

Harder, better, faster, stronger

Kleine aanpassingen die je website veel sneller maken



Contents

1.	Inleiding	3
	Wat is snel?	3
	Wat lijkt snel?	3
2.	Gebruik de cache	3
3.	Style sheets horen in de head sectie	4
4.	JavaScript hoort in de body sectie	4
5.	Beperk het aantal server requests.....	5
6.	Een woordje uitleg over afbeeldingen	6
7.	Bestanden comprimeren met gzip	6
8.	YSlow voor Firefox	7
9.	Bedankt Rudi	7

1. Inleiding

Dus, je hebt iemand overtuigd om je website te bezoeken. Waarom zou je ze laten wachten? Worden we niet wekelijks om de oren geslagen met onderzoeken die aantonen dat een bezoeker een website in slechts enkele microseconden beoordeelt?

Daarom hebben we enkele eenvoudige technieken verzameld die op minder dan een uur toegepast kunnen worden en die je zeker wat zullen opleveren.

Eerst zullen we eens kijken naar het verschil tussen wat snel is en snel lijkt:

Wat is snel?

- Focus op de tijd die verstrijkt tussen de start en het einde van een proces
- Theoretisch
- Verantwoordelijk voor slechts 5% van het probleem

Wat lijkt snel?

- Focus op de perceptie/ervaring van de gebruiker
- Praktisch
- Verantwoordelijk voor 95% van het probleem

Tijd voor actie...

2. Gebruik de cache

Voor de 0,001% onder u die niet weten wat de browser cache is, zo werkt het: Elke keer je browser een bestand downloadt, bijvoorbeeld een afbeelding of een stylesheet, zal hij het in een daarvoor gereserveerde map plaatsen zodat hij niet bij elke nieuwe pagina het bestand opnieuw hoeft op te vragen. Dat bespaart niet enkel op bandbreedte, het zorgt er ook voor dat een pagina sneller zal inladen.

Helaas maken de meeste sites hiervan geen of toch tenminste te weinig gebruik. Daardoor wordt de browser verplicht de meeste informatie bij elke bekeken pagina opnieuw op te vragen. Door enkele lijnen toe te voegen aan het .htaccess bestand kan je zelf aangeven welk type de browser moet opslaan in de cache en hoe lang het mag duren alvorens de bestanden opnieuw opgevraagd dienen te worden:

```
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType text/html "access plus 1 seconds"
ExpiresByType image/gif "access plus 2 years"
ExpiresByType image/jpeg "access plus 2 years"
ExpiresByType image/png "access plus 2 years"
ExpiresByType text/css "access plus 2 years"
ExpiresByType text/javascript "access plus 2 years"
ExpiresByType application/x-javascript "access plus 2 years"
</IfModule>
```

Vertaald naar verstaanbare taal zegt dit aan een browser dan elk media bestand (gif, jpg, css, ...) hetzelfde zal blijven gedurende 2 jaar en dus in de cache mag opgeslagen worden. Aangezien de content van de website zelf snel kan veranderen zullen bestanden die zichzelf identificeren als html onmiddellijk verlopen en dus niet in de cache opgeslagen worden.

Dit zorgt er natuurlijk niet voor dat de allereerst bekeken pagina sneller zal laden, maar vanaf de tweede pageview zou het verschil al merkbaar moeten zijn.

Hier hoort nog wel een kleine opmerking bij. Je hebt de browser duidelijk gemaakt dat bepaalde media bestanden niet zullen wijzigen gedurende 2 jaar. Als je nu kleine wijzigingen aanbrengt aan één van de bestanden zal de browser deze standaard niet opmerken. Dit kan omzeild worden door aanpassingen te doen aan de verlooptijd, of door eenvoudigweg de bestandsnaam te wijzigen.

