



Professional CronJob-Service

Startseite

Aktuelles

News
Pressemitteilungen

SELFPHP

Über SELFPHP
Werbung
SELFPHP Sponsoren
SELFPHP Medienpartner
SELFPHP Partner
Kontakt
Impressum
Datenschutz

shopware Partner

Unser Profil
shopware Plugins
shopware 4

SELFPHP Dokumentation

PHP-Skripte / Kochbuch
PHP Befehlsreferenz
PHP 5 Praxisbuch
PHP Code Snippets
SELFPHP in Buchform

Premium-Partner

Webhosting/Serverlösungen

Anbieterverzeichnis

Globale Branchen
Webhosting/Server

CronJob-Service

Standard & Professional
Versionsunterschiede
Registrieren

Services

Forum
RSS-Feeds
Downloads
SELFPHP auf Heft-CD

Produktempfehlung



News und Highlights
PHP auf Windows
Windows & SQL Server

Dateidownload realisieren

Am Ende der Seite finden Sie das komplette Beispiel als [Download](#).

Systemvoraussetzung

- Linux
- Windows
- PHP 3
- PHP 4
- PHP 5

Datei(en)

download.php

Problem

Sicher standen Sie schon öfter vor dem Problem, dass Sie eine Datei zum Download anbieten wollten, diese aber nicht direkt angesprochen werden durfte. Gerade wenn Sie einen Download mitzählen möchten und den Wert in eine Datenbank speichern, müssen Sie die Dateien mit einer anderen Variante ausliefern.

Eventuell verkaufen Sie aber auch auf Ihrer Webseite spezielle Bilder, die nicht direkt abgerufen werden dürfen, sondern erst nach einer Freigabe. Sie werden nachfolgend eine Lösung für diese Probleme finden.

Lösung

Das Problem ist mit wenigen Handgriffen beseitigt. Sie sollten allerdings darauf achten, dass das Verzeichnis mit den Dateien nicht über den Browser zu erreichen ist – also außerhalb des Document-Root liegt.

Sie können alternativ auch ein Verzeichnis mit einem Verzeichnisschutz erstellen. Im Kapitel „Verschlüsselung/Prüfsummen“ finden Sie ein Programm, das Ihnen den Verzeichnisschutz automatisch erstellt.

Weiterhin müssen Sie darauf achten, dass Sie bei dem angebotenen Download jeweils auch die richtigen Mime-Typen mitliefern. Dabei besteht ein Mime-Typ aus zwei Bereichen, zum einen die Angabe des Medientyps und zum anderen die Angabe des Subtyps. Beide Werte sind lediglich mit einem Schrägstrich (/) voneinander getrennt.

Beispiel

```
image/png
application/pdf
```

Nachfolgend finden Sie eine Auflistung von verschiedenen und häufig genutzten MimeTypen. Falls Sie weitere Typen benötigen, erhalten Sie auf der folgenden Webseite eine sehr große Auswahl an verschiedenen Typen. Sie haben dort auch die Möglichkeit, die Liste der MimeTypen zu erweitern. Webseite von IANA (Internet Assigned Numbers Authority):

- <http://www.iana.org/assignments/media-types/>

Content Types

:: Premium-Partner ::
Webhosting/Serverlösungen



weitere Premium-Partner

:: SELFPHP Sponsor ::



weitere Sponsoren



:: Buchempfehlung ::
Handbuch der Java-
Programmierung



zur Buchempfehlung

:: Anbieterverzeichnis ::

Webmatrix & Tools
Förderprogramme

Internet Security

sponsored by 

Suche

Suchbegriff

GO!

:: Anbieterverzeichnis ::
Globale Branchen

Informieren Sie sich über ausgewählte Unternehmen im Anbieterverzeichnis von SELFPHP [↗](#)

:: SELFPHP Forum ::

Fragen rund um die Themen PHP?
In über 130.000 Beiträgen finden Sie sicher die passende Antwort! [↗](#)

:: Newsletter ::

Abonnieren Sie hier den kostenlosen SELFPHP Newsletter!

