



mod_rewrite - Omskrivning af urls

I denne artikel bliver der gennemgået apache serverens smarte funktion til omskrivning af urls.

Skrevet den **03. feb 2009** af **morteeart** | kategorien **Webservere / Apache** | ★★★★★

Mod_rewrite - Voodoo til Apache

I denne artikel vil jeg gennemgå apache serverens smarte funktion til omskrivning af urls.

[1 Hvad er mod_rewrite ?]

Mod_rewrite er et modul til apache serveren som kan lave `www.domæne.dk/produkter.php?id=1` om til `www.domæne.dk/produkter/1`

Dette er smart, ser smart ud, og fjerner nogle sikkerhedshuller mht. SQL injektion i urls og lign.

[2 Hvem kan bruge mod_rewrite ?]

Alle kan bruge mod_rewrite, det eneste det kræver er en apache server hvor modulet er slået til. I næste afsnit bliver der gennemgået server opsætning.

mod_rewrite aktiveres via. en .htaccess fil, med nogle regulære expressions i.

[3 Opsætning af sin server]

(Skip dette afsnit hvis du ikke har egen server)

For at kunne bruge mod_rewrite på sin server kræves der følgende indstillinger i httpd.conf

```
LoadModule rewrite_module modules/mod_rewrite.so (Fjern evt. # foran, hvis det allerede står der)

AddModule mod_rewrite.c (Fjern evt. # foran, hvis det allerede står der)

AllowOverride skal være sat til "All"

AccessFileName .htaccess (Den er sat korrekt som standart)
```

HUSK at genstarte serveren efter disse indstillinger er rettet, det er en klassisk fejl at glemme dette.

[4 Brugen af mod_rewrite / .htaccess]

For at kunne bruge mod_rewrite skal vi bruge en .htaccess fil

Denne kan laves med notepad, men det er MEGET VIGTIGT at navnet er: .htaccess (Altså uden noget før .)
Efter du har lavet din .htaccess fil, så kan du lægge den ind i den mappe du vil arbejde med.

Og lad os så komme i gang.

Først et eksempel, og så forklaring:

```
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-F
RewriteRule ^/produkter/([0-9]+)$ /produkt.php?id=$1
```

Forklaring:

RewriteEngine

Sættes til "On" , fordi den er som standart "Off", og det virker ikke uden.

RewriteBase

Angiver det base directoty (hovede mappe) som vi vil arbejde med,
i dette tilfælde " / " altså samme mappe som .htaccess ligger i.

RewriteCond

RewriteCond er en slags if-sætning i mod_write's.
Hvis den returnerer true vil den efterfølgende RewriteRule blive udført.

Se dokumentationen for liste over kommandoerne (http://httpd.apache.org/docs/mod/mod_rewrite.html)

RewriteRule

```
RewriteRule ^/produkter/([0-9]+)$ /produkt.php?id=$1
```

Det essentielle med .htaccess filen, RewriteRule er den "regel" som angive hvad som skal betrages som hvad.

Dette gøres med en regulær expression.

`^/produkter/([0-9]+)$` Bliver altså til `/produkter.php?id=$1` hvor `$1` er `123` i `domæne.dk/produkter/123`

Det smarte er at hvis du skriver `domæne.dk/produkter/abc` vil den ikke reagere overhovedet, og derfor ingen action.

`[0-9]` angiver at det er tallene 0-9 der må bruges

Hvis du f.eks. laver

```
RewriteRule ^/brugere/([A-Za-z0-9]+)$ /produkt.php?id=$1
```

Så må du bruge A-Z (der er forskel på store og små bogstaver), a-z og 0-9 på én gang.

Du kan med f.eks. et php-script vise resultatet af din mod_rewrite:

```
<?php
if(isset($_GET['id']))
{
    $id = $_GET['id'];
    echo "Du har valgt produkt ID: ".$id;
}
else
{
    echo "Du har ikke valgt et produkt ID";
}
?>
```

Bemærk at pga. !-F parameteret i RewriteCond , må filen ikke hedder produkter.php.
Men f.eks. produkt.php som jeg har lavet det til.

Du kan også lave flere Rewrite rules, f.eks.:

```
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-F
RewriteRule ^/produkter/([0-9]+)$ /produkt.php?id=$1
RewriteRule ^/kategori/([0-9]+)$ /kategorier.php?id=$1
RewriteRule ^/brugere/([A-Za-z0-9]+)$ /produkt.php?id=$1
```

[5 For dem som vil vide mere]

Hvis i vil vide mere, og have mere uddybende forklaringer vil jeg anbefale at læse dokumentationen fra Apache, som er på engelsk.

Det er faktisk ret nemt at forstå.

http://httpd.apache.org/docs/mod/mod_rewrite.html

Changelog

3. Januar 2005 - Lidt slå fejl rettet.

4. Januar 2005 - Endnu flere slå fejl rettet. (jeg er ret god til at glemme dobbelt s)

Mht. artikelens sværhedsgrad så har den jo aldrig været ment til nybegyndere, da regulær expressions jo aldrig har været simple :-)

Men når i skriver "mere uddybende" så skriv gerne hvad i vil have uddybet.

Kommentar af mcookie d. 04. jan 2005 | 1

God artikel ;=)

Kommentar af medions d. 04. jan 2005 | 2

God og flot artikel! - Noget der tit har været spurgt om her på E.

Kommentar af madsass d. 03. jan 2005 | 3

Lidt flere detaljer ville have gjort underværker. Og der er stadig fejl. ".htacces (Altså uden noget før .)"

Kommentar af sorenbs d. 02. jan 2005 | 4

Meget nyttigt. Lidt rodet forklaring ind imellem, men ellers god og gratis :)

Kommentar af webgon d. 04. jan 2005 | 5

Vil sige at den er lige på grænsen til at være 'for' teknisk, men ellers god. En lidt mere uddybende forklaring ville være godt :-)

Kommentar af lyf d. 03. jan 2005 | 6

I afsnit 4 glemmer du det sidste s i .htaccess flere gange ;-) Ku være nogen bliver forvirret af det. Ellers godt med en artikel om det emne, det manglede!

Kommentar af alister_crowley d. 21. jan 2005 | 7

Rigtig god, men der mangler dog info omkring Options -MultiViews og brug af flere RewriteCond's.