

# Performance durch Minification / Minify

== Instrumente ==

Um die Ladezeiten der Webseiten besser einschätzen zu können, nutzen wir Firefox mit aktiviertem Firebug Plugin.  
Als Minifier wird der GPL lizenzierte SmartOptimizer genutzt: [farhadi.ir/works/smarteroptimizer](http://farhadi.ir/works/smarteroptimizer)

== 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](http://demo.easy-coding.de/performance/original)

== Optimierung statische Inhalte ==

JavaScript und CSS Dateien sollen nicht mehr direkt aufgerufen werden können - stattdessen rufen wir das Script über eine PHP Datei auf.

Am besten lässt sich das 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:

## Quellcode

```
1. <?php
2. require_once 'HeaderUtil.php';
3. HeaderUtil::startOutput();
4. // input params
5. $file = isset($_GET['file']) ? $_GET['file'] : exit;
6. $mime = isset($_GET['mime']) ? $_GET['mime'] : exit;
7. // settings
8. $whitelist = array();
9. $dirs = array('wiki_files');
10. $mimes = array(
11.   'js' => 'application/x-javascript',
12.   'css' => 'text/css'
13. );
14. // supported mimetype?
15. if(!array_key_exists($mime, $mimes)) {
16.   exit;
17. }
18. // includes in whitelist
19. foreach($dirs as $dir) {
20.   foreach(scandir($dir) as $filename) {
21.     $whitelist[] = $dir.'.'.$filename;
22.   }
23. }
24. if(in_array($file, $whitelist)) {
25.   exit;
26. }
27. // minify with one of the two supported minifiers
28. require_once 'minifier/'.$mime.'.php';
29. $func = "minify_". $mime;
```

```
38. $content = $func($content);
39. // ATTENTION: this exits if header timestamp (If-Modified-Since) is file timestamp
41. HeaderUtil::sendLastModified(filemtime($file));
42. // remember for 30 days in browser cache
44. HeaderUtil::sendMaxAge(86400 * 30);
46. // send content type
47. HeaderUtil::sendContentType($mimes[$mine]);
49. // output
50. echo $content;
51. ?>
```

Alles anzeigen

Die gzip.php stammt aus dem Tutorial: [wiki]Performance mit GZIP[/wiki]

Im nachfolgenden Diff sieht man, dass nur der content durch eine bestimmte Minify Funktion verändert wird.

Die mitgelieferten Minifier stammen aus dem SmartOptimizer Projekt und sind unter der GPL lizenziert:

[farhadi.ir/works/smarterminifier](http://farhadi.ir/works/smarterminifier)

## Quellcode

```
1. --- gzip/gzip.php 2009-10-29 21:42:27.000000000 +0100
2. +++ minify/gzip.php 2009-10-29 21:42:21.000000000 +0100
3. @@ -32,6 +32,11 @@
5. $content = file_get_contents($file);
6. // minify with one of the two supported minifiers
8. +require_once 'minifier/'.$mime.'.php';
9. +$func = "minify_". $mime;
10. +$content = $func($content);
11. +
12. // ATTENTION: this exits if header timestamp (If-Modified-Since) is file timestamp
13. HeaderUtil::sendLastModified(filemtime($file));
```

Alles anzeigen

== Demo + Download ==

Alles zu sehen gibt es unter [demo.easy-coding.de/performance/original/](http://demo.easy-coding.de/performance/original/) und [demo.easy-coding.de/performance/minify/](http://demo.easy-coding.de/performance/minify/).

Den Download der gzip Files findet ihr hier: [demo.easy-coding.de/performance/minify/download.zip](http://demo.easy-coding.de/performance/minify/download.zip)