Vorname:

Name:

E-Mail: 

© 2001-2022  SELFPHP Inh. Damir Enseleit, info@selfphp.de, Impressum, Kontakt
© 2005-2022  PHP5 Praxisbuch - Matthias Kannengiesser, matthias.kannengiesser@selfphp.de

text	Textdateien
image	Grafikdateien
video	Videodateien
audio	Sound-Dateien
application	programmgebundene Dateien
multipart	mehrteilige Daten (z. B. bei einem Formular inkl. Dateiupload)
message	Nachrichten
model	Dateien mit mehrdimensionalen Strukturen

Content Types/Subtypes

.avi	video/x-msvideo	.ppt	application/mspowerpoint
.class	application/octet-stream	.ppz	application/mspowerpoint
.css	text/css	.ps	application/postscript
.doc	application/msword	.qt	video/quicktime
.eps	application/postscript	.ra	audio/x-realaudio
.exe	application/octet-stream	.ram	audio/x-pn-realaudio
.gif	image/gif	.rm	audio/x-pn-realaudio
.gtar	application/x-gtar	.rpm	audio/x-pn-realaudio-plugin
.gz	application/x-gzip	.rtf	text/rtf
.htm	text/html	.rtx	text/richtext
.html	text/html	.sgm	text/sgml
.jpe	image/jpeg	.sgml	text/sgml
.jpeg	image/jpeg	.tar	application/x-tar
.jpg	image/jpeg	.tcl	application/x-tcl
.js	application/x-javascript	.tif	image/tiff
.midi	audio/midi	.tiff	image/tiff
.mov	video/quicktime	.txt	text/plain
.movie	video/x-sgi-movie	.vrml	model/vrml
.mp2	audio/mpeg	.wav	audio/x-wav
.mp3	audio/mpeg	.wrl	model/vrml
.mpe	video/mpeg	.xbm	image/x-xbitmap
.mpeg	video/mpeg	.xlc	application/vnd.ms-excel
.mpg	video/mpeg	.xll	application/vnd.ms-excel
.mpga	audio/mpeg	.xlm	application/vnd.ms-excel
.pbm	image/x-portable-bitmap	.xls	application/vnd.ms-excel
.pdf	application/pdf	.xlw	application/vnd.ms-excel
.png	image/png	.xml	text/xml
.pps	application/mspowerpoint	.zip	application/zip

Wie Sie sehen, ist die Liste bereits sehr groß, aber in der Regel braucht man nicht mehr als eine Hand voll, um gewisse Dateien ausliefern zu können.

Wir kümmern uns jetzt aber nicht weiter um diese Dateitypen, sondern schauen uns den Programmcode an. Zuerst werden wir uns mit der Funktion beschäftigen, die nachher die Datei ausliefern wird.

```
function makeDownload($file, $dir, $type)
@param string $file
@param string $dir
@param string $type
@return void
```

Webhosting/Serverlösungen

Suchen Sie den für Sie passenden IT-Dienstleister für Ihr Webhosting-Paket oder Ihre Serverlösung?

Sie sind nur ein paar Klicks davon entfernt! [↗](#)

Ausgewählter Tipp im Bereich PHP-Skripte

Übermittelte Dateinamen überprüfen

Weitere interessante Beispiele aus dem SELFPHP Kochbuch finden Sie im Bereich PHP-Skripte [↗](#)

SELFPHP Code Snippet

Temporäres Verzeichnis leeren (PHP 4)

Weitere interessante Code Snippets finden Sie auf SELFPHP im Bereich PHP Code Snippets [↗](#)

Die Funktion `makeDownload()` erwartet als Parameter den Dateinamen für den Download, das Verzeichnis, in dem die Datei liegt (inkl. abschließendem Slash (/)!), und den Mime-Typ der jeweiligen Datei.

Wurden alle Parameter übertragen, wird zuerst der Mime-Type der Datei zum Browser gesendet (**37**) und im Anschluss daran der Dateiname (**39**) für die Speicherung auf dem Clientrechner. Sie können an dieser Stelle einen freien Namen für die Datei auswählen, müssen also nicht den realen Namen der Datei nehmen.

