FREE eBook

LEARNING mod-rewrite

Free unaffiliated eBook created from
Stack Overflow contributors.

#mod-rewrite

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: mod-rewrite

It is an unofficial and free mod-rewrite ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official mod-rewrite.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with mod-rewrite

Remarks

mod_rewrite is a module for Apache. This module is used for internal rewrites (external requests that should load a different resource) and external redirects (external requests that should make the client request a different url).

mod_rewrite provides a finer control over internal rewrites than mod_alias, as the latter can only map requests to filenames. mod_rewrite provides some means of access control, but this is usually better done with mod_authz_core and mod_authz_host. mod_rewrite provides some integration with mod_proxy, but for performance reasons this integration should not be used and instead `ProxyPass` and `ProxyPassMatch` of the latter module should be used.

mod_rewrite can be set up in a way that allows for directives to be placed in the dynamic (.htaccess) configuration files. For performance reasons, one should always use the static (httpd.conf) configuration file whenever possible.

Versions

Version	Release date
2.2	2015-07-17
2.4	2016-07-05

Examples

Installation

mod_rewrite must be enabled before being used on an Apache server.

Debian/Ubuntu

Run `a2enmod rewrite`

Then restart Apache with `service apache2 restart`

General case

Add or uncomment the following line in the static configuration file (such as `httpd.conf`):

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Then restart Apache.

Using mod_rewrite in the static configuration file

Add the following directive *before* using any other mod_rewrite directive (RewriteRule, RewriteCond, RewriteBase or RewriteMap).

```
RewriteEngine on
```

By default the engine is turned off. mod_rewrite directives found while the engine is turned off are ignored. Enable it from within the virtual host context when using virtual hosts, or from specific directory contexts when applicable.

Using mod_rewrite from the dynamic configuration files

> **Important:** Using the dynamic configuration files (.htaccess) is a big performance hit. When you have access to the static configuration file (httpd.conf or something similar) you should use that instead.

In the static configuration file, allow dynamic configuration files to override "Fileinfo" using `AllowOverride`. This directive must be placed in directory context:

```
AllowOverride FileInfo
```

The filename used for dynamic configuration files is governed by the `AccessFileName` directive. By default, the dynamic configuration files are hidden files called `.htaccess`.

At the top of each dynamic configuration file containing mod_rewrite directives, add the following directive:

```
RewriteEngine on
```

Read Getting started with mod-rewrite online: https://riptutorial.com/mod-rewrite/topic/5687/getting-started-with-mod-rewrite

Chapter 2: Basic internal rewrites

Examples

Fancy url to php script

In this example, we rewrite url's of the form `http://example.com/topic/id-seoname` to a php script that takes an id as input. This example expects the rule to be in "per-directory" context.

```
RewriteEngine on

RewriteRule ^topic/([0-9]+)-[^/]*/?$ /topics.php?id=$1 [L]
```

In this example, `topic/` is the common prefix of all topics. It is followed by a number that is used by the script. Lastly, the seo name is displayed. This seo name is ignored by mod_rewrite, because it is only there for seo reasons. The second argument of `RewriteRule` contains the url to rewrite to. The placeholder `$1` is replaced with the content of the first capture group in the regex before it. In this case it will be replaced with what is matched with `([0-9]+)`.

Url with query string to php script

To match a query string, a condition must be added to the `RewriteRule`. This is done by putting `RewriteCond` directives before the corresponding rule. In the following example we dynamically internally rewrite an old url to a new url.

```
RewriteCond %{QUERY_STRING} ^name=([^&]*)$
RewriteRule ^oldscript\.php$ newscript.php?username=%1 [L]
```

Please note that to match the literal dot, we have to escape it with a slash. `%1` is replaced with the first capture group of the previous condition. In this case it is replaced by whatever is matched by `([^&]*)`.

Read Basic internal rewrites online: https://riptutorial.com/mod-rewrite/topic/6162/basic-internal-rewrites

Chapter 3: Contexts of rewrite rules

Remarks

This topic describes the two contexts in which `RewriteRule` can be used. In examples omitting `RewriteEngine on`, it is assumed this directive has occured before that example.

Examples

Rewrite rules in per-directory context

The per-directory context is a part of the static configuration file between `<Directory>` and `</Directory>` tags. The entire content of dynamic configuration files is within the per-directory context of the folder in which the .htaccess resides.

`RewriteRule`'s in per-directory context match against the part of an url after the protocol, hostname, port and prefix of the directory in which they reside, and before the query string.