Aan de hand van ETags communiceren browsers en servers met elkaar om na te gaan of de bestanden in de cache overeenkomen met de bestanden op de server. De meeste mensen hebben geen idee hoe dit exact in z'n werk gaat, en we zullen eerlijk zijn, wij ook niet. Wat we wel weten is dat het voor de snelheid van je website best is deze techniek niet toe te passen. Dit kan je doen door aan je .htaccess bestand volgende lijn toe te voegen:

```
FileETag none
```

3. Style sheets horen in de head sectie

Zodra oudere versies van Internet Explorer style sheets tegenkomen stoppen ze met het renderen van de pagina. Door de style sheet onder te brengen in de head sectie van het document kunnen we ervoor zorgen dat de rendering niet onderbroken wordt. Bovendien voorkomen we hierdoor de zogenaamde infamous flash of unstyled content (<http://bluerobot.com/web/css/fouc.asp/>).

Maar er zijn nog zaken die je CSS beter kunnen maken. Alle opmaak onderbrengen in een apart CSS bestand zal de bestandsgrootte van je pagina's dramatisch verkleinen en je website dus sneller maken. Splitsen over meerdere CSS bestanden is mogelijk, maar ook hier geldt weer met mate.

Style sheets hebben nogal snel de neiging om lang, te lang te worden. Je kan ze verkorten door alle comments en onnodige white space te verwijderen. Op <http://www.cleancss.com/> vind je een goede webbased CSS cleaner.

En vergeet niet dat we spreken over Cascading Style Sheets, met de nadruk op Cascading. Een basisfont of -kleur dien je slechts één keer te definiëren. Elke child zal properties overerven van z'n parent.

4. JavaScript hoort in de body sectie

JavaScripts hebben de onhebbelijke gewoonte het renderen van web pages te onderbreken. Door JavaScripts onderaan de code aan te brengen zorgen we ervoor dat de pagina eerst opgebouwd wordt en de bezoeker de pagina sneller te zien krijgt.

Daarnaast hebben we bovendien het voordeel dat als een extern script (bijvoorbeeld een web analytics script) niet beschikbaar is of niet geladen kan worden, het de weergave van de pagina niet in de weg staat. De bezoeker zal zo niet merken dat er iets fout loopt en zonder verpinken zijn aankoopproces verderzetten.

JavaScripts worden best ondergebracht in hun eigen bestanden. Dit vermindert de grootte van pagina's en de browser zal ze ook weer onderbrengen in de eerder vernoemde cache.

Net als CSS bestanden kunnen JavaScripts ook verkleind worden door commentaren en white space te verwijderen. Daarnaast bestaan er JavaScript minifiers die de code nog verder kunnen comprimeren. Op [CompressorRater](http://compressorater.thruhere.net/) (<http://compressorater.thruhere.net/>) vind je een overzicht.

5. Beperk het aantal server requests

Script libraries kunnen een site gemakkelijk dynamischer maken en schitterende effecten toevoegen. Ze hebben echter een groot nadeel: de meeste scripts vereisen individuele bestanden voor script en opmaak. Dit zorgt voor een toename in server requests. Vermits dit de snelheid van het laden van een pagina negatief beïnvloed, is het best verschillende kleine bestanden te verenigen in één groter bestand.

Hetzelfde geldt voor rollover images. Dankzij CSS sprites kan je verschillende afbeeldingen combineren in één grote afbeelding. Op [A List Apart](http://www.alistapart.com/articles/sprites/) (<http://www.alistapart.com/articles/sprites/>) kan je hierover een interessante tutorial lezen. Een website die dit tot in het extreme toepast is Amazon. Dankzij CSS sprites besparen ze jaarlijks een enorme hoeveelheid bandbreedte en server requests. Als illustratie de ene afbeelding die voor zowat de volledige opbouw van de Amazon web pages verantwoordelijk is:



6. Een woordje uitleg over afbeeldingen

Elk afbeeldingsformaat heeft z'n eigenschappen en voor- en nadelen. Alvorens afbeeldingen aan een website toe te voegen is het misschien nuttig even stil te staan bij welk type afbeelding gebruikt dient te worden. De verschillen situeren zich hoofdzakelijk op het gebied van compressieratio en kwaliteit.