Zum Schluss müssen wir jetzt noch die Datei ausgeben. Dafür nutzen wir `readfile()` (**41**), lesen also die komplette Datei ein und geben Sie am Browser aus. Der Benutzer erhält z. B. in einer Zip-Datei den Dialog für das Speichern der Datei.

```
35: function makeDownload($file, $dir, $type) {
36:
37:     header("Content-Type: $type");
38:
39:     header("Content-Disposition: attachment; filename=\"$file\"");
40:
41:     readfile($dir.$file);
42:
43: }
```

Beispiel 6.2: download.php

Sie müssen zuerst zwei Konfigurationsparameter setzen. Zum einen benötigen Sie den Pfad (inkl. abschließendem Slash) zu Ihrem Downloadverzeichnis (**51**), zum anderen den Mime-Typen für die Datei (**53**). Falls Sie später mehrere verschiedene Dateien zum Download anbieten wollen, könnten Sie die Mime-Typen in ein Array schreiben und jeweils den richtigen Wert zuweisen.

Falls jetzt per GET ein Dateiname übermittelt wird, also nicht leer ist und die Variable keine Slashes (/) enthält, so können wir die Abfrage (**55**) weiter durchlaufen. Die Abfrage dient lediglich zum Schutz, damit niemand eine Datei außerhalb Ihres Verzeichnisses downloaden kann.

Im nächsten Schritt (**56**) überprüfen wir, ob die Datei vorhanden ist und sie somit zum Download angeboten werden kann. Wurde die Datei gefunden, so rufen wir unsere Funktion `makeDownload()` auf und übergeben den Dateinamen, das Verzeichnis und den Mime-Type der Datei.

```
51: $dir = '/is/htdocs/wp1042434_BDE06BJ7VB/www/download/';
52:
53: $type = 'application/zip';
54:
55: if(!empty($_GET['file']) && !preg_match('!=', $_GET['file'])) {
56:     if(file_exists($dir.$_GET['file'])) {
57:         makeDownload($_GET['file'], $dir, $type);
58:     }
59:
60: }
```

Beispiel 6.3: download.php

Der Aufruf würde dann für eine Datei z.B. folgendermaßen aussehen:

```
http://www.ihreDomain.de/download.php?file=dieDatei.zip
```

Der Download wurde sehr schnell gelöst. Sie könnten das Programm noch erweitern und vor allem sicherer machen. Wir möchten Ihnen hierfür ein paar Tipps geben, sodass Sie es selbst umsetzen können.

Erstellen Sie ein Array mit den Dateinamen, die zum Download angeboten werden. Wird jetzt die Variable `$_GET['file']` übermittelt, überprüfen Sie, ob der übermittelte Dateiname auch im Array vorkommt. Dies ist ein zusätzlicher Schutz, damit keine anderen Dateien ausgeliefert werden.

Erstellen Sie ein Array mit den Mime-Typen und setzen anhand der Dateiendung jeweils automatisch den richtigen Mime-Typ für die Datei.

Erzeugen Sie bei der Auslieferung der Dateien für die Header-Content-Disposition einen zufälligen Dateinamen und liefern diesen aus.

Es gibt also noch viele Ansätze, die man berücksichtigen kann. Lassen Sie Ihre Fantasie spielen, und erweitern Sie das Programm.

Sie sollten allerdings immer Ihre Downloads vorher mit verschiedenen Browsern testen, um sicher gehen zu können, dass die Dateien richtig ausgeliefert werden.



Dieses Skript aus dem SELFPHP KOCHBUCH wurde von SELFPHP unter dem "**Tarif Mc500**" von McAc.net-Webhosting erfolgreich ausgeführt und getestet!

Auf der Übersichtseite unter "**McAc.net – Webhosting zu diesem Buch**" finden Sie weitere Informationen zu dem Webhostingpaket, dass durch SELFPHP getestet wurde.

Download

Info	ZIP	BZIP2	TARGZ
Größe	954,00 Byte	1.10 Kilobyte	1.03 Kilobyte
<u>Alle Beispiele als PHP-Datei(en)</u>			
Hier haben Sie die Möglichkeit, sich sämtliche auf dieser Seite gezeigten Beispiele als PHP-Dateien direkt downzuloaden.			
Umständliches Copy & Paste ist daher nicht mehr notwendig, da alle Beispiele sofort ausprobiert werden können.			
Sie haben bei der Auswahl der bereitgestellten Downloads mehrere Möglichkeiten, je nachdem welches Komprimierungsverfahren Sie bevorzugen.			

[Zurück zur Übersichtseite](#)
