

Scripting: PHP-Scripte auf der Commandozeile / per Cron ausführen

Problem:

Manche Dinge (Newsletter-Versand, Datenbank-Aktionen, etc.) will man gerne Zeitgesteuert starten. Andere Funktionen möchte man aus Sicherheitsgründen ausserhalb des Web-DocumentRoots verstecken. Oder man will auch Scripte für die Systemadministration schreiben, ohne Perl- oder Shell-Script-Programmierung zu erlernen.

Schließlich können PHP-Scripte auch in der Commando-Zeile ausgeführt werden. Was ist aber dabei zu beachten?

Ansatz 1:

PHP-Scripte als Browser-Aufrufe.

Es gibt zwei Programme die (z.B. per Cron/Shell) PHP-Scripte per URL aktivieren können: [wget](#) und [lynx](#). (Es gibt noch weitere Consolen-Webbrowser, aber ich bleib mal bei den beliebtesten.)

Ein Aufruf könnte dann so aussehen:

```
wget 'http://www.meinedomain.tld/admin/newsletter.php'
```

Mithilfe von Pipes kann man sich das Ergebnis z.B. dann zumailen lassen oder in ein Logfile schreiben.

Nachteil:

Die Scripte sind an die Ausführungszeiten des Webservers/PHP gebunden. Sprich, wenn das Script nicht innerhalb der (z.B.) eingestellten 30 Sekunden zum Ende kommt, wird es abgebrochen.

Ansatz 2:

PHP-Scripte per CLI aufrufen.

CLI ist das 'Command-Line-Interface' von PHP. Soll heißen: Ein PHP-Interpreter ohne Webserver (vergleichbar mit [perl](#)). Unter Linux/Unix wird dieser mit `php script.php` gestartet.

Ein Eintrag im Cron braucht aber immer die vollständigen Pfade:

```
/usr/bin/php /home/www/webX/html/admin/newsletter.php
```

(Auch hier kann wieder mit Pipes gearbeitet werden.)

Nachteil:

Eine evtl. vorhandene User-Authentifizierung kann weder per HTTP-Protokoll noch per Cookie statt

Scripting: PHP-Scripte auf der Commandozeile / per Cron ausführen

finden.

Ansatz 3:

Reine PHP-Scripte für die Shell.

Der Ansatz ist der selbe wie beim Zweiten. Diesmal soll aber der explizite Aufruf des PHP-CLI gespart werden, so daß ein `./newsletter.php` bereits das gewünschte erreicht:

Dazu muß eine Shebang-Zeile in das PHP-Script eingefügt werden:

```
#!/usr/bin/php
```

und danach das Script mit Ausführungsrechten versorgt werden:

```
chmod +x newsletter.php
```

(Nun könnte man die Änderung `.php` übrigens auch weg lassen.)

Nachteil:

Die Scripte sind so nicht mehr fürs Web zu gebrauchen. Ausser man führt sie als CGI aus. (Endung von `.php` zu `.cgi` ändern.)

Eindeutige ID: #1090

huschi

2006-01-03 18:38