

URL rewrite zonder mod_rewrite

Inleiding

Naast mod_rewrite en isapi_rewrite is er ook een andere manier om urls gebruiksvriendelijk te maken. Hiervoor maken we gebruik van de `$_SERVER['PATH_INFO']` variabele.

Voordelen

Een voordeel is dat je niet meer een hele verzameling rewrites in bijvoorbeeld htaccess hoeft te gaan schrijven.

Een ander voordeel is dat je ook nette urls kan maken op bijvoorbeeld IIS servers, waar geen isapi_rewrite filter geïnstalleerd is.

Hoe werkt het?

Doel

Wat we gaan doen is deze url:

`http://site.com/index.php?pagina=gastenboek&actie=plaatsen`
veranderen in:

`http://site.com/index.php/gastenboek/plaatsen`

De code

De code opzich is vrij eenvoudig, dat is ook het mooie van deze manier:

```
<?php
$pathInfo = explode('/', trim($_SERVER['PATH_INFO'], '/'));
$pagina = $pathInfo[0];
$actie = $pathInfo[1];
?>
```

Of zo je wil:

```
<?php
list($pagina, $actie) = explode('/', trim($_SERVER['PATH_INFO'], '/'));
?>
```

Alternatief

Een nadeel van de gebruikte methode is dat in dit geval altijd het eerste gedeelte van de url de pagina moet bevatten. Een manier om te zorgen dat je weet wat je krijgt kan zijn door een url samen te stellen als:

`http://site.com/index.php/pagina/gastenboek/actie/plaatsen`

Het eerste gedeelte zegt nu wat voor soort gegeven er in het tweede gedeelte volgt, enzovoort.

```
<?php
$pathInfo = explode('/', trim($_SERVER['PATH_INFO'], '/'));

$myPath = array();
for($i=0, $count=count($pathInfo); $i+1<$count && $count>1; $i+=2) {
    $myPath[$pathInfo[$i]] = $pathInfo[$i+1];
}

var_dump($myPath);
?>
```

Tips

Apache Mod_rewrite

Hoewel je in principe geen rewrite regels meer nodig hebt, zou je er nog wel een kunnen toevoegen. Bijvoorbeeld deze voor Apache mod_rewrite:

```
RewriteEngine On
RewriteCond %{SCRIPT_FILENAME} !-f
RewriteCond %{SCRIPT_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1
```

Het toevoegen van deze rewrite regel heeft als gevolg dat alle requests naar niet bestaande bestanden en mappen automatisch worden doorgestuurd naar index.php, waardoor het niet meer nodig is index.php op te nemen in de url. Onze url ziet er dan als volgt uit:

`http://site.com/gastenboek/plaatsen`

of

`http://site.com/pagina/gastenboek/actie/plaatsen`

Apache MultiViews

In apache is er nog een andere optie dan mod_rewrite om de urls een stukje netter te maken, namelijk multiviews. Als multiviews in apache aanstaat, dan betekent dit dat als apache geen directory met de opgegeven naam vindt, hij opzoek gaat naar een bestand met die naam waarbij de extensie er niet toe doet. Op die manier kunnen wij onze url er als volgt uit laten zien:

`http://site.com/index/gastenboek/plaatsen`

of

`http://site.com/index/pagina/gastenboek/actie/plaatsen`

Apache multiviews via httpd.conf

Ga opzoek naar de volgende regel:

```
Options Indexes FollowSymLinks
```

En voeg daar MultiViews aan toe, zo:

```
Options Indexes FollowSymLinks MultiViews
```

Vergeet vervolgens niet om Apache opnieuw op te starten!

Apache multiviews via htaccess

In het geval je geen toegang hebt tot de httpd.conf, kan je multiviews ook aanzetten via htaccess. Maak dan een htaccess bestand aan en plaats daar de volgende regel in:

```
Options Indexes FollowSymLinks MultiViews All
```

Zend Framework

Zend Framework maakt ook gebruik van `$_SERVER['PATH_INFO']` waardoor het mogelijk is dit framework zonder mod_rewrite te gebruiken en toch nette urls te hebben.

Het kan zijn dat je de volgende code nodig hebt om dit te laten werken:

```
<?php
$request=new Zend_Controller_Request_Http();
$request->setBaseUrl('/index.php');
$request->setPathInfo(substr($request->getRequestUri(), strlen($request-
>getBaseUrl())));
?>
```

Vergeet vervolgens niet om het request object mee te geven aan de dispatcher:

```
<?php
$frontController->dispatch($request);
?>
```