URL redirect/rewrite using the .htaccess file

By default your website can be accessed with both www.example.com and example.com. Since Google penalizes this due to duplicated content reasons, you should restrict the access to either www.example.com or example.com. Some links may be outside of your website scope and/or the search engines may have already indexed your website under both addresses.

- [Using the Redirect tool in the cPanel](#)

REDIRECTING TO OR FROM WWW

How do I redirect all links for www.example.com to example.com?

Create a 301 redirect forcing all http requests to use either www.example.com or example.com:

- **Redirect example.com to www.example.com:**

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

- **Redirect www.example.com to example.com:**

  ```
  RewriteEngine on
  RewriteCond %{HTTP_HOST} ^www\.example\.com$
  RewriteRule ^/?$ "http\:\/\/example\.com\/" [R=301,L]
  ```

EXPLANATION

The first line tells apache to start the rewrite module.
The next line:

```
RewriteCond %{HTTP_HOST} !^www.example.com$ [NC]
```

specifies that the next rule only fires when the http host (that means the domain of the queried url) is not (specified with the !) www.example.com.

The `$` means that the host ends with www.example.com and the result is that all pages from www.example.com will trigger the following rewrite rule. Combined with the inversive `!` is the result every host that is not www.example.com will be redirected to this domain.

The `[NC]` specifies that the http host is case insensitive. The escapes the `.` because this is a special character (normally, the dot (.) means that one character is unspecified).

The final line describes the action that should be executed:

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

The `^(.*)$` is a little magic trick. Can you remember the meaning of the dot? If not, this can be any character(but only one).

So `.*` means that you can have a lot of characters, not only one.

This is what we need because `^(.*)$` contains the requested url, without the domain.

The next part `http://www.example.com/$1` describes the target of the rewrite rule. This is our "final" used domain name, where $1 contains the content of the `(.*)`.

The next part is also important, since it does the 301 redirect for us automatically: `[L,R=301]`. `L` means this is the last rule in this run. After this rewrite the webserver will return a result. The `R=301` means that the webserver returns a 301 moved permanently to the requesting browser or search engine.

HOW TO REDIRECT VISITORS TO HTTPS?

In case you set a valid certificate for a domain name, we recommend that visitors redirect an unsafe http:// address to secure https://.

You can easily redirect the redirection by modifying the .htaccess file in your document_root folder for that domain name.

Use the following rows in this example to redirect all users of a domain name from insecure (http://) URLs to secure (https://) URLs.

Replace the example.com domain name with your domain name.

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://example.com/$1 [R=301,L]
```

Revision #2
Created 24 October 2023 14:41:13 by Judit Pásztor
Updated 26 October 2023 10:23:57 by Judit Pásztor