



Opret egne logfiler til din hjemmeside.

Logfiler for hjemmesider er endnu ikke særligt udbredt her hjemme. Det er syndt da det er et rigtigt godt værktøj for webmastre til at holde øje med deres sider. Logfiler er små hjælpere der konstant holder øje med din side og kan derfor opdage ting du sa

Skrevet den **09. Feb 2009** af **hkb-x** i kategorien **Programmering / PHP** | ★★★★★

Logfiler for hjemmesider er endnu ikke særligt udbredt her hjemme. Det er syndt da det er et rigtigt godt værktøj for webmastre til at holde øje med deres sider. Logfiler er små hjælpere der konstant holder øje med din side og kan derfor opdage ting du sandsynligvis aldrig ville have opdaget.

I denne artikel vil jeg gennemgå det alt det tekniske bag oprettelsen og brugen af en logfil og så nogle små idéer som inspiration til brugen af logfiler. Dette er min første artikel så i må endelig komme med ider til rettelser osv. på: bjerregaard_6@msn.com

Oprettelse af logfilen

Logfilen er bare et ganske almindeligt .txt dokument du ligger i din hjemmesides rod. Det eneste du skal huske er at højreklikke på filen og gå ind under Egenskaber. Her sætter du hak i alle 9 felter under Tilladelser. Så kan vi skrive til filen.

Log Scriptet

Scriptet er lavet som en funktion så det er nemt og overskueligt at skrive til logfilen:

```
function LOGFILE ($input) {

    $input = str_replace("#IP", $_SERVER['REMOTE_ADDR'], "$input");
    $input = str_replace("#REF", $_SERVER['HTTP_REFERER'], "$input");
    $input = str_replace("#AGENT", $_SERVER['HTTP_USER_AGENT'], "$input");
    $input = str_replace("#HOST", $_SERVER['REMOTE_HOST'], "$input");
    $input = str_replace("#PORT", $_SERVER['REMOTE_PORT'], "$input");
    $input = str_replace("#SELF", $_SERVER['PHP_SELF'], "$input");

    $sidsteLinie = end(file("logfile.txt"));
    if(preg_match('/\[[.+\?\\]\/', $sidsteLinie, $match))
    $olddate = str_replace("[", "", "$match[0]");
    $olddate = str_replace("]", "", "$olddate");

    $handle = fopen("logfile.txt", "a");
    if($olddate < date("d-m-Y")) { fwrite($handle, "\n----- ".date("d-m-Y")." -----
    -----\n\n"); }
    fwrite($handle, "[".date("d-m-Y H:i:s")."] ".$input." \n");
    fclose($handle);
}
```

Ok... det kan måske nok virke lidt uoverskueligt så lad os bryde funktionen op i et par smådele:

```
$indput = str_replace("#IP", $_SERVER['REMOTE_ADDR'], "$indput");
$indput = str_replace("#REF", $_SERVER['HTTP_REFERER'], "$indput");
$indput = str_replace("#AGENT", $_SERVER['HTTP_USER_AGENT'], "$indput");
$indput = str_replace("#HOST", $_SERVER['REMOTE_HOST'], "$indput");
$indput = str_replace("#PORT", $_SERVER['REMOTE_PORT'], "$indput");
$indput = str_replace("#SELF", $_SERVER['PHP_SELF'], "$indput");
```

Med PHP's indbyggede funktion `str_replace()` kan vi erstatte nogle hjemmelavede forkortelser. Dette gør det hele meget nemmere og mere overskueligt når vi skriver i logfilen. F.eks kan man i øverste linie se at `#IP` erstattes af `$_SERVER['REMOTE_ADDR']`. Så hvis vi i scriptet skriver "Min ip adresse er `#IP`" så vil der i logfilen stå "Min ip adresse er 235.155.10.5" (eller hvad ip adressen nu end er). Som du kan se er der en række funktioner inkluderet. Denne liste kan forkortes / forlænges efter eget valg husk dog at ikke alle af dem understøttes af alle servere.

```
$sidsteLinie = end(file("logfile.txt"));
if(preg_match('/\[[.+\?\\]\/', $sidsteLinie, $match))
$olddate = str_replace("[", "", "$match[0]");
$olddate = str_replace("]", "", "$olddate");
```

I denne del starter vi med at finde den sidste linie i dokumentet så vi altid skriver efter den. Vi udskiller og undersøger også datoen for den sidst skrevne linie så vi er klar til at lave en datolinie hvis det er nødvendigt.

```
$handle = fopen("logfile.txt", "a");
if($olddate < date("d-m-Y")) { fwrite($handle, "\n----- ".date("d-m-Y")." -----
-----\n\n"); }
fwrite($handle, "[".date("d-m-Y H:i:s")."] ".$indput." \n");
fclose($handle);
```

Her kommer så den centrale del af funktionen, delen hvor vi skriver til logfilen. Først gør vi klar til skrivning ved at åbne filen med `fopen()`. Så undersøger vi om datoen for den sidste linie er ældre end dags dato og hvis den er det skriver vi en datolinie med dags dato. Så, nu er vi endeligt klar til at skrive i scriptet. Det gør vi med `fwrite()` læg mærke til at vi skriver en dato i [] klammerne ud for tekstens så vi altid ved hvornår den er skrevet. Efter vi har skrevet lukkes filen med `fclose()`.

