

Performance mit GZIP

== Instrumente ==

Um die Ladezeiten der Webseiten besser einschätzen zu können, nutzen wir Firefox mit aktiviertem Firebug Plugin.

== Original ==

Als Vorlage für dieses Tutorial nehmen wir die Webseite der deutschen Wikipedia. Die Seite ist schon hoch optimiert, dennoch können wir das Beispiel nutzen um noch besser zu sein.

Link: demo.easy-coding.de/performance/original

== Optimierung - Dynamische Seiten ==

Wir nutzen die eigene, sehr schlanke Bibliothek HeaderUtil. Der Quellcode der Datei ist ausreichend dokumentiert. In allen dynamischen Skripte mit PHP fügen wir das Script wie folgt an den Beginn der Datei ein.

Quellcode

```
1. --- original/index.php 2009-10-29 21:10:09.000000000 +0100
2. +++ gzip/index.php 2009-10-29 21:15:01.000000000 +0100
3. @@ -1,4 +1,7 @@
4. -<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
5. +<?php
6. +require_once 'HeaderUtil.php';
7. +HeaderUtil::startOutput();
8. +?><!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
9. <html dir="ltr" xml:lang="de" xmlns="http://www.w3.org/1999/xhtml" lang="de"><head>
```

[Blockierte Grafik: <http://demo.easy-coding.de/performance/gzip/screenshot/index-original.png>]

[Blockierte Grafik: <http://demo.easy-coding.de/performance/gzip/screenshot/index-gzip.png>]

Wir sehen, dass die Index Datei von 37 KB auf 10 KB schrumpft.

== Optimierung statische Inhalte ==

Statische Inhalte lassen sich ähnlich einfach manipulieren. Wir geben sie nicht direkt an den Browser, sondern lassen die Auslieferung über ein PHP Script steuern.

Dass die Dateien dabei den selben Dateinamen behalten, lässt sich über mod rewrite realisieren.

Wir erzeugen folgende .htaccess Datei:

Quellcode

```
1. RewriteEngine On
2. RewriteRule (^.\.(js|css))$ gzip.php?file=$1&mime=$2 [L]
```

Die gzip.php sieht wie folgt aus. In der Variable \$dirs = array('wiki_files') habe ich alle erlaubten Ordner gepackt. Damit wird verhindert, dass ungewollt geschützte Dateien wie z.B. eine config.php ausgeliefert werden.

Quellcode

```
1. <?php
2. require_once 'HeaderUtil.php';
3. HeaderUtil::startOutput();
4. // input params
5.
6. $file = isset($_GET['file']) ? $_GET['file'] : exit;
7. $mime = isset($_GET['mime']) ? $_GET['mime'] : exit;
```

```

9. // settings
10. $whitelist = array();
11. $dirs = array('wiki_files');
12. $mimes = array(
13. 'js' => 'application/x-javascript',
14. 'css' => 'text/css'
15. );
16. // supported mimetype?
18. if(!array_key_exists($mime, $mimes)) {
19. exit;
20. }
22. // includes in whitelist
23. foreach($dirs as $dir) {
24. foreach(scandir($dir) as $filename) {
25. $whitelist[] = $dir.'/'.$filename;
26. }
27. }
29. if(!in_array($file, $whitelist)) {
30. exit;
31. }
32. $content = file_get_contents($file);
35. // ATTENTION: this exits if header timestamp (If-Modified-Since) is file timestamp
36. HeaderUtil::sendLastModified(filemtime($file));
37. // remember for 30 days in browser cache
39. HeaderUtil::sendMaxAge(86400 * 30);
40. // send content type
42. HeaderUtil::sendContentType($mimes[$mime]);
43. // output
45. echo $content;
46. ?>

```

Alles anzeigen

[Blockierte Grafik: <http://demo.easy-coding.de/performance/gzip/screenshot/static-original.png>]

[Blockierte Grafik: <http://demo.easy-coding.de/performance/gzip/screenshot/static-gzip.png>]

Wir sehen, dass die Gesamtdateigröße von 264 KB auf 128 KB reduziert wird.

== Demo + Download ==

Alles zu sehen gibt es unter demo.easy-coding.de/performance/original/ und demo.easy-coding.de/performance/gzip/.

Den Download der gzip Files findet ihr hier: demo.easy-coding.de/performance/gzip/download.zip