# HTTP URL Redirection

HTML-5.com is an HTML User's Guide and quick reference of HTML elements and attributes for web developers who code HTML web pages, not only for HTML 5 but for HTML coding in general, with demos and examples of HTML code plus a cheat sheet for web developers. TV Series & Actors and Actresses. Follow TV Series and HTML 5 on Google+.

HTML-5.com ➢ itemscopehttp://data-vocabulary.org/Breadcrumb<span itemprop="title">HTML 5</span> ➢ itemscopehttp://data-vocabulary.org/Breadcrumb **HTTP URL Redirection**

## HTTP URL Redirect

A <dfn>RewriteRule</dfn> is one way to redirect a request for a URL to a different page, which is called a <dfn>URL redirect</dfn> or <dfn>URL forwarding</dfn>. It is the recommended way to do a URL redirect for static web pages.

If you move a site, directory or one or more web pages you should create a URL redirect in the old location to automatically send users to the new location. To potentially preserve the page rank of those pages, it is recommended to create URL redirects on a one-to-one page-for-page basis (see Google Webmaster Central Help on "Moving Your Site"). In addition, the instructions in Google Webmaster Tools "Site configuration" - "Change of address" should be followed.

Although this also works for dynamically-generated pages it requires knowledge on how to code regular expressions.

When generating HTML with a program or server-side scripting, the language probably has an API to send the proper HTTP headers.

- URL Redirect in ASP .NET
- URL Redirect in Java
- URL Redirect in JSP
- URL Redirect in Perl CGI
- URL Redirect in PHP
- URL Redirect in Ruby on Rails

### Regular Expression Metacharacters

Any regular expression metacharacters in the RewriteRule must be escaped with a backslash (\). These metacharacters that need to be escaped include:

- \ - backslash
- ^ - caret
- $ - dollar sign
- . - period (dot)
- | - vertical bar (pipe)
- ? - question mark

- * - asterisk (star)
- + - plus sign
- ( and ) - parentheses
- [ and ] - square brackets

See the URL Redirect Examples below for an example.

back to top

## URL Redirect Code

### Code for HTTP URL Redirect

#### URL redirect to a different extension

<samp><var>/path/</var>.htaccess</samp>

```
RewriteEngine On

RewriteRule ^(.+).<var mode="pre"><#ext1#></var>  $ /<var mode="pre"><#path/#></var>  $1.<var mode='
```

#### URL redirect to a different page

<samp><var>/old-path/</var>.htaccess</samp>

```
RewriteEngine On

RewriteRule ^<var mode="pre"><#old-page.html#></var>  $ /<var mode="pre"><#new-path/new-page.html#><
```

#### URL redirect to a different directory

<samp><var>/old-path/</var>.htaccess</samp>

```
RewriteEngine On

RewriteRule ^(.*)$ /<var mode="pre"><#new-path/#></var>  $1 [R=permanent,L]
```

Although this is the simplest way to redirect all requests to a different directory, it blindly does the redirect to the new page. To verify that the target page exists, include a `RewriteCond` with `-f`:

<samp><var>/old-path/</var>.htaccess</samp>

```
RewriteEngine On

RewriteCond %{REQUEST_URI} ^/<var mode="pre"><#old-path/#></var>  (.*)$ [NC]
RewriteCond %{DOCUMENT_ROOT}/<var mode="pre"><#new-path/#></var>  %1 -f
RewriteRule ^(.*)$ /<var mode="pre"><#new-path/#></var>  $1 [R=permanent,L]

RewriteCond %{DOCUMENT_ROOT}/<var mode="pre"><#new-path/#></var>  index.html -f
RewriteRule ^(.*)$ /<var mode="pre"><#new-path/#></var>   [R=permanent,L]
```

The `RewriteCond` with `-f` for the first `RewriteRule` will verify that the file for the target page exists before doing the redirection there. The `%1` back reference in that condition matches the regular expression group (`(.*)`) in the `RewriteCond` above it. The `RewriteCond` for the second `RewriteRule` will verify that an <samp>index.html</samp> file exists before the following RewriteRule does a redirect of any request for a document that does not exist in the new location, including a directory level request for the `DirectoryIndex` document. That one can be omitted if you already know that the <samp>index.html</samp> file exists in the target directory.

