will continue to index the original page. The best kind of redirect is this unless you're absolutely certain you'll never change it again.

The following is a list of the most common and useful redirects that can be set through the .htaccess file (the domains in the examples should be replaced with your own ones):

Redirect from example.com to domain.com permanently

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^example\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.example\.com$
RewriteRule ^(.*)$ "http://\./domain\.com/$1" [R=301,L]
```

Redirect from example.com to domain.com temporarily

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^example\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.example\.com$
RewriteRule ^(.*)$ "http://\./domain\.com/" [R=302,L]
```

Examples of permanent redirects can be found below. Temporary redirects can be defined by replacing [R=301,L] with [R=302,L] at the end of the code (where necessary).

Redirect from example.com/subfolder to domain.com

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^example\.com$ [OR]
RewriteCond %{HTTP_HOST} ^www\.example\.com$
```

```
RewriteRule ^subfolder/$ "http://\./domain\.com/" [R=301,L]
```

Redirect from HTTP to HTTPS, for example.com

```
RewriteEngine On  
RewriteCond %{HTTPS} !=on  
RewriteRule .* https://example.com%{REQUEST_URI} [R,L]
```

or

```
RewriteCond %{SERVER_PORT} 80  
RewriteCond %{HTTP_HOST} ^example\.com$ [OR]  
RewriteCond %{HTTP_HOST} ^www\.example\.com$  
RewriteRule ^(.*)$ https://www.example.com/$1 [R,L]
```

Redirect from non-WWW to WWW

The .htaccess file takes effect for any domain on:

```
RewriteEngine On  
RewriteCond %{HTTP_HOST} !^www\  
RewriteRule ^(.*)$ http://www.%{HTTP_HOST}/$1 [R=301,L]
```

For a certain domain, example.com:

```
RewriteEngine On  
RewriteCond %{HTTP_HOST} ^example\.com$ [NC]  
RewriteRule ^(.*)$ http://www.example.com/$1 [R=301,L]
```

Redirect from WWW to non-WWW

For any domain .htaccess takes effect on:

```
RewriteEngine On  
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]  
RewriteRule ^(.*)$ http://%1/$1 [R=301,L]
```

For a certain domain, example.com:

```
RewriteEngine On  
RewriteCond %{HTTP_HOST} ^www\.example\.com [NC]
```

```
RewriteRule (.*) http://example.com/$1 [R=301,L]
```

Redirect all example.com pages to the corresponding domain.com pages

```
RedirectMatch 301 ^/(.*)$ http://domain.com/$1
```

The URLs for all pages must match both domains otherwise, a "Page not found" message will appear on the target site.

Redirect one page to a new URL

```
Redirect 301 /old_page.html http://www.domain.com/new_page.html
```

Using this method, you can redirect a deleted page to a 404 error or use it for SEO purposes after updating the content references.

Domain Root Directory Change

Sets the directory root for the main domain to public_html/subfolder

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^(www)?example.com$
RewriteCond %{REQUEST_URI} !^/subfolder/
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ /subfolder/$1
RewriteCond %{HTTP_HOST} ^(www)?example.com$
RewriteRule ^(/)?$ subfolder/index.php [L]
```

To configure certain rules for a domain, you should place the .htaccess file in the root directory.

Disabling the Existing .htaccess Rules

If you need to disable any existing rules, such as for testing purposes, you can comment them out by adding the pound sign # before each line. You can also disable a line or even a block of lines by selecting them and pressing Ctrl + /.

Do you need any assistance? Contact us in [services page](#).

Join Arashtad Community

Follow Arashtad on Social Media

We provide variety of content, products, services, tools, tutorials, etc. Each social profile according to its features and purpose can cover only one or few parts of our updates. We can not upload our videos on SoundCloud or provide our eBooks on Youtube. So, for not missing any high quality original content that we provide on various social networks, make sure you follow us on as many social networks as you're active in. You can find out Arashtad's profiles on different social media services.



Get Even Closer!

Did you know that only one universal Arashtad account makes you able to log into all Arashtad network at once? Creating an Arashtad account is free. Why not to try it? Also, we have regular updates on our newsletter and feed entries. Use all these beneficial free features to get more involved with the community and enjoy the many products, services, tools, tutorials, etc. that we provide frequently.

[SIGN UP](#)[NEWSLETTER](#)[RSS FEED](#)