



Simpel caching med PHP

Med udbredelsen af WYSIWYG editorer glemmes ofte overvejelser om database optimering vha. caching. Hermed en simpel løsningsmodel.

Skrevet den **12. Mar 2009** af **limemedia** | kategorien **Programmering / PHP** | ★★☆☆☆☆

Med udbredelsen af WYSIWYG editorer til manipulering af indholdet på websider, ser jeg ofte at udviklerne udelukkende gemmer side data i databasen, og ved hver visning af siderne henter indholdet fra databasen - derved skabes et unødigt load på database niveauet, som kan undgås. Ved at undgå dette, skabes der ekstra "kræfter" til fx. søgninger der udføres hurtigere m.v. En god tommelfinger regel er, at alt der kan køres udenom databasen, bør køres udenom.

Jeg mener hermed ikke, at vi skal droppe databasen i ethvert WYSIWYG tool - hvad jeg mener er, at når vi som indholds redaktører gemmer en side i toolet, så gemmes både en version i databasen til nem tilgang når vi retter, men samtidig gemmes en version på disken, som vores webapplikation henter fremfor at hente i databasen. Vores database laver således intet når brugeren henter siden, og arbejder kun når redaktøren retter siden.

Ved at lade redaktørens version være i databasen, tillader vi søgninger i indholdet, som pga den formindskede load på serveren kan laves "tungere" / mere avanceret - smart ik ?

Hvad udviklere ofte glemmer er, at man kan gemme disse "cachede" filer som PHP filer, og dermed stadig have dynamisk indhold på siderne eller andre database kald - ved at bruge include til at inkludere den cachede fil, eksekveres filen inden den vises, således at enhver PHP kode i filen eksekveres

Nok snak, hermed et kodeeksempel der giver tre funktioner, loadCache, saveCache og deleteCache - navnene burde være ret sigende. Jeg bruger personligt ikke loadCache da den ikke eksekverer indhold fremfor en simpel include, men jeg har den i mit "bibliotek" for god ordens skyld.

```
<?php
function loadCache($dir, $file) {
    if (!is_readable($dir.$file)) {
        return false;
        exit;
    } else {
        $buffer = "";

        $fd = fopen ($dir.$file, "r");
        while (!feof($fd)) $buffer .= fgets($fd);
        fclose ($fd);
    }

    return $buffer;
}

function saveCache($dir, $file, $cache, $backupdir='') {
    # 0x1 : File cached
    # 0x2 : Directory doesnt exist
    # 0x4 : Directory not writable
```

```

# 0x8 : Backup directory doesnt exist
# 0x10 : Backup directory not writable

if (!is_dir($dir)) $status += 0x2;
else if (!is_writable($dir)) $status += 0x4;
else {
    $cachefile = $dir.$file;
    $backupfile = $backupdir.$file;

    if ($backupdir && file_exists($cachefile)) {
        if (!is_dir($backupdir)) $status += 0x8;
        else if (!is_writable($backupdir)) $status += 0x10;
        else {
            if (file_exists($backupfile)) unlink($backupfile);
            rename($cachefile, $backupfile);
        }
    }
}

$fp = fopen($cachefile, 'w');
fputs($fp, $cache);
fclose($fp);

$status += 0x1;
}

return $status;
}

function deleteCache($dir, $file) {
    if (file_exists($dir.$file)) unlink($dir.$file);
    return true;
}
?>

```

Mere information om brugen af bit værdier til rapportering af fejlkoder :

<http://www.eksperten.dk/artikler/229>

Kommentar af olebole d. 11. Nov 2004 | 1

Kanon, Lars! En alt for længe ventet artikel ... det er hårdt at kæmpe mod dynamisk overkill helt alene ;o)
 Måske, du kunne udvide den lidt, så mindre øvede også kan få noget ud af den - men at kalde artiklen
 'dårligt formuleret' er en decideret dårlig formulering.

At forklare metoden helt fra Adam og Eva vil dog også være en skidt idé. Har man brug for dette, kan man
 sagtens læse artiklen. Kan man ikke forstå den, har man med højglansforkromet statsgaranti en bunke
 andet at optimere - og som er vigtigere at få styr på :)

/mvh

Kommentar af nil.dk d. 09. Nov 2004 | 2

Kommentar af jensgram d. 12. Nov 2004 | 3

Stille og rolig ;)

Kommentar af mikkels d. 25. Oct 2006 | 4

Kommentar af homsabu d. 20. Jan 2005 | 5

enig med el_barto og xyborx!

alt for lidt kommentarer og forklaring - man kunne f.eks ha' kommenteret koden linie for linie med // kommentarer...

Hvis man ved hvad der snakkes om er den fin nok, men IMO ikke 5 point værd!

Kommentar af xyborx d. 12. Nov 2004 | 6

Jeg er enig med el_barto. Der mangler noget. Bl.a. en dybere forklaring af hvad funktionerne gør eller i det mindste hvordan de bruges.

Kommentar af el_barto (nedlagt brugerprofil) d. 12. Nov 2004 | 7

Jeg sidder med den der "hvor er resten henne" fornemmelse.

Jeg synes artiklen tager afsæt i for mange antagelser, tag hellere fat på en konkret måde folk laver sider på, f.eks. `index.php?side=etellerandet` og forklar derfra.

Kommentar af vodka101 d. 02. Jan 2005 | 8

Hvad?