

mod.rewrite w zastosowaniach



Strona Główna



Forum



Admin

Podstawy CMS | RkCMF | PHP-Fusion | Postnuke - MD-Pro | Jportal | Inne CMSy
 PHP i Bazy danych | Skrypty PHP | Benchmarki Skrypt | Tworzenie stron WWW | Pozycjonowanie i Promocja

mod.rewrite w zastosowaniach [Zaproponuj zmiany w artykule]

Serwer Apache ma bardzo ciekawą opcję - mod.rewrite. Umożliwia on "przepisanie" adresu wynikowej strony o określonej nazwie na inną nazwę. Np. index.html może stać się strona.html, index.php może być index.html, index.php?mod=test&strona=1 może być test1.html itd. Zalety mod.rewrite pojawiają się gdy używamy skryptów, CMSów napisanych w PHP lub innym języku dynamicznego generowania stron www. Długie linki stają się krótsze oraz co ważne przyjazne dla wyszukiwarek, gdyż nie wszystkie indeksują dynamiczne strony z ?, &, =, w linkach.

● Krok pierwszy - Czy mod.rewrite u nas działa?

Na darmowych serwerach raczej nie będzie działał (na lycosie nie działa). Jak to łatwo sprawdzić? Zrobić .htaccess z prostą regułą i zobaczyć czy działa - patrz niżej.

Plik .htaccess traktowany jest jako ukryty (serwery unixowe). By zobaczyć w kliencie FTP pliki .htaccess musimy zaznaczyć odpowiednią opcję. Dla Total Commandera będzie to w zakładce Sieć - "Pokaż ukryte Pliki".

● Krok drugi - RewriteRule i .htaccess

Mod.rewrite sterowany jest przez plik .htaccess. Edycję pliku .htaccess zaczynamy od dodania:

```
Options FollowSymLinks
RewriteEngine On
```

Następnie dodajemy zasady przepisywania stron, tzw RewriteRule. Najprostsza reguła:

```
RewriteRule ^index.html$ index.php [L]
```

Index.html, pierwsza nazwa to link jaką mamy otrzymać, index.php to link do przepisania. Umieść taki plik .htaccess na serwerze. Jeżeli pojawi się **Internal Server Error** lub podobne błędy oznacza to że serwer nie obsługuje mod.rewrite albo całego .htaccess! Czasami żaden błąd nie pojawi się a przepisywanie linków nie będzie działać - mod.rewrite nie jest włączony.

Po umieszczeniu powyższe reguły w .htaccess oba adresy - index.php i index.html będą otwierały tą samą stronę. Inne reguły tego typu (link bez zmiennej):

```
RewriteRule ^login.html$ login.php [L]
RewriteRule ^banner.html$ banner.php [L]
RewriteRule ^links.html$ links.php [L]
RewriteRule ^gbook.html$ module.php?op=gbook [L]
```

Ta są proste reguły zastępujące określone linki bez części "zmiennych". Ostatni przykład jest już bardziej zaawansowany, lecz też jest "stały", nie ma elementów "zmiennych", tj. takich, których zmiana wywoła zmianę adresu wynikowego. Przykład, mamy linki:

```
links.php?id=1
links.php?id=2
links.php?id=3
links.php?id=4
```

Jak za pomocą 1 reguły przepisywać te pliki? A takiej:

```
RewriteRule ^links([^-]+).html$ links.php?id=$1 [L]
```

Co my tam mamy. Zamiast zmiennej - cyfry (1, 2, 3, 4) dałem \$1 - zmienną nr.1 a w linku wynikowym dałem coś takiego: ([^-]+) - ta "frazę" zwróci w tym miejscu wartość zmiennej, \$1. A więc links.php?id=1 zamieni się na links1.html, links.php?id=3 na links3.html. Oto kolejny przykład takiej samej sytuacji:

```
articles.php?topic=1
articles.php?topic=2
articles.php?topic=3
```

Reguła:

```
RewriteRule ^arttopic-([^-]+).html$ articles.php?topic=$1 [L]
```

Co da odpowiednio:

```
arttopic-1.html
arttopic-2.html
arttopic-3.html
```

A gdybyśmy mieli kilka zmiennych w linku? Np.:

```
forum.php?page=&cmd=show&id=1&category=2
forum.php?page=&cmd=show&id=2&category=6
```

Zmieniają się cyfry przy id i category. Za nie wstawiamy zmienne:

```
forum.php?page=&cmd=show&id=$1&category=$2
RewriteRule ^post([^-]+)cat([^-]+).html$ forum.php?page=&cmd=show&id=$1&category=$2 [L]
```

Pierwszy ([^-]+) zwróci wartość \$1, drugi ([^-]+) wartość \$2. czyli dla pierwszego linku otrzymamy

```
post1cat2.html
```

Inny przykład:

```
adverts.php?cmd=show_cat&id=1&type=2
adverts.php?cmd=show_cat&id=$2&type=$1
RewriteRule ^adverts-([^-]+)-([^-]+).html$ adverts.php?cmd=show_cat&id=$2&type=$1 [L]
```

W tym przykładzie najpierw dałem \$2 a potem \$1. Jaki to da efekt? A taki że otrzymamy taki link końcowy:

```
adverts-2-1.html
```

Pierwsze ([^-]+) zwraca \$1, mimo iż jest ona "dalej", po \$2.

A oto plik .htaccess zawierający reguły przepisywania dla jPortala. Aktualny plik .htaccess znajdziesz w modzie "Krótkie Linki"

```
Options FollowSymLinks
RewriteEngine On

RewriteRule ^index.html$ index.php [L]
RewriteRule ^login.html$ login.php [L]
RewriteRule ^banner.html$ banner.php [L]
RewriteRule ^links.html$ links.php [L]
RewriteRule ^forum.html$ forum.php [L]
RewriteRule ^articles.html$ articles.php [L]
RewriteRule ^tnews.html$ tnews.php [L]
RewriteRule ^download.html$ download.php [L]
RewriteRule ^gbook.html$ module.php?op=gbook [L]
RewriteRule ^recomm.html$ module.php?op=recomm [L]
RewriteRule ^mailer.html$ mailer.php [L]

# Linki
RewriteRule ^links([^-]+).html$ links.php?id=$1 [L]

# Artykuły
RewriteRule ^arttopic-([^-]+).html$ articles.php?topic=$1 [L]
RewriteRule ^articles-([^-]+).html$ articles.php?id=$1 [L]

# Newsy kategorie
RewriteRule ^newstop-([^-]+).html$ news.php?topic=$1 [L]

# Download
RewriteRule ^download-([^-]+).html$ download.php?id=$1 [L]

# Forum
RewriteRule ^forumcat-([^-]+).html$ forum.php?category=$1 [L]
```

Dodany: 2004-09-01 | Autor: Riklaunim | [Dodaj komentarz](#)