In the static configuration file

When the following rule is used on the url `http://example.com/foo?id=1`, the regex in the first argument of `RewriteRule` is matched against `foo`. The protocol (`http`), hostname (`example.com`) and prefix for this directory (`/`) are removed. At the other end, the query string (`?id=1`) is also removed.

```
<Directory "/">
  RewriteRule ^foo$ bar [L]
</Directory>
```

In the following example, using the url `http://example.com/topic/15-my-topic-name`, the first argument of `RewriteRule` would be matched against `topic/15-my-topic-name`:

```
<Directory "/topic/">
  RewriteRule ^topic/([0-9]+)-[^/]*/?$ topics.php?id=$1 [L]
</Directory>
```

In the dynamic configuration file

When the following rule is placed in a `.htaccess` file that is in the www-root folder and then used on the url `http://example.com/foo?id=1`, the first regex is matched against `foo`.

```
RewriteRule ^foo$ bar [L]
```

> In per-directory context, the matched url **never** starts with a `/`. In such a context, a directive starting with `RewriteRule ^/` will never match anything.

Rewrite rules in virtual host context

The virtual host context is a part of the static configuration file between `<VirtualHost>` and `</VirtualHost>` tags.

`RewriteRule`'s in virtual host context match against the part of an url after the protocol, hostname and port, and before the query string.

When the following rule is used for the url `http://example.com/foo?id=1`, the regex in the first argument of `RewriteRule` is matched against `/foo`.

```
<VirtualHost 1.2.3.4:80>
  ServerName example.com

  RewriteEngine on
  RewriteRule ^/foo$ /bar [L]
</VirtualHost>
```

Read Contexts of rewrite rules online: https://riptutorial.com/mod-rewrite/topic/6065/contexts-of-rewrite-rules

Chapter 4: Directives provided by mod-rewrite in Apache 2.4

Syntax

- RewriteBase URL-path
- RewriteCond TestString CondPattern
- RewriteEngine on|off
- RewriteMap MapName MapType:MapSource
- RewriteOptions Options
- RewriteRule Pattern Substitution [flags]

Examples

List of directives available in Apache 2.4

Apache 2.4 provides the following 6 directives via the `mod_rewrite` module:

1. RewriteBase
2. RewriteCond
3. RewriteEngine
4. RewriteMap
5. RewriteOptions
6. RewriteRule

The following directives, available previously in Apache 2.2 have been removed:

1. RewriteLock
2. RewriteLog
3. RewriteLogLevel

All the directives (with the exception of `RewriteMap`) defined by `mod_rewrite` can be allowed to override on a per-directory `.htaccess` through the `AllowOverride FileInfo`.

Directive	Context	Description
RewriteBase	directory, .htaccess	Sets base URL for per directory rewrite
RewriteCond	Everywhere	Defines conditions under which the rewrite action will occur
RewriteEngine	Everywhere	Sets status of rewrite engine
RewriteMap	server config, virtual host	Defines a key lookup function

Directive	Context	Description
RewriteOptions	Everywhere	Sets special *options* for rewrite engine
RewriteRule	Everywhere	Defines specific rules for rewrite engine

The context `Everwhere` means that the directive can be defined in any of the following four locations:

1. server config
2. virtual host config
3. directory context
4. .htaccess file

The `RewriteLog` and `RewriteLogLevel` directives have been merged with the global `LogLevel` directive and would be used as:

```
LogLevel rewrite:<level>
```

where `<level>` is a value from `trace8` (least significant) to `emerg` (most significant). This list is available here.

RewriteBase and RewriteEngine

Directive	Default	Context	Description
RewriteBase	None	Directory, .htaccess	Sets base URL for per directory rewrite
RewriteEngine	off	everywhere	Enable or disable runtime rewrite engine

RewriteBase

The directive specifies URL prefix to be used for substituting relative paths.

RewriteEngine

The directive, if set to `off`, will perform no runtime rewrite processing. These rules are not inherited by the virtual hosts (from server config), and will have to be defined individually.

RewriteMap

The directive defines a function which'll lookup a key in the defined map and substitutes the lookup with its replacement from the map.

The mapping function is defined with the `RewriteMap` directive itself as follows:

```
RewriteMap MAPNAME Type:Source
```

and can be referenced in any of the `RewriteCond` or `RewriteRule` directives to act as a substitution guide as follows:

```
${ MAPNAME : KEY | DEFAULT }
```

The following are valid values for `Type` in the map definition:

1. `int` - allows `toupper`, `tolower`, `escape` and `unescape` only
2. `txt` - searches a text file
3. `dbd` - looks up in a database using SQL `SELECT` statement
4. `rnd` - random lookups from text file
5. `dbm` - similar to `txt`, except that the `httxt2dbm` needs to convert the data to hashes
6. `fastdbd` - looks up in a database using SQL `SELECT` statement with caching

Read Directives provided by mod-rewrite in Apache 2.4 online: https://riptutorial.com/mod-rewrite/topic/5981/directives-provided-by-mod-rewrite-in-apache-2-4

Credits

S. No	Chapters	Contributors
1	Getting started with mod-rewrite	4444, Community, Sumurai8
2	Basic internal rewrites	Sumurai8
3	Contexts of rewrite rules	Sumurai8
4	Directives provided by mod-rewrite in Apache 2.4	hjpotter92