

Minimales Cache-System in PHP

Description

Trotz WordPress, Joomla und Konsorten kann es passieren, dass man seine eigene Webengine braucht. Um beispielsweise Userverhalten auszuwerten, sind hier Statistiken eine feine Sache. Doch das Laden solcher Seiten kann nervtötend sein.

Das Thema „Caching“ ist in der Webentwicklung ein riesiges Gebiet. Vorrangig werden hierbei Daten, die noch aktuell sind, zwischengespeichert und ausgegeben. Das hat u. A. den Vorteil, dass man nicht jedes mal die Datenbank mit Millionen von Abfragen belästigen muss.

Bei einem aktuellen Projekt hatte ich genau das Problem. Ich erstellte in der Administration eine Statistik-Seite. Hierbei wurden nicht nur Tabellen erzeugt und Daten ausgewertet, teilweise wurden diese auch als Grafiken ausgegeben. Das heißt, dass PHP die Diagramme in dem Moment erzeugt, in dem die Daten abgerufen werden. Bei einem geringen Datensatz und wenigen Grafiken ist das zu verschmerzen, aber irgendwann geht es in den Bereich von Sekunden – und dann nervt es.

Eine kleine Funktion

Ein simpler Weg der Verbesserung ist ein Cache, wobei mein Weg kein klassischer Cache ist. Es wird lediglich geprüft, wie alt die letzte Auswertung ist. Derzeit steht das Limit auf 10 Minuten. Das heißt: Wenn die Grafiken älter sind als 600 Sekunden, werden sie neu erstellt. In PHP geht das ganz einfach. Zur Hilfe habe ich eine kleine Funktion erstellt:

```
function check_gfx_date($filename, $time)
{
 if (file_exists($filename))
 {
  $now = gmdate('F d Y H:i:s');
  $fileTime = gmdate('F d Y H:i:s', (filemtime($filename)));
  $div = strtotime($now)-strtotime($fileTime);

  if ($div >= $time)
  {
```

```
   return false;
  } else {
   return true;
  }
 } else {
  return false;
 }
}
```

Die Funktion bekommt zwei Werte mit. Dateiname und Zeit. Die Zeit sind die 600 Sekunden, Dateiname kann irgendwas sein. Die erste Prüfung schaut, ob die Datei überhaupt existiert. Wenn sie nicht existiert, wird **false** zurück gegeben. In dem Fall wird die Grafik neu erstellt. Wenn sie existiert, wird in *$now* das aktuelle Datum mit Uhrzeit festgehalten. Dann wird in der Variable

$fileTime das Datum der Datei gespeichert. Das ist das Wichtigste. Mit *filemtime($filename)* lässt sich dies in PHP sehr einfach realisieren. Nun wird die Differenz geprüft. Ist sie >= $time (600 Sekunden), wird erneut **false** zurück gegeben, ansonsten **true**. Fertig ist der Lack.

Aufruf der Funktion

In den Funktionen, wo die Grafik erstellt wird, ruft man dann nur noch die Prüfung auf.

```
$filename = 'statistic.png';
$cacheTime = 600;

$cacheOK = check_gfx_date($filename, $cacheTime);

if (!$cacheOK)
{
     // … ab hier wird die Grafik erzeugt
}
```

Es versteht sich von selbst, dass man hier auch den Abruf der Daten mit der Abfrage umschließt. Das Erzeugen der Grafik selbst ist meistens nicht das Problem, sondern eher die Auswertung der benötigten Datensätze.

Date Created
23. Mai 2018
Author
sven