

How to enable .htaccess in Apache HTTP server

Author : Dan Nanni

Categories : [Networking](#)

Tagged as : [apache2htaccess](#)

Question: I wanted to use .htaccess file to configure something for my website running on Apache web server. But the .htaccess file I placed in the website's document root directory is not working. How can I enable .htaccess in Apache web server?

An .htaccess file is a directory-level configuration file for Apache HTTP server, which allows one to override the web server's system-wide settings without modifying the global configuration file (e.g., httpd.conf or apache2.conf). Things like per-directory access control, password protection, URL redirection or hot link prevention can be configured in the .htaccess file.

An .htaccess file can be placed in the website's document root folder and/or any of its subfolders. Especially in a shared web hosting environment, where multiple websites are provisioned on a single web server, an .htaccess file is a convenient way for each website to override global web server settings without root privilege.

If you want to use the .htaccess file in Apache HTTP server, you need to explicitly enable it from the web server's main configuration file. Otherwise, your .htaccess file will be ignored.

Here is **how to enable .htaccess for Apache HTTP server**.

Enable .htaccess on CentOS or Fedora

Open a global Apache configuration file with a text editor.

```
$ sudo vi /etc/httpd/conf/httpd.conf
```

Look for directives. Each section enclosed by `<#>` and defines web server settings for a particular document root directory and all its subdirectories. Within the section, replace "**AllowOverride None**" with "**AllowOverride All**". This will activate .htaccess for the corresponding directory tree.

Ask Xmodulo

Find answers to commonly asked Linux questions

<http://ask.xmodulo.com>

```
<Directory "/var/www">
  AllowOverride None
  # Allow open access:
  Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/html">
  #
  # Possible values for the Options directive are "None", "All",
  # or any combination of:
  #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
  #
  # Note that "MultiViews" must be named *explicitly* --- "Options All"
  # doesn't give it to you.
  #
  # The Options directive is both complicated and important. Please see
  # http://httpd.apache.org/docs/2.4/mod/core.html#options
  # for more information.
  #
  Options Indexes FollowSymLinks
  # AllowOverride controls what directives may be placed in .htaccess files.
  # It can be "All", "None", or any combination of the keywords:
  #   Options FileInfo AuthConfig Limit
  AllowOverride All
```

.htaccess is disabled for /var/www

.htaccess is enabled for /var/www/html and its sub-directories

After modifying the configuration, restart Apache.

On CentOS 7 or later:

```
$ sudo systemctl restart httpd
```

On CentOS 6 or earlier:

```
$ sudo service restart httpd
```

Enable .htaccess on Debian or Ubuntu

Open a global Apache configuration (/etc/apache2/apache2.conf) or site-specific configuration (/etc/apache2/sites-available/mywebsite.conf) with a text editor.

Ask Xmodulo

Find answers to commonly asked Linux questions

<http://ask.xmodulo.com>

Look for directives. In the section surrounded by `<#>`, replace "**AllowOverride None**" with "**AllowOverride All**". This will enable `.htaccess` for the whole directory tree in "directory-path". For example, the following will enable `.htaccess` for `/var/www` and all its subdirectories.

```
Options Indexes FollowSymLinks AllowOverride All
Require all granted
```

After modifying the configuration, restart Apache:

```
$ sudo service apache2 restart
```

Note: Once `.htaccess` is enabled, modifying `.htaccess` content does not require restarting Apache. Any change made in `.htaccess` will take immediate effect. As a side effect, every HTTP request involves file system access for every `.htaccess` file in the `htaccess-enabled` directory tree. Thus enabling `.htaccess` comes with a performance hit in the web server.