### URL redirect to a script

<samp><var>/doc-root/</var>.htaccess</samp>

```
RewriteEngine On

RewriteCond %{REQUEST_URI} !\.<var mode="pre"><#ext#></var>
RewriteCond %{REQUEST_URI} ^/(.+)$
RewriteCond %{DOCUMENT_ROOT}/%1 !-d
RewriteCond %{DOCUMENT_ROOT}/%1 !-f
RewriteRule ^([^/]+)$ /index.<var mode="pre"><#ext#></var>  ?url=$1 [R=temp,L]
```

Here is a line-by-line explanation of that code:

1. Line 1 eliminates the index.<var>ext</var> file right off the bat, to avoid redirecting requests to the script itself.
2. In line 2, the %{REQUEST_URI}% will start with a "/" but it is not included in the subpattern, so the %1 back reference will not include the leading "/".
3. Line 3 makes sure a directory does not exist at the requested URL.
4. Line 4 makes sure a file does not exist at the requested URL.
5. Line 5 does a temporary redirect (HTTP 302) to the index.<var>ext</var> script with the original URL path as the `url` parameter in the query part of the rewritten URL.

### HTTP Status Code Flags in Redirect RewriteRules

**R**

**R=temp**

**R=302**

Found - use See Other or Temporary Redirect at user agent's discretion

**R=permanent**

**R=301**

Moved Permanently (Permanent Redirect)

**R=303**

See Other

**R=307**

Temporary Redirect

back to top

## URL Redirect Examples

### Examples of URL Redirects

### Permanent redirect after moving pages to a different directory

<samp>/old-path/.htaccess</samp>

```
RewriteEngine On

RewriteCond %{REQUEST_URI} ^/old-path/(.*)$ [NC]
RewriteCond %{DOCUMENT_ROOT}/new-path/%1 -f
RewriteRule ^(.*)$ /new-path/$1 [R=permanent,L]

RewriteRule ^(.*)$ /new-path/ [R=permanent,L]
```

### Permanent redirect to fix extensions after converting them from <span class="nobr">DOS 8.3</span> names to long names

<samp>/path/.htaccess</samp>

```
DirectoryIndex index.html

RewriteEngine On

RewriteRule ^(.+).htm$ /path/$1.html [NC,R=permanent,L]
```

### Permanent redirect after moving a single page to a new location

<samp>/old-path/.htaccess</samp>

```
RewriteEngine On

RewriteRule ^old-page.html$ /new-path/new-page.html [NC,R=permanent,L]
```

Any regular expression metacharacters in the RewriteRule must be escaped with a backslash (\). See Regular Expression Metacharacters above for the list of characters that need to be escaped. Therefore, in the following example, the plus sign (+) has been escaped:

<samp>/old-path/.htaccess</samp>

```
RewriteEngine On

RewriteRule ^<a mode="pre" href="http://www.SPitBalls.com/Blog/Entries/2011/7/23_Terminology_on_Goog
```

http://www.spitballs.com/Blog/Entries/2011/7/23_Terminology_on_Google_Plus.html

### URL redirect to a PHP script

<samp><var>/doc-root/</var>.htaccess</samp>

```
RewriteEngine On

RewriteCond %{REQUEST_URI} !\.php
RewriteCond %{REQUEST_URI} ^/(.+)$
RewriteCond %{DOCUMENT_ROOT}/%1 !-d
RewriteCond %{DOCUMENT_ROOT}/%1 !-f
RewriteRule ^([^/]+)$ /index.php?url=$1 [R=307,L]
```

Here is a line-by-line explanation of that code:

1. Line 1 eliminates the index.php file right off the bat, to avoid redirecting requests to the script itself.
2. In line 2, the %{REQUEST_URI}% will start with a "/" but it is not included in the subpattern, so the %1 back reference will not include the leading "/".
3. Line 3 makes sure a directory does not exist at the requested URL.
4. Line 4 makes sure a file does not exist at the requested URL.
5. Line 5 does an HTTP 307 Temporary Redirect to the index.php script with the original URL path as the `url` parameter in the query part of the rewritten URL.

back to top

***THE END***