

RewriteRule

RewriteRule is the main directive (heart) of mod_rewrite.

Syntax

Overview

The directive consists of three arguments, separated by a space:

```
RewriteRule pattern substitution [flags]
```

The third argument (flags) is optional. Let's go now into the details for each argument.

Pattern

The pattern (rule search-pattern, rule-pattern) is a defined regular expression (RegEx) to match against the URI-path (in [per-dir context](#) local filepath) of the current request. That is, everything after `_http://hostname_`, and not including any query string arguments. (For examples of how to deal with the query string, see the discussion of the [RewriteCond](#) directive.)

You must define a [RegEx](#) to say "if the URI looks like *this*, apply the substitution".

To negate the result of the [RegEx](#), prefix the [RegEx](#) with an exclamation mark.

```
RewriteRule !regex substitution [flags]
```

Substitution

The substitution (sometimes called *target*) is a string to replace the current filename with it. So it's the location you want rewrite to. The syntax of the substitution depends upon in which context you're using the [RewriteRule](#).

Once you have set the substitution value, the [RewriteRule](#) says "if the requested URI looks like *this* (pattern) go *here* (substitution) instead."

per-server context:

- `/abc/def.html` (rewriting to an URL-path)
- `http://example.com/abc/def.html` (rewriting to an absolute URL)

per-dir context:

- `/abc/def.html` (rewriting to an URL-path)
- `abc/def.html` (rewriting to a local filepath [note the missing leading slash]. In the end, mod_rewrite needs an URL-path to go into the [per-dir bcontext internal redirect]. So you can use a URL-path directly or mod_rewrite must built such an URL-path internally from your local filepath)
- `http://example.com/abc/def.html` (rewriting to an absolute URL)

Flags

see [RewriteFlags](#)

Order of processing

Consider the following snippet:

```
RewriteEngine on
RewriteCond b =b
RewriteCond c =c
RewriteRule ^/a$ /b
```

A wrong assumption is that the conditions are processed before the pattern of the [RewriteRule](#). But this is wrong. The processing order is as follows:

- The Pattern of the [RewriteRule](#) (`^/a$`) is checked first.
- If the pattern results *true*,
- the Condition `b` would be checked.
- If that Condition is true or the OR-Flag was set to `Cond b`,
- the second condition would be checked.

- If this condition is true, too,
- the substitution from the [RewriteRule](#) is applied.

Examples

```
# rewriting a request starting with /a to /b and last rule
RewriteRule ^/a /b [L]

# rewriting e.g. /user.html --> /user.php
RewriteRule ^/([^.]+)\.html$ /$1.php [L]

# matching every request, do nothing (no substitution) and set ENV myenv with the value 1
RewriteRule ^ - [E=myenv:1]
```