Foto's en afbeeldingen zonder al te veel details vragen om een jpeg. Het jpeg formaat is in staat om grote oppervlakken van een afbeelding of foto te comprimeren zonder al te groot kwaliteitsverlies. Als gevolg hiervan zal een afbeelding met veel details snel groot worden in jpeg formaat.

Afbeeldingen met lijnen, tekst of veel details kunnen best in png formaat gebruikt worden. Het compressie algoritme van png is uitermate geschikt voor dit type van afbeeldingen, evenals dat van het gif formaat. Al levert png over het algemeen betere resultaten dan gif.

7. Bestanden comprimeren met gzip

De details over hoe het exact werkt zouden ons misschien iets te ver leiden, maar gzip maakt het mogelijk tekstbestanden te comprimeren alvorens ze van de server naar de browser worden doorgestuurd. De browser pakt deze dan na ontvangst uit. De overgrote meerderheid van de browsers (IE, Firefox, Opera, Safari, Chrome, ...) ondersteunt deze techniek. Enkel exotische of antieke browsers durven deze functionaliteit niet te ondersteunen.

Aangezien de gecomprimeerde bestanden kleiner zijn besparen we hier ook weer op de hoeveelheid data die van op de server bij de browser moet geraken. Gevolg: snellere laadtijden.

De meeste content management systems voorzien in een optie om de pagina's gecomprimeerd te verzenden. Maar hoe zit het dan met style sheets en JavaScripts? Wel, deze kunnen ook gecomprimeerd verzonden worden. Je plaatst je CSS en JavaScript bestanden in een aparte directory en in die map plaats je daarenboven twee extra bestandjes: een .htaccess bestand en een PHP bestand dat de bestanden kan comprimeren.

In het .htaccess bestand komt volgende code:

```
AddHandler application/x-httpd-php .css
php_value auto_prepend_file gzip-css.php
php_flag zlib.output_compression On
```

```
AddHandler application/x-httpd-php .js
php_value auto_prepend_file gzip-js.php
php_flag zlib.output_compression On
```

Het PHP bestand voor de CSS compressie bestaat uit volgende code:

```
<?php
header("Content-type: text/css; charset: UTF-8");
header("Cache-Control: must-revalidate");
$offset = 2678400; /* One month */
```

```
$ExpStr = "Expires: " .  
gmdate("D, d M Y H:i:s",  
time() + $offset) . " GMT";  
header($ExpStr);  
  
?>
```

Het PHP bestand voor de JavaScript compressive bestaat uit deze code:

```
<?php  
  
header("Content-type: text/js; charset: UTF-8");  
header("Cache-Control: must-revalidate");  
$offset = 2678400; /* One month */  
$ExpStr = "Expires: " .  
gmdate("D, d M Y H:i:s",  
time() + $offset) . " GMT";  
header($ExpStr);  
  
?>
```

Vergeet niet beide PHP bestanden de juiste naam te geven. In dit geval gzip-css.php en gzip-js.php.

8. YSlow voor Firefox

Yahoo beschikt over een volledig team dat zich louter bezig houdt met de research naar hoe websites sneller kunnen gemaakt worden en bericht daar ook uitgebreid over. Een groot deel van onze inspiratie is dan ook van hen afkomstig.

Hetzelfde Yahoo team heeft een interessante Firefox add-on (<http://developer.yahoo.com/yslow/>) in elkaar gestoken die je toelaat te monitoren hoe websites presteren op het gebied van snelheid. Het geeft je bovendien een inzicht in hoe een website gebruikt maakt van de cache.

YSlow is een extensie van de Firebug add-on, die je kan vinden op <http://getfirebug.com/>

9. Bedankt Rudi

We hopen dat je iets gehad hebt aan dit document en we wensen je veel plezier toe bij het tweakken van de prestaties van je website. Commentaar zoals steeds welkom op geeks@queromedia.com

Bezoek ook eens ons [Zoekmachine Marketing Blog](#).