

Webseiten- Entwicklung

Kleine To-Dos für das Betreiben einer Webseite, Erklärungen und Anleitungen etc.

- [.htaccess - Grundlagen & Verständnis](#)
- [Regulärer Ausdruck \(RegEx\)](#)

.htaccess – Grundlagen & Verständnis

Zur Erklärung von `.htaccess` wird das folgende `mod_rewrite`-Beispiel genutzt. Damit lassen sich URL-Strukturen verstehen und eigene Regeln sauber aufbauen.

```
<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteBase /

  # %{HTTP_HOST} enthält den Hostnamen der Anfrage
  # [NC] No Case - Groß-/Kleinschreibung ignorieren
  # [L] Last - keine weiteren Regeln anwenden
  # [QSA] Query String Append - vorhandene Query-Parameter beibehalten

  # "www" entfernen
  RewriteCond %{HTTP_HOST} ^www\.(.*)$
  RewriteRule ^ https://%1%{REQUEST_URI} [L]

  # Download Server - download.aporie.me
  RewriteRule ^download/([^/]+)/([^/]+$) index.php?controller=download&source=$1&file=$2
[QSA,L]

  # Shell-Subdomain - shell.aporie.me
  RewriteCond %{HTTP_HOST} ^shell\.aporie\.me$ [NC]
  RewriteRule ^([^/]+$) index.php?controller=shell&action=$1 [QSA,L]

  # Addons
  RewriteRule ^addons/([^/]+$) index.php?controller=addons&browser=$1 [QSA,L]

  # Catch-All für Seiten
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule ^([^/]+)/?(.*)$ index.php?controller=$1&action=$2 [QSA,L]
</IfModule>
```

RegEx-Basics

RegEx beschreibt, **wie eine URL aussehen muss**, damit eine Regel greift.

Typische Bausteine:

```
^      → Start der URL
$      → Ende der URL
/      → echter Slash
[^/]   → alles außer Slash
+      → mindestens 1 Zeichen
(...)  → Gruppe (wird später als $1, $2 genutzt)
```

Beispiel:

```
^download/([^/]+)/([^/]+)$
```

Passt auf:

```
download/chrome/setup.exe
```

Ergebnis:

```
$1 = chrome
$2 = setup.exe
```

RewriteEngine & RewriteBase

```
RewriteEngine On
RewriteBase /
```

RewriteEngine On

Aktiviert das Rewrite-System überhaupt.
Ohne das passiert nichts.

RewriteBase /

Definiert den Basispfad für relative Regeln.

In deinem Fall:

- `/` = Root der Domain
- alle Regeln beziehen sich auf URLs ab Root

RewriteCond

```
RewriteCond %{HTTP_HOST} ^www\.(.*)$
```

Eine Bedingung, die **vor einer RewriteRule erfüllt sein muss**.

Wichtige Punkte:

- `%{HTTP_HOST}` → Domain der Anfrage
- RegEx wird darauf angewendet
- mehrere `RewriteCond` = UND-Verknüpfung

Beispiel:

```
www.aporie.me → matcht  
aporie.me     → matcht nicht
```

RewriteRule

Grundstruktur:

```
RewriteRule PATTERN ZIEL [FLAGS]
```

Beispiel 1: www entfernen

```
RewriteCond %{HTTP_HOST} ^www\.(.*)$  
RewriteRule ^ https://%1%{REQUEST_URI} [L]
```

Ablauf:

1. Domain ist `www.aporie.me`
2. `(.*)` speichert `aporie.me` als `%1`
3. Weiterleitung zu:

```
https://aporie.me/der/pfad
```

☐ `%{REQUEST_URI}` = kompletter ursprünglicher Pfad

Beispiel 2: Download-Route

```
RewriteRule ^download/([^/]+)/([^/]+)$ index.php?controller=download&source=$1&file=$2 [QSA,L]
```

Ablauf:

URL:

```
/download/chrome/setup.exe
```

Wird intern zu:

```
/index.php?controller=download&source=chrome&file=setup.exe
```

☐ `$1`, `$2` kommen aus den RegEx-Gruppen

Beispiel 3: Shell-Subdomain

```
RewriteCond %{HTTP_HOST} ^shell\.aporie\.me$ [NC]
RewriteRule ^([^/]+)$ index.php?controller=shell&action=$1 [QSA,L]
```

Ablauf:

Nur wenn Domain = `shell.aporie.me`

URL:

```
/install
```

→ wird zu:

```
index.php?controller=shell&action=install
```

Beispiel 4: Catch-All

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^([^/]+)/?(.*)$ index.php?controller=$1&action=$2 [QSA,L]
```

Bedeutung:

Nur wenn:

- keine echte Datei (`!-f`)
- kein echtes Verzeichnis (`!-d`)

Dann:

```
/blog/post-1
```

→

```
controller = blog
action = post-1
```

Das ist dein generisches Routing-System

FLAGS – [NC], [L], [QSA]

Flags steuern das Verhalten der Regel.

[NC] – No Case

```
[NC]
```

Groß-/Kleinschreibung ignorieren

```
Shell.Aporie.me = shell.aporie.me
```

[L] – Last

```
[L]
```

Sobald diese Regel greift:

keine weiteren Regeln werden geprüft

Sehr wichtig für Kontrolle und Reihenfolge.

[QSA] – Query String Append

```
[QSA]
```

Bestehende URL-Parameter bleiben erhalten.

Beispiel:

```
/download/chrome/setup.exe?lang=de
```

→ wird zu:

```
index.php?...&lang=de
```

Ohne QSA → `lang=de` wäre weg.

Mentales Modell (wichtigster Teil)

So kannst du das System im Kopf behalten:

1. Anfrage kommt rein
 2. Apache prüft Regel für Regel (von oben nach unten)
 3. Wenn Pattern + Bedingungen passen:
 - URL wird intern umgebaut
 - ggf. wird abgebrochen ([L])
 4. Wenn nichts greift → normale Datei wird geladen
-

Kurz zusammengefasst

- **RewriteCond** → Wann gilt die Regel?
 - **RewriteRule** → Was passiert mit der URL?
 - **RegEx** → Wie muss die URL aussehen?
 - **Flags** → Wie verhält sich Apache danach?
-

Mündlicher Versuch der Erklärung:

```
RewriteCond %{HTTP_HOST} ^www\.(.*)$
```

```
RewriteRule ^ https://%1%{REQUEST_URI} [L]
```

```
RewriteRule ^download/([^\/]+)/([^\/]+)$ index.php?controller=download&source=$1&file=$2 [QSA,L]
```

- `RewriteCond` - Wenn beim ersten Parameter die Regel (in RegEx ausgedrückt) vom zweiten Parameter zutrifft, wird das nächste "RewriteRule" ausgeführt
- `%{HTTP_HOST}` - `%{...}` ist eine Apache-Variable und `HTTP_HOST` beinhaltet die Domain (fast quasi nach dem `http://` kommen würde). Sowohl "www", Subdomains und TLD → `shell.aporie.me`, `www.aporie.me`, `aporie.me`
- `^www\.(.*)$`:
 - `^www` - es muss mit "www" beginnen.
 - `\.` - darauf folgend ein Punkt.
 - `(.*)` - 0 oder mehr Zeichen können folgen und keine Zeicheneinschränkung
 - `$` - bis zum Variablende
- `RewriteRule` - Wenn vorher ein `RewriteCond` war und die Regel zutrifft, wird dieses `RewriteRule` ausgeführt. Wenn direkt davor keins war, wird dieses `RewriteRule` ausgeführt.
- `^` - Bedeutet Zeichenanfang und da nichts darauf folgt, bedeutet es die ganze Zeichenfolge wird ersetzt durch das nachfolgende
- `https://%1` - `%1` beinhaltet die Zeichen aus der Klammer () von `RewriteCond`, was die Domain ist, bloß ohne das "www"
- `%{REQUEST_URI}` - Apache-Variable mit allem was nach der Domain (`HTTP_HOST`) kommt
- `[L]` - Wenn dieses `RewriteRule` ausgeführt wurde, wird mit der neuen Zeichenfolge diese `.htaccess`` von vorne gelesen
- `RewriteRule` - Da direkt davor kein `RewriteCond` ist, wird bei Zutreffen das RegEx angewendet
- Die Rewrite Regeln beziehen sich hauptsächlich auf `%{REQUEST_URI}` und nur wenn die explizit genannt wird, dann auch die Domain/der Host.
- `^download/([^\/]+)/([^\/]+)$` - Direkt am Anfang muss ein "download" stehen + "/" + 1 bis beliebig Zeichen außer / + "/" + 1 bis beliebig Zeichen außer / + Ausdruckende (\$)
- `index.php?controller=download&source=$1&file=$2` - Das erste "1 bis beliebig Zeichen außer /" befindet sich in `$1` und das zweite "1 bis beliebig Zeichen außer /" befindet sich in `$2`.
- Die neue Zeichenfolge ist deshalb z.B.
`index.php?controller=download&source=android&file=instander``

Regulärer Ausdruck (RegEx)

Typische Bausteine:

- ^ → Start der URL
- \$ → Ende der URL
- / → echter Slash
- [^/] → alles außer Slash
- + → mindestens 1 Zeichen
- (...) → Gruppe (wird später als \$1, \$2 genutzt)