Brug af scriptet

Når hele funktionen er sat ind på ens side er det meget nemt at skrive til sin logfil. Det gøres (som de fleste nok har gættet) ved at kalde funktionen.

```
LOGFILE("en tekst til min logfile");
LOGFILE("og en anden tekst og en ip: #IP");
LOGFILE("men så gider vi heller ikke mere");
```

Giver så følgende resultat i logfilen:

```
----- 18-11-2005 -----

[18-11-2005 17:53:25] en tekst til min logfile
[18-11-2005 17:53:25] og en anden tekst og en ip: 235.155.10.5
[18-11-2005 17:53:25] men så gider vi heller ikke mere
```

Sikkerhed!

Efter som logfilen ligger i roden af dit system er den jo åben for alle der vil læse i den... og det er jo ikke så smart. Men vi er heldigvis så privilegerede at hvis man ligger på en Apache server (hvilket de fleste gør) kan man bruge lidt smart kode i en .htaccess -fil. Hvis dette ikke siger dig noget kan jeg kun anbefale to glimrende artikler på webcafe.dk:

www.webcafe.dk/artikler/apache/htaccesspassword/
www.webcafe.dk/artikler/apache/htaccesspassword2/

Men baseret på disse to artikler kan man lynhurtigt lave et glimrende lille script der kun beskytter din logfil.

```
AuthName "-- Logfile! Ingen Adgang! --"
AuthType Basic
AuthUserFile /et/sted/på/din/server/htpasswd

<Files logfile.txt>
require valid-user
</Files>
```

Denne kode gemmer du i din .htaccess -fil samme sted som din logfil og vupti, en password beskyttet logfil.

Men hvad skal jeg bruge det til?

Som jeg også skrev i starten af denne artikel er logfiler klart undervurderet til brug på hjemmesider. Jeg bruger personligt logfiler til rigtig meget.

På mindre hjemmesider med et begrænset antal besøg om dagen kan man bruge dem til dybte gående analyser af folks vandren på ens sider. Dette gør at du nemt og hurtigt kan se hvad folk laver på din side, hvilke sektioner de besøger og ikke mindst hvilke de IKKE besøger.

Til større sider er logfiler perfekte til at gemme erros i. F.eks hvis du har en side med mange forskellige og vekslende ting i din DB så er logfilen god til at gemme SQL syntaks erros i. Mange af disse erros vil du måske ikke opdage og deroft kan filen hjælpe som ekstra sikkerhed.

Men mulighederne er uendelige. Så god fornøjelse ;)

HKB

Kommentar af lenk d. 06. Dec 2005 | 1

Udemærket artikel til folk som selv vil lave en web baseret logging

Kommentar af rasmus-madsen d. 01. Dec 2005 | 2

Super dejlig.

Lidt, men kun lidt smartere end den jeg har lavet ;)

Kommentar af nielspt d. 15. Jan 2007 | 3

Rigtig fin artikel! Ville dog gerne have uddybet, hvordan logfilen holder øje med hvilke sektioner besøgende besøger og ikke-besøger.

Kommentar af el_barto (nedlagt brugerprofil) d. 30. Nov 2005 | 4

Lidt slå/stavefejl men ellers en god og nyttig artikel. Det kunne være sjovt med en sammenligning med en database-baseret løsning og regne lidt på hvilken der er hurtigst eksekveret.

Kommentar af nizo d. 16. Dec 2005 | 5

Rigtig fed artikel! Jeg bruger personligt min MySQL database til at holde styr paa min logs, og ikke .txt filer :) Jeg synes det er lidt nemmere at hente tingene ud i admin delen paa den maade..

Kommentar af gentlebug d. 27. Jun 2006 | 6

God artikel, har tidl. brugt tekst-filer til mine logs, gik senere over til MySQL da dette er nemmere at søge i, men denne artikel fik mit til at gå tilbage til de gode gamle tekstfiler, det er også lidt mere blær-blær med en tekstfil ;) - det er svært at snakke om design i en tekstfil, men kan godt lide din opstilling :)
- Godt arbejde :)